The Strength of Multilinear Proofs

Ran Raz* Iddo Tzameret†

August 10, 2007

Abstract

We introduce an algebraic proof system that manipulates multilinear arithmetic formulas. We show that this proof system is fairly strong, even when restricted to multilinear arithmetic formulas of a very small depth. Specifically, we show the following:

1. Algebraic proofs manipulating depth 2 multilinear arithmetic formulas polynomially simulate Resolution, Polynomial Calculus (PC) and Polynomial Calculus with Resolution (PCR) proofs;
2. Polynomial size proofs manipulating depth 3 multilinear arithmetic formulas for the functional pigeonhole principle;
3. Polynomial size proofs manipulating depth 3 multilinear arithmetic formulas for Tseitin's graph tautologies.

By known lower bounds, this demonstrates that algebraic proof systems manipulating depth 3 multilinear formulas are strictly stronger than Resolution, PC and PCR, and have an exponential gap over bounded-depth Frege for both the functional pigeonhole principle and Tseitin's graph tautologies.

We also illustrate a connection between lower bounds on multilinear proofs and lower bounds on multilinear circuits. In particular, we show that (an explicit) super-polynomial size separation between proofs manipulating *general* arithmetic circuits and proofs manipulating *multilinear* circuits implies a super-polynomial size lower bound on multilinear circuits for an explicit family of polynomials.

Keywords: proof complexity, algebraic proofs, multilinear arithmetic formulas, polynomial calculus, propositional pigeonhole principle, Tseitin tautologies.
MSC 2000: 03F20, 68Q17, 13P10

*Department of Computer Science, Weizmann Institute, Rehovot 76100, Israel, email: ranraz@wisdom.weizmann.ac.il. Supported by The Israel Science Foundation and The Minerva Foundation.

†School of Computer Science, Tel Aviv University, Tel Aviv 69978, Israel, email: tzameret@post.tau.ac.il. Supported in part by The Israel Science Foundation (grant no. 250/05).

Contents

1. Introduction

This paper considers an algebraic proof system Formula Multilinear Calculus (fMC) that manipulates multilinear arithmetic formulas. A multilinear proof (that is, an fMC proof) begins with an initial set of multilinear polynomial equations representing the clauses of a CNF formula (where a polynomial is multilinear if in each of its monomials the power of every variable is at most one), and the goal is to prove that the CNF formula is unsatisfiable by showing the equations have no $0,1$ solutions over a given fixed field.

Specifically, let Q be a set of initial multilinear polynomial equations in the formal variables $\{x_1,\ldots,x_n,\bar{x}_1,\ldots,\bar{x}_n\}$ over some fixed field. A multilinear proof of the insolvability of Q is a sequence of multilinear polynomial equations, where *each polynomial is represented as (an arbitrarily chosen) multilinear arithmetic formula*. The sequence uses the initial equations plus the polynomial equations $x_i+\bar{x}_i-1=0$ and $x_i\cdot\bar{x}_i=0$ (for all variables $x_i,\bar{x}_i$) as axioms, and terminates with the unsatisfiable equation $1=0$. Derivations of polynomial equations in the sequence are done by applying the following two basic algebraic inference rules to previous equations in the sequence:

- from $p=0$ and $q=0$ one can deduce $\alpha\cdot p+\beta\cdot q=0$, where α,β are elements of the field;
- from $p=0$ one can deduce $q\cdot p=0$, for any polynomial q such that $q\cdot p$ is multilinear.

(The inclusion of the equalities $x_i+\bar{x}_i-1=0$ and $x_i\cdot\bar{x}_i=0$ forces the variables x_i and $\bar{x}_i$ to take on only the Boolean values 0 and 1, where $\bar{x}_i$ takes the negative value of x_i.) If such a sequence exists then there is no assignment of $0,1$ values that satisfies all the initial equations. Such a proof of insolvability is then called a *multilinear refutation* of the initial polynomial equations.

We can obtain in this way a proof system for (unsatisfiable) CNF formulas. Given a CNF formula F in the variables $x_1,\ldots,x_n$ we translate F to a system of multilinear polynomial equations in the variables $x_1,\ldots,x_n,\bar{x}_1,\ldots,\bar{x}_n$. Each clause C of F translates into a multilinear polynomial equation $q_C=0$. F is satisfiable if and only if the system of polynomial equations $q_C=0$, for all clauses C of F, has a common root in the field, where the root also satisfies the axioms $x_i+\bar{x}_i-1=0$ and $x_i\cdot\bar{x}_i=0$ (for all variables $x_i,\bar{x}_i$). For example, the CNF $(x_1\vee x_2\vee\neg x_3)\wedge(\neg x_2\vee x_4)$ translates into the polynomial equations $\bar{x}_1\cdot\bar{x}_2\cdot x_3=0\,,\;\; x_2\cdot\bar{x}_4=0$.

The minimal refutation size of a given set of initial polynomial equations (i.e., the number of symbols that takes to write down the refutation of these equations) is a natural measure for the strength of an algebraic proof system. In algebraic proof systems such as the Polynomial Calculus (PC) and Polynomial Calculus with Resolution (PCR) (cf. Clegg *et al.* (1996)[1], Alekhnovich *et al.* (2002) for PC and PCR, respectively), one represents the polynomials inside refutations as explicit sum of monomials. Then, the size of a PC or a PCR refutation is usually defined as the total number of all monomials appearing in the refutation. On the other hand, in the multilinear proof system presented in this paper, polynomials inside refutations are represented as *multilinear arithmetic formulas*. Accordingly, the size of a multilinear refutation is defined to be the total size of all the multilinear arithmetic formulas appearing in the refutation.

The aim of this paper is first, to show that algebraic proof systems manipulating multilinear arithmetic formulas – and further, very small depth multilinear arithmetic formulas – constitute rather strong proof systems, that are strictly stronger than PC, PCR and Resolution. Moreover, such multilinear proof systems are capable of refuting efficiently (negations of) tautologies that were

[1] In Clegg *et al.* (1996) the Polynomial Calculus was referred to as *the Gröbner proof system*.

found hard[2] for other proof systems, such as the bounded-depth Frege proof system. And second, to illustrate a link between multilinear proofs and multilinear arithmetic circuit lower bounds.

1.1. Background and motivation. Understanding the efficiency of propositional proof systems and in particular proving lower bounds on the size of propositional proofs are fundamental problems in both logic and computational complexity. This area of research has gained much interest in the last two decades. A propositional proof system is usually described by a finite set of inference rules and axiom schemata. A propositional proof is then a derivation of some tautology, that applies the prescribed inference rules to the set of axioms. We can sometimes take the dual view in which proof systems establish that some formula is unsatisfiable by deriving FALSE from the formula and axioms. Thus, the proofs in such systems are usually called *refutations*.

In the course of investigating the complexity of different propositional proof systems, connections were found between proofs manipulating Boolean formulas and proofs manipulating polynomials over a field (cf. Beame *et al.* (1996)). Proof systems manipulating polynomials are called *algebraic proof systems*. Algebraic proofs usually demonstrate that a collection of polynomial equations, derived from the clauses of an unsatisfiable CNF formula, has no $0,1$ solutions over some fixed field.

The Polynomial Calculus proof system (PC), introduced in Clegg *et al.* (1996), is a well studied algebraic proof system. Fix some field $\mathbb{F}$ and let Q be a collection of multivariate polynomial equations $Q_1 = 0, \ldots, Q_m = 0$, where each Q_i is taken from the ring of polynomials $\mathbb{F}[x_1, \ldots, x_n]$. In PC the fact that the collection Q has no $0,1$ solutions over the field $\mathbb{F}$ is proved by using the following basic algebraic inference rules: from two polynomial equations $p = 0$ and $q = 0$, we can deduce $\alpha \cdot p + \beta \cdot q = 0$, where α, β are elements of $\mathbb{F}$; and from $p = 0$ we can deduce $x_i \cdot p = 0$, for any variable x_i $(1 \leq i \leq n)$. A sequence of polynomials that uses $Q_1 = 0, \ldots, Q_m = 0$ and $x_i^2 - x_i = 0$ (for any variable x_i) as initial polynomial equations, follows the above algebraic inference rules, and ends with $1 = 0$, is called a PC *refutation* of the polynomial equations $Q_1 = 0, \ldots, Q_m = 0$.

In recent years intensive efforts were made to prove lower bounds on the maximal *degree* of polynomials appearing in PC refutations of some set of initial polynomials (cf. Razborov (1998), Impagliazzo *et al.* (1999), Buss *et al.* (2001), Ben-Sasson & Impagliazzo (1999), Alekhnovich *et al.* (2000), Alekhnovich & Razborov (2001), Razborov (2002-2003)). These lower bounds imply a lower bound on the *size* of the refutations only when polynomials are represented as *a sum of monomials*, that is, as depth 2 arithmetic formulas. For instance, Impagliazzo *et al.* (1999) showed that any degree lower bound that is linear in the number of variables implies an exponential lower bound on the number of monomials in the refutation.

With respect to lower bounds on the refutation size of algebraic proof systems other than PC and PCR (in which the size of refutations is measured by the number of monomials appearing in the refutations), not much is known. Moreover, extending the PC proof system by allowing it to manipulate general (i.e., not necessarily multilinear) arithmetic formulas makes this proof system considerably strong (cf. Buss *et al.* (1996/97); Grigoriev & Hirsch (2003); Pitassi (1997)). In particular, such an extended PC proof system that manipulates general arithmetic formulas polynomially simulates the entire Frege proof system, which is regarded as a rather strong proof system, and for which no super-polynomial size lower bounds are currently known. Thus, if one seeks to prove

[2] Given a proof system P and a tautology τ, we say that τ *is hard for* P, if there is no polynomial-size P-proof of τ (or equivalently, if there is no polynomial-size P-refutation of $\neg\tau$).

size lower bounds on refutation size, it is more reasonable to concentrate on (apparently weaker) extensions of PC (and PCR).

Furthermore, it is well known in proof complexity theory (cf. Beame & Pitassi (1998)) that there is an (informal) correspondence between circuit-based complexity classes and proof systems based on these circuits (i.e., proofs in which the proof lines[3] consist of circuits from the prescribed circuit-class). Moreover, super-polynomial size lower bounds on *proofs* manipulating circuits from a given circuit-class were only found after super-polynomial size lower bounds were already proved for *circuits* from the circuit-class itself. Keeping in mind this correspondence, it is important to note that super-polynomial lower bounds on multilinear arithmetic formulas for the determinant and permanent functions, as well as other functions, were recently proved in Raz (2004a,b) and Aaronson (2004). On the other hand no super-polynomial lower bounds are known for general arithmetic formulas.

In light of the aforesaid, the results of this paper show that algebraic proof systems manipulating multilinear arithmetic formulas (even of a very small depth) constitute on the one hand fairly strong proof systems extending PC and PCR — and on the other hand, the corresponding circuit-class (i.e., multilinear formulas) does have known super-polynomial lower bounds. This makes the multilinear proof systems a prime target for attempts to prove refutation-size lower bounds.

Moreover, as mentioned above, the correspondence between proof systems and circuit-classes is not a formal one, but instead it acts more as a working conjecture in proof complexity theory. Nevertheless, using multilinear proofs we are able to pinpoint an interesting case where this correspondence can be formulated explicitly.

1.2. Overview of proof systems. We now give a list of proof systems we consider in this paper (we shall define these formally in the sequel). A proof sequence in some proof system is also called a *refutation* if its terminal formula (or terminal polynomial equation) is FALSE (or the unsatisfiable polynomial equation 1=0).

The *algebraic* proof systems we consider are the following:

Polynomial Calculus, denoted PC (described above). A proof system for the set of unsatisfiable CNF formulas written as an unsatisfiable set of polynomial equations over a field. Each polynomial in a PC proof is represented as an explicit sum of monomials.

Polynomial Calculus with Resolution, denoted PCR. This is an extension of PC where for each variable x_i a new formal variable $\bar{x}_i$ is added. The variable $\bar{x}_i$ equals $1 - x_i$. Each polynomial in a PCR proof is represented as an explicit sum of monomials. PCR can polynomially simulate both PC and Resolution.

Formula Multilinear Calculus, denoted fMC (described above). fMC is a proof system for the set of unsatisfiable CNF formulas written as unsatisfiable set of multilinear polynomial equations over a field. Each polynomial in an fMC proof is a multilinear polynomial represented as a multilinear arithmetic formula (we consider arithmetic formulas that can use unbounded fan-in + (addition) and × (product) gates). It is important to note that we allow each multilinear polynomial in an fMC proof to be represented by an *arbitrarily chosen* multilinear formula. Hence, fMC can be considered a semantic proof system which is probabilistically polynomial-time verifiable. This is in contrast with the usual requirement that proofs should be polynomial-time verifiable (see discussion in Section 2.3.5).

[3] A formula (or circuit) in a proof sequence is sometimes referred to as *a proof line*.

Depth-k Formula Multilinear Calculus, denoted depth-k fMC. This is a restriction of fMC to multilinear arithmetic formulas of depth at most k.

cMC *and* cPCR *proof systems*. The cMC proof system is similar to fMC, except that multilinear polynomials are represented by multilinear arithmetic *circuits* (instead of multilinear arithmetic formulas). In the same manner, cPCR is a proof system similar to PCR, except that polynomials are represented by (general) arithmetic circuits (instead of sums of monomials).

Other proof systems (manipulating Boolean formulas) considered in this paper are:

Resolution. A proof system for unsatisfiable CNF formulas. Each Resolution proof-line consists of a clause (i.e., a disjunction of variables or their negations). The last line of a Resolution refutation is the empty clause, which has no satisfying assignment.

Bounded-depth Frege. Usually considered as a proof system for the set of Boolean tautologies. The lines in a bounded depth Frege proof consists of *constant-depth* formulas over the connective NOT and the unbounded fan-in connectives AND, OR. We can consider bounded-depth Frege to be also a proof system for the set of *unsatisfiable* Boolean formulas, by treating a proof sequence (starting from some initial set of unsatisfiable formulas) that ends with FALSE, as a refutation.

We shall consider all the proof systems above to be proof systems for the set of unsatisfiable CNF formulas (or polynomial translations of unsatisfiable CNF formulas). We shall say that a proof system P_2 *polynomially simulates* another proof system P_1 if for any unsatisfiable CNF formula F and a P_1 refutation π of F, there exists a refutation of F in P_2 of size polynomial in the size of π. In case P_2 polynomially simulates P_1 while P_1 does not polynomially simulates P_2 we say that P_2 is *strictly stronger* than P_1. Given an unsatisfiable CNF formula F, we say that P_2 *has an exponential gap over* P_1 *for* F if there exists a polynomial size P_2 refutation of F and the smallest P_1 refutation of F is of exponential size.

1.3. Our results. We prove three kinds of results. The results of the first kind are polynomial simulations. The results of the second kind are upper bounds on the refutation size of combinatorial principles that were found hard for other proof systems. Both the simulations and upper bounds results are valid when one restricts the multilinear arithmetic formulas in the refutations to depth at most 3. The third kind of result concerns the problem of proving multilinear arithmetic circuit size lower bounds in connection to multilinear proof systems. Specifically, we show the following simulation and upper bounds results:

1. Depth-2 fMC polynomially simulates Resolution, PC and PCR (Sections 4 and 5);
2. Depth-3 fMC over fields of characteristic 0 has polynomial-size refutations of the Functional Pigeonhole Principle (Section 7);
3. Depth-3 fMC has polynomial-size refutations of the Tseitin mod p contradictions (for any p) over fields of characteristic $q \nmid p$, that include a primitive p-th root of unity (Section 8).

Haken (1985) has shown an exponential lower bound on the size of Resolution refutations of the Functional Pigeonhole Principle. Moreover, exponential lower bounds on the size of Resolution refutations of certain Tseitin mod 2 tautologies (that is, Tseitin tautologies based on expanding graphs) are also known (see Ben-Sasson & Wigderson (1999); Urquhart (1987)). We conclude then, by (1,2,3), that depth-3 fMC is strictly stronger than Resolution.

From the known exponential lower bounds on PC and PCR refutation size of certain Tseitin mod p tautologies (cf. Alekhnovich *et al.* (2000); Ben-Sasson & Impagliazzo (1999); Buss *et al.* (2001)), we conclude by (1,3) that depth-3 fMC is strictly stronger than PC and PCR.

Note also that Razborov (1998) and subsequently Impagliazzo *et al.* (1999) have shown an exponential-size lower bound on the size of PC (and PCR) refutations of a low-degree version of the Functional Pigeonhole Principle. Our depth-3 fMC upper bound is also applicable to this low-degree version (see Section 7 for a discussion).

Exponential lower bounds on the size of bounded-depth Frege proofs of the Functional Pigeonhole Principle were proved in Pitassi *et al.* (1993) and independently in Krajíček *et al.* (1995). Thus (2) shows an exponential gap of depth-3 fMC over bounded depth Frege for the Functional Pigeonhole Principle. Similarly, an exponential lower bound on the size of bounded-depth Frege proofs of certain Tseitin mod 2 tautologies was shown in Ben-Sasson (2002). Thus, (3) implies that also for these Tseitin mod 2 tautologies, depth-3 fMC has an exponential gap over bounded-depth Frege proofs.

In Section 5 we provide a general simulation result for multilinear proofs. Specifically, Let S be a sequence of polynomials (not formulas) that forms a PCR proof sequence for some given set Q of multilinear polynomials, and consider the corresponding sequence S' of multilinear polynomials formed by 'multilinearization' (see Definition 2.2) of the polynomials in S. Then, the general simulation result essentially says that there is an fMC proof of Q of size polynomial in the total size of all the multilinear formulas that compute the polynomials in S'.

By this general simulation we are able (in Section 6) to assert the following: Proving (an explicit) super-polynomial size separation between algebraic proofs manipulating *general* arithmetic circuits and algebraic proofs manipulating *multilinear* arithmetic circuits implies a super-polynomial size lower bound on multilinear arithmetic circuits for an explicit family of polynomials.

2. Preliminaries

For a natural number m, we use $[m]$ to denote $\{1, \ldots, m\}$.

2.1. CNF formulas. A CNF formula over the variables $x_1, \ldots, x_n$ is defined as follows. A literal is a variable x_i or its negation $\neg x_i$. A clause is a disjunction of literals. We treat a clause as a set of literals, that is, we delete multiple occurrences of the same literal in a clause. A CNF formula is a conjunction of clauses.

The size of a clause is the number of literals in it. The size of a CNF is the total size of all the clauses in it.

2.2. Arithmetic and multilinear circuits and formulas.

2.2.1. Arithmetic circuits and formulas. An *arithmetic circuit* is a directed acyclic graph with unbounded (finite) fan-in and unbounded (finite) fan-out. Every leaf of the graph (i.e., a node of fan-in 0) is labeled with either an input variable or a field element. A field element can also label an edge of the graph. Every other node of the graph is labeled with either $+$ or $\times$ (in the first case the node is a plus gate and in the second case a product gate). We assume that there is only one node of out-degree zero, called the *root*. The *size* of an arithmetic circuit C is the total number of nodes in its graph and is denoted by $|C|$.

An arithmetic circuit computes a polynomial in the ring of polynomials $\mathbb{F}[x_1, \ldots, x_n]$ in the following way. A leaf just computes the input variable or field element that labels it. A field

element that labels an edge means that the polynomial computed at its tail (i.e., the node where the edge is directed from) is multiplied by this field element. A plus gate computes the sum of polynomials computed by the tails of all incoming edges. A product gate computes the product of the polynomials computed by the tails of all incoming edges. (Subtraction is obtained using the constant -1.) The output of the circuit is the polynomial computed by the root.

The *depth* of a circuit C is the maximal number of edges in a path from a leaf to the root of C, and is denoted by $\mathrm{dp}(C)$.

We shall consider only *leveled circuits*, that is, circuits where all the nodes of a given level (excluding the bottom level nodes, i.e., the leaves) in the circuit-graph have the *same* labels, and two consequent levels have different labels (i.e., the gates in any path in the circuit alternate between plus and product gates). Any arithmetic circuit with unbounded fan-in gates can be transformed into a leveled circuit that computes the same polynomial, with only a polynomial increase in the size of the circuit. Hence, considering only leveled circuits is not a real restriction here.

We say that a variable x_i *occurs in* an arithmetic circuit if x_i labels one of the leaves of the arithmetic circuit, i.e., x_i is an input variable. We say that an arithmetic circuit has a *plus (product) gate at the root* if the root of the circuit is labeled with a plus (product) gate.

An arithmetic circuit is an *arithmetic formula* if its underlying graph is a tree (with edges directed from the leaves to the root).

2.2.2. Multilinear circuits and formulas. A polynomial is *multilinear* if in each of its monomials the power of every input variable is at most one.

DEFINITION 2.1. *An arithmetic circuit is* a multilinear circuit *(or equivalently,* multilinear arithmetic circuit*) if the polynomial computed by* each *gate of the circuit is multilinear (as a formal polynomial, i.e., as an element of* $\mathbb{F}[x_1, \ldots, x_n]$*). Similarly, an arithmetic formula is* a multilinear formula *(or equivalently,* multilinear arithmetic formula*) if the polynomial computed by* each *gate of the formula is multilinear.*

An additional definition we shall use extensively is the following:

DEFINITION 2.2. *Given a field* $\mathbb{F}$ *and a polynomial* $q \in \mathbb{F}[x_1, \ldots, x_n]$*, we denote by* $\mathbf{M}[q]$ *the unique multilinear polynomial equal to* q *modulo the ideal generated by all the polynomials* $x_i^2 - x_i$*, for all variables* x_i*.*

For example, if $q = x_1^2 x_2 + \alpha x_4^3$ (for some $\alpha \in \mathbb{F}$) then $\mathbf{M}[q] = x_1 x_2 + \alpha x_4$.

Notational conventions. We shall often abuse notation by identifying arithmetic formulas with the polynomials they compute. For instance, if Φ is an arithmetic formula computing the polynomial f, then $\mathbf{M}[\Phi]$ is the multilinear *polynomial* $\mathbf{M}[f]$, and not a formula (note that there can be many arithmetic formulas computing a given polynomial). We can also write, for instance, $\Phi \cdot x_i$ to mean the polynomial $f \cdot x_i$, or $\Phi + x_i$ to mean the polynomial $f + x_i$ (we shall often state explicitly when we refer to the polynomial and not the formula).

Also, given m formulas $\Phi_1, \ldots, \Phi_m$, we usually write $\Phi_1 + \ldots + \Phi_m$ and $\Phi_1 \times \ldots \times \Phi_m$ to designate the *formula* with a plus gate at the root with m children $\Phi_1, \ldots, \Phi_m$, and product gate at the root with m children $\Phi_1, \ldots, \Phi_m$, respectively. When writing a formula like $\Phi_1 \times x_i + \Phi_2$, then the $\times$ gate has clearly precedence over the $+$ gate.

2.3. Algebraic proof systems. Algebraic proof systems are proof systems for finite collections of polynomial equations having no $0,1$ solutions over some fixed field. (Formally, each different field yields a different algebraic proof system.) In this paper, all collections of polynomial equations that are being refuted are translations of CNF formulas, according to a fixed translation scheme we shall define explicitly below (Definition 2.5).

The lines of an algebraic refutation consists of polynomials p_i over the given fixed field. Each such proof line is interpreted as the polynomial equation $p_i = 0$. If we want to consider the *size* of algebraic refutations we should fix the way polynomials inside refutations are represented.

2.3.1. Polynomial Calculus. The Polynomial Calculus is a complete and sound proof system for unsatisfiable CNF formulas translated to polynomial equations.

DEFINITION 2.3. **(Polynomial Calculus (PC)).** *Let $\mathbb{F}$ be some fixed field and let $Q := \{Q_1, \ldots, Q_m\}$ be a collection of multivariate polynomials from $\mathbb{F}[x_1, \ldots, x_n]$. Call the set of polynomials $x_i^2 - x_i$, for all variables x_i ($1 \leq i \leq n$), the set of* Boolean axioms of PC.

A PC proof *from Q of a polynomial g is a finite sequence $\pi = (p_1, ..., p_\ell)$ of multivariate polynomials from $\mathbb{F}[x_1, \ldots, x_n]$ (each polynomial p_i is interpreted as the polynomial equation $p_i = 0$), where $p_\ell = g$ and for each $i \in [\ell]$, either $p_i = Q_j$ for some $j \in [m]$, or p_i is a Boolean axiom, or p_i was deduced from p_j, p_k, where $j, k < i$, by one of the following inference rules:*

Product: *from p deduce $x_i \cdot p$, for some variable x_i;*

Addition: *from p and q deduce $\alpha \cdot p + \beta \cdot q$, for some $\alpha, \beta \in \mathbb{F}$.*

All the polynomials inside a PC *proof are represented as* sums of monomials. *A* PC refutation of *Q is a proof of 1 (which is interpreted as $1 = 0$) from Q.*

The *size* of a PC proof is defined to be the total number of monomials appearing in the polynomials of the proof. The *degree* of a PC proof is the maximum degree of the polynomials in the proof.

Notice that the Boolean axioms have only $0, 1$ solutions.

Note also that the formal variables of the PC proof system are $x_1, \ldots, x_n$. In order to refute in PC an unsatisfiable CNF formula in the variables $x_1, \ldots, x_n$, we translate a CNF formula into a system of polynomials as follows (again, a polynomial p is interpreted as the polynomial equation $p = 0$). A positive literal x_i translates into $1 - x_i$. A negative literal $\neg x_i$ translates into x_i. A clause, i.e., a disjunction of literals $\ell_1 \vee \ldots \vee \ell_k$, translates into the product of the translations of the literals ℓ_i. A CNF is translated into the set of polynomial translations of its clauses. For example, $(x_1 \vee x_2 \vee \neg x_3) \wedge (\neg x_1 \vee \neg x_4)$ translates into the two polynomials $(1 - x_1) \cdot (1 - x_2) \cdot x_3$ and $x_1 \cdot x_4$. It is not hard to see that any assignment of 0,1 (where 0 is interpreted as FALSE and 1 as TRUE) to the variables of a CNF formula F satisfies F if and only if it is a common root of the corresponding system of polynomials, over any given field.

2.3.2. Polynomial Calculus with Resolution. The translation of CNF formulas into collections of polynomials, discussed in the previous paragraph, makes PC unable to polynomially simulate Resolution (see Definition 2.9 for polynomial simulations, and Definition 2.8 for Resolution). For instance, the clause $\bigvee_{i=1}^{n} x_i$ is translated into the polynomial $\Pi_{i=1}^{n}(1 - x_i)$. The number of monomials in $\Pi_{i=1}^{n}(1 - x_i)$ is 2^n, exponential in the number of variables in the clause. For this reason an extension of PC, denoted PCR, that *is* capable of simulating Resolution was defined as follows (cf. Alekhnovich *et al.* (2002)).

DEFINITION 2.4. **(Polynomial Calculus with Resolution (PCR)).** *Let $\mathbb{F}$ be some fixed field and let $Q := \{Q_1, \ldots, Q_m\}$ be a collection of multivariate polynomials from $\mathbb{F}[x_1, \ldots, x_n, \bar{x}_1, \ldots, \bar{x}_n]$. The variables $\bar{x}_1, \ldots, \bar{x}_n$ are treated as new formal variables. Call the set of polynomial equations $x^2 - x$, for $x \in \{x_1, \ldots, x_n, \bar{x}_1, \ldots, \bar{x}_n\}$, plus the polynomial equations $x_i + \bar{x}_i - 1$, for all $1 \leq i \leq n$, the set of* Boolean axioms of PCR*.*

The inference rules*,* proofs *and* refutations *of* PCR *are defined the same as in* PC *(except that in* PCR *the polynomials are taken from $\mathbb{F}[x_1, \ldots, x_n, \bar{x}_1, \ldots, \bar{x}_n]$). Similar to* PC*, all the polynomials in a* PCR *proof are represented as* sum of monomials*.*

The *size* of a PCR proof is defined to be the total number of monomials appearing in the polynomials of the proof. The *degree* of a PCR proof is the maximum degree of the polynomials in the proof. The *number of steps* of a PCR proof is defined to be the number of polynomials in it (i.e., the length of the proof sequence).

Note that the Boolean axioms of PCR have only $0, 1$ solutions, where $\bar{x}_i = 0$ if $x_i = 1$ and $\bar{x}_i = 1$ if $x_i = 0$.

2.3.3. Translation of CNF formulas. In the case of PCR, the polynomial translation of CNF formulas is the following (this is the translation we shall also work with when dealing with multilinear proofs.)

DEFINITION 2.5. **(polynomial translation of CNF formulas).** *The literal x_i translates into $\bar{x}_i$. The literal $\neg x_i$ translates into x_i. A clause, i.e., a disjunction of literals $\ell_1 \vee \ldots \vee \ell_k$, translates into the product of the translations of its literals. A CNF is translated into the set of polynomial translations of its clauses.*

Note that this way the clause $\bigvee_{i=1}^{n} x_i$ translates into $\Pi_{i=1}^{n} \bar{x}_i$, which consists of only one monomial (see the discussion in the first paragraph of Section 2.3.2).

It is clear that any assignment of 0,1 values to the variables $x_1, \ldots, x_n$ of a CNF formula F satisfies F if and only if it is a common root of the set of polynomial translations of the clauses of F over the given fixed field (where each variables $\bar{x}_i$ gets the negative value of x_i , i.e., $\bar{x}_i = 0$ if $x_i = 1$ and $\bar{x}_i = 1$ if $x_i = 0$).

2.3.4. Formula Multilinear Calculus. We now come to define proof systems manipulating multilinear formulas.

DEFINITION 2.6. **(Formula Multilinear Calculus (fMC)).** *Fix a field $\mathbb{F}$ and let $Q := \{Q_1, \ldots, Q_m\}$ be a collection of multilinear polynomials from $\mathbb{F}[x_1, \ldots, x_n, \bar{x}_1, \ldots, \bar{x}_n]$ (the variables $\bar{x}_1, \ldots, \bar{x}_n$ are treated as formal variables). Call the set of polynomials consisting of $x_i + \bar{x}_i - 1$ and $x_i \cdot \bar{x}_i$ for $1 \leq i \leq n$, the* Boolean axioms of fMC*.*

An fMC proof from Q of a polynomial g *is a finite sequence $\pi = (p_1, ..., p_\ell)$ of multilinear polynomials from $\mathbb{F}[x_1, \ldots, x_n, \bar{x}_1, \ldots, \bar{x}_n]$, such that $p_\ell = g$ and for each $i \in [\ell]$, either $p_i = Q_j$ for some $j \in [m]$, or p_i is a Boolean axiom of* fMC*, or p_i was deduced by one of the following inference rules using p_j, p_k for $j, k < i$:*

Product: *from p deduce $q \cdot p$, for some polynomial q in $\mathbb{F}[x_1, \ldots, x_n, \bar{x}_1, \ldots, \bar{x}_n]$ such that $p \cdot q$ is* multilinear*;*

Addition: *from p, q deduce $\alpha \cdot p + \beta \cdot q$, for some $\alpha, \beta \in \mathbb{F}$.*

All the polynomials in an fMC *proof are represented as multilinear formulas. (A polynomial* p_i *in an* fMC *proof is interpreted as the polynomial equation* $p_i = 0$*.) An* fMC *refutation of* Q *is a proof of* 1 *(which is interpreted as* $1 = 0$*) from* Q*.*

The *size* of an fMC proof π is defined as the total sum of all the formula sizes in π and is denoted by $|\pi|$.

Note that the Boolean axioms have only $0, 1$ solutions, where $\bar{x}_i = 0$ if $x_i = 1$ and $\bar{x}_i = 1$ if $x_i = 0$, for each $1 \leq i \leq n$.

Remark. The product inference rule of fMC in Definition 2.6 allows a polynomial p to be multiplied by an arbitrary polynomial q as long as $p \cdot q$ is multilinear. We could have restricted this product rule to allow a polynomial p to be multiplied only by a *variable* from $\{x_1, \ldots, x_n, \bar{x}_1, \ldots, \bar{x}_n\}$ (not occurring already in p). It is not hard to show that fMC refutations with such restricted product rule can polynomially simulate fMC refutations as defined in Definition 2.6.

Definition 2.7. **(Depth-k Formula Multilinear Calculus (depth-k fMC)).** *For a natural number* k*,* depth-k fMC *denotes a restriction of the* fMC *proof system, in which proofs consist of multilinear polynomials from* $\mathbb{F}[x_1, \ldots, x_n, \bar{x}_1, \ldots, \bar{x}_n]$ *represented as multilinear formulas of depth at most* k*.*

In order to refute an unsatisfiable CNF formula in fMC, we first translate the CNF formula into a system of polynomials via the translation scheme in Definition 2.5. Note that this translation scheme yields a set of *multilinear monomials*, since each literal occurs at most once inside a clause. From now on, we shall assume that any CNF formula is translated to a system of polynomials via Definition 2.5.

2.3.5. A discussion about the fMC proof system. It is important to clarify the following matter. A proof in fMC, as defined in Definition 2.6, is a sequence of formal (multilinear) polynomials, that is, (multilinear) elements of $\mathbb{F}[x_1, \ldots, x_n, \bar{x}_1, \ldots, \bar{x}_n]$. The *representation* of multilinear polynomials inside an fMC proof sequence is done by *arbitrary multilinear formulas*. Thus, each polynomial in an fMC proof can be represented in more than one way by a multilinear formula (in contrast to PC and PCR proofs, where each polynomial has a unique representation as a sum of monomials (disregarding the order of monomials inside a polynomial)). This means that we can think of the inference of new polynomials from previous ones, via the fMC inference rules, as a *semantic inference* of polynomials from preceding ones, rather than a *syntactic inference* of formulas from preceding formulas (the inference is semantic in the sense that any root of p in $\mathbb{F}$ is also a root of $q \cdot p$ in $\mathbb{F}$; and any common root of p and q in $\mathbb{F}$ is also a root of $\alpha \cdot p + \beta \cdot q$, for any $\alpha, \beta \in \mathbb{F}$).

Accordingly, when we talk about the *size* of an fMC proof (or refutation), we take into account a specific choice of multilinear formulas representing each of the polynomials in the proof sequence (naturally, we shall be interested in the most efficient way to represent each multilinear polynomial by a multilinear formula).

It stems from the aforesaid, that fMC is not necessarily a propositional proof system *in the formal sense*. Formally, a *propositional proof system* is defined to be a polynomial-time algorithm A that receives a Boolean formula F (usually a CNF) and a string π over some finite alphabet ("the (proposed) refutation of F"), such that there exists a π with $A(F, \pi) = 1$ if and only if F is unsatisfiable (cf. Cook & Reckhow (1979)). The reason that fMC is not necessarily a propositional

proof system in the formal sense is that it is an open question whether there exists a polynomial-time algorithm that can decide the identity of two (multilinear) arithmetic formulas. Hence, it is open whether there exists a polynomial-time algorithm that can verify the correctness of a given refutation, represented as a sequence of multilinear formulas.

Nevertheless, it is known that there is a *probabilistic* polynomial-time algorithm that can verify the identity of two given arithmetic formulas (cf. Schwartz (1980); Zippel (1979)). Thus, any proof of fMC can be checked in polynomial-time (in the proof size) by a probabilistic algorithm (cf. Pitassi (1997) for some facts about algebraic proof systems over arithmetic circuits and formulas). Also, it is worth noting that for some restricted classes of arithmetic formulas (and circuits) there are known *deterministic* polynomial-time algorithms that decide the identity of any two given arithmetic formulas (and circuits) belonging to the prescribed classes (see for example Raz & Shpilka (2004), Dvir & Shpilka (2005), Kayal & Saxena (2006)).

Grigoriev & Hirsch (2003) introduced algebraic proof systems over (general) arithmetic formulas that *are* propositional proof systems in the above formal sense. This was done by augmenting the system with, so-called, primitive rules that help demonstrating that two terms represent the same polynomial (the primitive rules express associativity, commutativity, distributivity, etc.).

2.4. Resolution and bounded-depth Frege proof systems.

Resolution. Resolution is a complete and sound proof system for unsatisfiable CNF formulas. For two clauses C and D, the *resolution rule* allows to derive $C \vee D$ from $C \vee x_i$ and $D \vee \neg x_i$. The clause $C \vee D$ is called the *resolvent* of the clauses $C \vee x_i$ and $D \vee \neg x_i$ on the variable x_i.

DEFINITION 2.8. *A* Resolution refutation *for a CNF formula F is a sequence of clauses $C_1, C_2, \ldots, C_\ell$, such that: (1) Each clause C_j is either a clause of F or a resolvent of two previous clauses in the sequence; (2) The last clause, C_ℓ, is the empty clause (which stands for* FALSE*, that is, the empty clause has no satisfying assignments). The size of a Resolution refutation is the total size of the clauses in it.*

Without loss of generality, we assume that no clause in a Resolution refutation contains both x_i and $\neg x_i$ (such a clause is always satisfied and hence it can be removed from the proof).

Bounded-depth Frege. We shall not need an explicit definition for the bounded-depth Frege proof system in this paper. We only state several exponential gaps between multilinear refutations and bounded depth Frege refutations for specific (families of) CNF formulas (based on known exponential lower bounds on the sizes of bounded-depth Frege refutations of these CNF formulas). For a formal definition of bounded-depth Frege see, e.g., Ben-Sasson (2002).

A Frege proof system is an implicationally complete proof system (meaning that given any set of propositional formulas T, every formula that is semantically implied from T has a proof from T in the system) whose proof lines consist of formulas over some finite complete set of connectives (a complete set of connectives is one that can represent any Boolean function; usually the connectives $\wedge, \vee, \neg$, which stand for AND, OR, NOT respectively, are used, augmented with the constant FALSE). A Frege proof system is specified by a finite set of sound and complete inference rules, rules for deriving new propositional formulas from existing ones by (consistent) substitution of formulas for variables in the rules.

A *bounded-depth* Frege proof system is a Frege proof system whose proof lines consist of constant-depth formulas, for some fixed constant (in the case of constant-depth formulas, the connectives $\wedge, \vee$ have unbounded fan-in). As mentioned above (Section 1.2), we can consider bounded-depth Frege to be a proof system for the set of *unsatisfiable* Boolean formulas, by treating a proof sequence (starting from some initial set of unsatisfiable formulas) that ends with FALSE, as a refutation.

2.5. Polynomial simulations. When comparing the strength of different proof systems, we shall restrict ourselves to CNF formulas only. That is, we consider propositional proof systems such as Resolution and bounded-depth Frege as proof systems for the set of unsatisfiable CNF formulas and we consider algebraic proof systems to be proof systems for the set of polynomial translations (defined above) of unsatisfiable CNF formulas.

DEFINITION 2.9. *Let P_1, P_2 be two proof systems for the set of unsatisfiable CNF formulas. We say that* P_2 polynomially simulates P_1 *if given a P_1 refutation π of a CNF F, there exists a refutation of F in P_2 of size polynomial in the size of π. Given an unsatisfiable CNF formula F, we say that* P_2 has an exponential gap over P_1 for F, *if there exists a polynomial size P_2 refutation of F, and the smallest P_1 refutation of F is of exponential size. If either P_1 or P_2 are* algebraic *proof systems, then we identify the CNF formula F with its translation to system of polynomial equations.*

For the sake of convenience we shall sometimes write simply *simulates* to mean *polynomially simulates*. Since we do not talk about other concepts of simulations, there should be no confusion.

3. Basic Manipulations of Arithmetic Formulas

In this section we shall prove simple propositions concerning manipulations of arithmetic formulas in fMC. These propositions will be very useful in the sequel. In particular, we take special care to maintain the depth of arithmetic formulas inside fMC refutations small (this makes the arguments of this section and Section 5 a bit tedious).

Notational convention. We say that a polynomial f_1 is *subtracted from* f_2 in an fMC proof, if f_2 is added to $-1 \cdot f_1$ by the addition rule.

Also, recall that a constant from the field (e.g., -1) can label an *edge* in an arithmetic formula, which means that the polynomial computed at the tail of the edge (i.e., the node where the edge is directed from) is multiplied by this constant.

PROPOSITION 3.1. *Let Φ be a multilinear formula whose root is a plus gate. Let $x \in \{x_1, \ldots, x_n, \bar{x}_1, \ldots, \bar{x}_n\}$ be some variable. Then there exists a multilinear formula $\Phi' := x \times \Phi_1 + \Phi_2$, where x does not occur in Φ_1, Φ_2, such that:*

(i) *Φ' and Φ compute the same polynomial;*
(ii) *$|\Phi_1| = 2 \cdot |\Phi|$ and $|\Phi_2| = |\Phi|$;*
(iii) *$\mathrm{dp}(\Phi_1) = \mathrm{dp}(\Phi_2) = \mathrm{dp}(\Phi)$;*
(iv) *The roots of both Φ_1 and Φ_2 are plus gates.*

PROOF. Let a be an element of the base field. Denote by $\Phi[a/x]$ the formula that results by substituting each occurrence of x in Φ with a.

Let f be the polynomial computed by Φ. Consider the polynomial f as a polynomial in the variable x only, denoted by $f(x)$ (where now the coefficients of the variable x in $f(x)$ also contain

variables). Since f is multilinear, $f(x)$ is of degree 1. Thus, by the Lagrange interpolation formula, the following is equal to the polynomial $f(x)$:

$$x \cdot (f(1) - f(0)) + f(0)$$

(this equality can also be verified in a straightforward manner). Hence, the multilinear formula

$$x \times (\Phi[1/x] - \Phi[0/x]) + \Phi[0/x]$$

computes the polynomial $f(x)$. When considering $f(x)$ as a polynomial in all the variables occurring in Φ, $f(x)$ is precisely the polynomial f.

Therefore, letting $\Phi_1 := \Phi[1/x] - \Phi[0/x]$ and $\Phi_2 := \Phi[0/x]$ concludes the proof (note that Φ_1 is of the same depth as that of Φ, since the root of Φ is a plus gate, and since subtraction can be achieved by labelling the *edge* going out of the root of $\Phi[0/x]$ with -1). (Also notice that if x does not occur in Φ then the proof holds trivially, since $\Phi_1 := \Phi[1/x] - \Phi[0/x] = \Phi - \Phi$, which is a formula computing the zero polynomial.) □

PROPOSITION 3.2. *Let Φ be a multilinear formula of depth d, whose root is a plus gate, and let $x_i \in \{x_1, \ldots, x_n\}$ be some variable. Then there is a multilinear formula*

$$\bar{x}_i \times x_i \times \varphi_1 + \bar{x}_i \times \varphi_2 + x_i \times \varphi_3 + \varphi_4$$

that computes the same polynomial as Φ, and such that for all $1 \leq j \leq 4$:

(i) *φ_j does not contain $x_i, \bar{x}_i$;*
(ii) *φ_j has depth at most d and size $O(|\Phi|)$;*
(iii) *φ_j has a plus gate at the root.*

PROOF. We simply apply Proposition 3.1 three times. Specifically, by Proposition 3.1, there are two depth d and size $O(|\Phi|)$ multilinear formulas Φ_1, Φ_2 that do not contain $\bar{x}_i$, such that:

$$\Phi = \bar{x}_i \cdot \Phi_1 + \Phi_2 .$$

By Claim 3.1 again, there exist four depth d multilinear formulas $\varphi_1, \varphi_2, \varphi_3, \varphi_4$ that do not contain $x_i, \bar{x}_i$, where each formula is of size $O(|\Phi|)$, and has a plus gate at the root, such that:

$$\begin{aligned}
&\bar{x}_i \cdot \Phi_1 + \Phi_2 = \bar{x}_i \cdot (x_i \cdot \varphi_1 + \varphi_2) + \Phi_2 && \text{apply Claim 3.1 on } \Phi_1, x_i \\
&= \bar{x}_i \cdot x_i \cdot \varphi_1 + \bar{x}_i \cdot \varphi_2 + \Phi_2 \\
&= \bar{x}_i \cdot x_i \cdot \varphi_1 + \bar{x}_i \cdot \varphi_2 + x_i \cdot \varphi_3 + \varphi_4 && \text{apply Claim 3.1 on } \Phi_2, x_i
\end{aligned}$$

(we treat here all formulas as the polynomials they compute; so the equalities are between polynomials, and not formulas). □

We need the following claim for the proposition that follows.

CLAIM 3.3. *Let Φ_1 be a depth $d \geq 2$ multilinear formula computing the polynomial f. Let Φ_2 be a multilinear formula for a monomial M (i.e., either a depth 1 multilinear formula having a product gate at the root, or a single variable, or an element of the field). Assume that no variable that occurs in Φ_2 also occurs in Φ_1. Then there is a multilinear formula for $f \cdot M$, with the same gate at the root as that of Φ_1, depth d and size $O(|\Phi_1| \cdot |\Phi_2|)$.*

PROOF. The claim holds simply by distributivity of multiplication over addition.

If Φ_1 has a product gate at the root then $\Phi_1 \times \Phi_2$ is the desired multilinear formula (note this formula is of depth d).

Assume that Φ_1 has a plus gate at its root.

Recall that we consider all formulas to be leveled. Thus, for some m, there exist m multilinear formulas $\varphi_1, \ldots, \varphi_m$, each either has an (unbounded fan-in) $\times$ gate at the root and depth $\leq d-1$, or has depth 0 (i.e., is an input variable or a field element), such that $\Phi_1 = \varphi_1 + \ldots + \varphi_m$. We can assume w.l.o.g. that $\mathrm{dp}(\Phi_2) = 1$ (otherwise, we consider Φ_2 to be the formula $\Phi_2 \times 1$). For all $1 \leq i \leq m$, $\mathrm{dp}(\Phi_2 \times \varphi_i) \leq d-1$. Thus, by distributivity of multiplication over addition, the formula

$$\Phi_2 \times \varphi_1 + \ldots + \Phi_2 \times \varphi_m \tag{3.4}$$

computes the polynomial $f \cdot M$ and has size $O(|\Phi_1| \cdot |\Phi_2|)$ and depth d. Since, by assumption, no variable that occurs in Φ_2 also occurs in Φ_1, (3.4) is a *multilinear* formula. □

PROPOSITION 3.5. *Let $\Phi = \Phi_1 + \ldots + \Phi_k$ be a multilinear formula of depth d. Let $\varphi_1, \ldots, \varphi_k$ be k formulas, where each φ_i is a multilinear formula of size $\leq s$ for a monomial (i.e., φ_i is either a depth 1 multilinear formula having a product gate at the root, or a single variable, or an element of the field). Denote by f the polynomial computed by $\varphi_1 \times \Phi_1 + \ldots + \varphi_k \times \Phi_k$, and assume that no variable that occurs in φ_i also occurs in Φ_i (for all $1 \leq i \leq k$). Then f has a multilinear formula of size $O(s \cdot |\Phi|)$ and depth $\max\{d, 2\}$.*

PROOF. We show that for all $1 \leq i \leq k$, there exists a multilinear formula Φ'_i of size $O(s \cdot |\Phi_i|)$ that computes the polynomial computed by $\Phi_i \times \varphi_i$, and such that one of the following holds:

(i) $\mathrm{dp}(\Phi'_i) = \mathrm{dp}(\Phi_i)$ and the gate at the root of Φ'_i is the same as that of Φ_i;

(ii) $\mathrm{dp}(\Phi'_i) = 2$ and the root of Φ'_i is a plus gate.

Therefore, the multilinear formula $\Phi'_1 + \ldots + \Phi'_k$ computes the polynomial f, and has depth $\max\{d, 2\}$ and size $O(\sum_{i=1}^{k} |\Phi_i| \cdot s) = O(s \cdot |\Phi|)$.

Case 1: Assume that $\mathrm{dp}(\Phi_i) \geq 2$, for some $1 \leq i \leq k$. Then by Claim 3.3, the polynomial computed by $\Phi_i \times \varphi_i$ has a multilinear formula Φ'_i of depth $\mathrm{dp}(\Phi_i)$ and size $O(s \cdot |\Phi_i|)$, with the same gate at the root as Φ_i.

Case 2: Assume that $\mathrm{dp}(\Phi_i) < 2$, for some $1 \leq i \leq k$. Then we can switch to a new formula Φ''_i that computes the same polynomial as Φ_i, such that $\mathrm{dp}(\Phi''_i) = 2$ and Φ''_i has a plus gate at the root[4]. Thus, we can apply Case 1 on Φ''_i. □

[4] If $\Phi_i = a$, for a a variables or a field element, then switch to $a \times 1 + 0$. If Φ_i has a product gate at the root, then switch to $\Phi_i + 0$. If Φ_i is a sum of variables (and/or field elements) with constant coefficients, $\alpha_1 x_{i_1} + \ldots + \alpha_m x_{i_m}$, then switch to $(\alpha_1 x_{i_1} \times 1) + \ldots + \alpha_m x_{i_m}$.

PROPOSITION 3.6. *Let $\Phi_2 + \Phi_1$ be a multilinear formula of depth d, where Φ_2 computes the polynomial f_2 (possibly the zero polynomial) and Φ_1 computes the polynomial f_1. Assume that Φ_1 contains neither the variable x_i nor $\bar{x}_i$. Let $d' := \max\{d, 2\}$. Then the polynomials $f_2 + f_1 \cdot \bar{x}_i$ and $f_2 + f_1 \cdot (1 - x_i)$ can be proved from one another in depth-d'* fMC *proofs of size $O(|\Phi_2| + |\Phi_1|)$.*

PROOF. Start from the Boolean axiom $x_i + \bar{x}_i - 1$, and multiply it by f_1 in order to get:

$$(x_i + \bar{x}_i - 1) \cdot f_1 \,. \tag{3.7}$$

If we subtract (3.7) from

$$f_2 + f_1 \cdot \bar{x}_i \,, \tag{3.8}$$

we get

$$f_2 + f_1 \cdot (1 - x_i) \,. \tag{3.9}$$

Similarly, if we add (3.7) to (3.9), we get (3.8).

Open parentheses in (3.7), (3.9) and (3.8), and observe that by Proposition 3.5 these three polynomials have all multilinear formulas of depth at most d' and size $O(|\Phi_2| + |\Phi_1|)$. □

4. Soundness and Completeness of fMC

We show in this section that fMC is a sound proof system. Then we show a simple completeness proof, by demonstrating a depth-2 fMC simulation of Resolution. In fact, this simulation also stems from the simulation of PCR by depth-2 fMC (Section 5), since PCR simulates Resolution.

PROPOSITION 4.1. fMC *is a sound proof system. That is, if there exists an* fMC *refutation of a system of multilinear polynomials Q over a field $\mathbb{F}$ then the system of multilinear polynomials Q has no common root in $\mathbb{F}$ with $0, 1$ values.*

PROOF. Note that the inference rules of fMC are sound: Any root of p in $\mathbb{F}$ is also a root of $q \cdot p$ in $\mathbb{F}$; and any common root of p and q in $\mathbb{F}$ is also a root of $\alpha \cdot p + \beta \cdot q$ (for any $\alpha, \beta \in \mathbb{F}$).

Let $\pi = (p_1, \ldots, p_\ell)$ be an fMC refutation of Q. Any p_i in π is either a Boolean axiom or a polynomial from Q or p_i was deduced from previous polynomials in π by one of the inference rules.

Then, by the soundness of the inference rules, any common root of the system Q that also satisfies the Boolean axioms in $\mathbb{F}$, is also a root of $p_j \in \pi$, for all $j \leq \ell$ (by induction on the refutation length). Since by definition, the last polynomial in π (i.e., p_ℓ) is 1, then there exists no common root of the system Q and the Boolean axioms in $\mathbb{F}$. This means that Q has no common root in $\mathbb{F}$ with $0, 1$ values. □

We show now that fMC is complete for (polynomial translations of) CNF formulas, when the lines of the refutations consist of depth 2 multilinear formulas. Note that any CNF formula translates via Definition 2.5 into a system of multilinear monomials. In particular, we prove that any Resolution refutation of an unsatisfiable CNF can be transformed into a depth-2 fMC refutation of (the polynomial translation of) that CNF, with at most a polynomial increase in size.

PROPOSITION 4.2. *Depth-2 fMC polynomially simulates Resolution.*

PROOF. Let π be a Resolution refutation of some CNF formula. By induction on the number of clauses in π, we show how to translate each step in π into a depth-2 fMC proof, with at most a polynomial increase in size.

Recall that a clause is a *set* of literals, hence, each literal occurs only once in a clause. The base case are the initial clauses, which translate into multilinear monomials via Definition 2.5.

Let C, D be two clauses such that $C \vee D$ is the resolvent of $C \vee x_i$ and $D \vee \neg x_i$ on the variable x_i. Denote by E the clause containing the common literals of C and D. Thus, there exist two clauses A, B having no common literals such that $C = A \vee E$ and $D = B \vee E$. By definition of the resolution rule, A, B, E do not contain the variable x_i (recall that we assume without loss of generality that no clause in a Resolution refutation contains both x_i and $\neg x_i$ and we also delete multiple occurrences of the same literal in a clause, and so C, D contains neither x_i nor $\neg x_i$).

For a given clause K, denote by q_K the polynomial translation of K (via Definition 2.5). By induction hypothesis we have already the multilinear monomials $q_{A \vee E \vee x_i} = q_A \cdot q_E \cdot \bar{x}_i$ and $q_{B \vee E \vee \neg x_i} = q_B \cdot q_E \cdot x_i$. We need to derive the monomial $q_{C \vee D} = q_{A \vee B \vee E} = q_A \cdot q_B \cdot q_E$ with an fMC proof of size polynomial in the sizes (i.e., number of literals) of A, B, E.

By Proposition 3.6 we can prove from $q_A \cdot q_E \cdot \bar{x}_i$ the multilinear polynomial $q_A \cdot q_E \cdot (1 - x_i)$, with a polynomial-size depth-2 fMC proof. Now, multiply $q_A \cdot q_E \cdot (1 - x_i)$ by q_B. Since the literals in A, B and E are pairwise disjoint, we get a multilinear polynomial $q_A \cdot q_E \cdot q_B \cdot (1 - x_i)$.

The polynomial $q_B \cdot q_E \cdot x_i$ is multiplied by q_A, which yields the multilinear polynomial $q_A \cdot q_E \cdot q_B \cdot x_i$. Adding $q_A \cdot q_E \cdot q_B \cdot (1 - x_i)$ and $q_A \cdot q_E \cdot q_B \cdot x_i$ we arrive at $q_A \cdot q_E \cdot q_B$.

Notice that it is possible to represent each arithmetic formula in the simulation with a depth 2 formula (i.e., as a sum of monomials). □

5. Simulation Results

In this section we prove a general simulation result for fMC (Theorem 5.1). Specifically, we show the following: Let π be a PCR refutation of some initial collection of multilinear polynomials Q over some fixed field. Assume that π has polynomially many steps (i.e., the number of proof lines in the PCR proof sequence is polynomial). If the 'multilinearization' (i.e., the result of applying the $\mathbf{M}[\cdot]$ operator – see Definition 2.2) of each of the polynomials in π has a polynomial-size depth d multilinear formula (with a plus gate at the root), then there is a polynomial-size depth-d fMC refutation of Q. (Note that we only require that the number of steps in π is polynomial. The *size* (i.e., the total number of monomials) of the PCR proof might not be polynomially-bounded.)

A simple consequence of the simulation result is that any PC and PCR refutations of a set of initial multilinear polynomials over some fixed field can be simulated by a depth-2 fMC refutation. Since CNF formulas are translated into sets of multilinear polynomials (via Definition 2.5), this shows that with respect to (translations of) CNF formulas, depth-2 fMC is at least as strong as PC and PCR.

Another merit of the simulation result is that it can help in proving upper bounds for fMC refutations. In particular, we shall use it in proving the upper bound for the Functional Pigeonhole Principle in Section 7.

THEOREM 5.1. *Fix a field $\mathbb{F}$ and let Q be a set of multilinear polynomials from $\mathbb{F}[x_1, \ldots, x_n, \bar{x}_1, \ldots, \bar{x}_n]$. Let $\pi = (p_1, \ldots, p_m)$ be a* PCR *refutation of Q. For each $p_i \in \pi$, let*

Φ_i be a multilinear formula for the polynomial $\mathbf{M}[p_i]$. Let S be the total size of all formulas Φ_i, i.e., $S = \Sigma_{i=1}^{m}|\Phi_i|$, and let $d \geq 2$ be the maximal depth of all formulas Φ_i. Assume that the depth of all the formulas Φ_i that have a product gate at the root is at most $d-1$. Then there is a depth-d fMC *refutation of Q of size polynomial in S.*

COROLLARY 5.2. *Depth-*2 fMC *polynomially simulates* PC *and* PCR*.*

PROOF. Since PCR obviously simulates PC it is sufficient to consider only PCR proofs. Recall that the size of a PCR proof is the total number of monomials in it. Note also that given any multivariate polynomial q, the total number of monomials in q is greater or equal than the total number of monomials in $\mathbf{M}[q]$.

Given a PCR proof $\pi := (p_1, \ldots, p_m)$, represent each multilinear polynomial $\mathbf{M}[p_i]$ as a sum of monomials, and denote this multilinear formula by Φ_i. Each Φ_i is of depth at most 2.

Let $|\pi|$ denote the size of the PCR proof π, i.e., the number of monomials in π, and let $\ell \leq n$ be the total number of variables that appear in the polynomials in π. In light of Theorem 5.1, we need to show that the total size of all the formulas Φ_i is polynomial in $|\pi|$. Since $|\pi| \geq \ell$, it suffices to show that the total size of all the formulas Φ_i (for $1 \leq i \leq m$) is $O(\ell \cdot |\pi|)$.

Since each Φ_i is a sum of (multilinear) monomials then the total size of all the formulas Φ_i is just the total size of all the monomials occurring in $\Phi_1, \ldots, \Phi_m$ (ignoring constant factors). Each multilinear monomial in $\Phi_1, \ldots, \Phi_m$ is of size $O(\ell)$. Thus (by the first paragraph of this proof), the total size of all the formulas Φ_i is $O(\ell \cdot |\pi|)$. □

Proof of Theorem 5.1. Denote by U the sequence of multilinear polynomials $\mathbf{M}[p_1], \ldots, \mathbf{M}[p_m]$. Suppose that π contains an instance of the PCR product rule: from p_i deduce $x \cdot p_i$, for some $x \in \{x_1, \ldots, x_n, \bar{x}_1, \ldots, \bar{x}_n\}$. Then U contains the polynomials $\mathbf{M}[p_i]$ and $\mathbf{M}[x \cdot p_i]$. Note that $\mathbf{M}[x \cdot p_i]$ does not necessarily equal $x \cdot \mathbf{M}[p_i]$. Thus, an instance of a PCR product rule in π does not necessarily turn into an instance of an fMC product rule in U. This means that the sequence U does not necessarily form a legitimate fMC proof sequence.

Nevertheless, with at most a polynomial increase in size, it is possible to turn U into a depth-d fMC proof. That is, we build from the sequence U a depth-d fMC refutation of Q, denoted π'. The size of π' will be polynomial in the total size of all formulas in U. This is done by adding to U new depth-d fMC proof sequences that simulate all instances of the PCR product rule occurring in π (that is, depth-d fMC proof sequences of $\mathbf{M}[x \cdot p_i]$ from $\mathbf{M}[p_i]$, according to the notations of the previous paragraph).

Claim 5.3 and Lemma 5.4, that follow, illustrate how to build a depth-preserving small multilinear proofs of $\mathbf{M}[x \cdot p_i]$ from $\mathbf{M}[p_i]$.

CLAIM 5.3. *Let Φ_1 and Φ_2 be two multilinear formulas with a plus gate at the root for the polynomials f_1, f_2, respectively. Assume that both Φ_1 and Φ_2 contain neither x_i nor $\bar{x}_i$. Let $d := \max\{\mathrm{dp}(\Phi_1), \mathrm{dp}(\Phi_2), 2\}$. Then there is a depth-$d$* fMC *proof of $x_i \cdot f_1 + x_i \cdot f_2$ from $x_i \cdot f_1 + f_2$ with size $O(|\Phi_1| + |\Phi_2|)$.*

PROOF. Apply the following fMC proof sequence:

1. $x_i \cdot f_1 + f_2$	hypothesis
2. $(1 - \bar{x}_i) \cdot (x_i \cdot f_1 + f_2)$	product of (1)
3. $x_i \cdot \bar{x}_i$	Boolean axiom
4. $(x_i \cdot \bar{x}_i) \cdot f_1$	product of (3)
5. $(1 - \bar{x}_i) \cdot (x_i \cdot f_1 + f_2) + (x_i \cdot \bar{x}_i) \cdot f_1$	(2) plus (4)
6. $x_i + \bar{x}_i - 1$	Boolean axiom
7. $(x_i + \bar{x}_i - 1) \cdot f_2$	product of (6)
8. $(1 - \bar{x}_i) \cdot (x_i \cdot f_1 + f_2) + (x_i \cdot \bar{x}_i) \cdot f_1 + (x_i + \bar{x}_i - 1) \cdot f_2$	(5) plus (7)

Note that the last line 8 is equal to $x_i \cdot f_1 + x_i \cdot f_2$.

We need to make sure that each polynomial in the above proof sequence has a depth d multilinear formula of size $O(|\Phi_1| + |\Phi_2|)$.

The polynomials in lines 3,6 can obviously be written as constant size depth 1 multilinear formulas.

Considering all other lines in the proof sequence; First open parentheses. We get a sum of constant number of terms, where each term is a product of f_1 or f_2 with a multilinear monomial (or a field element, e.g. -1). For example, line 2 equals: $x_i \cdot f_1 + f_2 - \bar{x}_i \cdot x_i \cdot f_1 - \bar{x}_i \cdot f_2$.

Thus, by Proposition 3.5, the polynomials in all the lines of the above proof sequence have depth d multilinear formulas of size $O(|\Phi_1| + |\Phi_2|)$. □

LEMMA 5.4. *Let p_i be a polynomial from $\mathbb{F}[x_1, \ldots, x_n, \bar{x}_1, \ldots, \bar{x}_n]$, and let $x \in \{x_1, \ldots, x_n, \bar{x}_1, \ldots, \bar{x}_n\}$. Let Φ be a multilinear formula for $\mathbf{M}[p_i]$ having a plus gate at the root and let $d := \max\{\mathrm{dp}(\Phi), 2\}$. Then there is a depth-$d$* fMC *proof of $\mathbf{M}[x \cdot p_i]$ from $\mathbf{M}[p_i]$, of size $O(|\Phi|)$.*

PROOF. We assume that $x = x_i$ for some $x_i \in \{x_1, \ldots, x_n\}$ (the case of $x \in \{\bar{x}_1, \ldots, \bar{x}_n\}$ is similar).

By Proposition 3.2, there are multilinear formulas $\varphi_1, \varphi_2, \varphi_3, \varphi_4$ such that

$$\mathbf{M}[p_i] = \bar{x}_i \cdot x_i \cdot \varphi_1 + \bar{x}_i \cdot \varphi_2 + x_i \cdot \varphi_3 + \varphi_4 \tag{5.5}$$

(the equality here is between polynomials), where for all $1 \leq j \leq 4$: (i) φ_j does not contain $x_i, \bar{x}_i$, and (ii) φ_j has depth at most d and size $O(|\Phi|)$, and (iii) φ_j has a plus gate at the root.

Multiply the Boolean axiom $\bar{x}_i \cdot x_i$ by φ_1, to get

$$\bar{x}_i \cdot x_i \cdot \varphi_1 . \tag{5.6}$$

Subtract (5.6) from (5.5). We arrive at the polynomial

$$\bar{x}_i \cdot \varphi_2 + x_i \cdot \varphi_3 + \varphi_4 . \tag{5.7}$$

By (i,ii,iii) above, both φ_1 and $\varphi_2 + \varphi_3 + \varphi_4$ have depth at most d and size $O(|\Phi|)$ multilinear formulas that do not contain $x_i, \bar{x}_i$. Therefore, by Proposition 3.5, both (5.6) and (5.7) have multilinear formulas of depth at most d and size $O(|\Phi|)$.

By Proposition 3.6, we can derive from (5.7), with a depth-d fMC proof of size $O(|\Phi|)$,

$$(1 - x_i) \cdot \varphi_2 + x_i \cdot \varphi_3 + \varphi_4 \,,$$

which is equal to

$$x_i \cdot (\varphi_3 - \varphi_2) + (\varphi_2 + \varphi_4) \,. \tag{5.8}$$

Since all formulas φ_j have plus gates at their root, then $(\varphi_3 - \varphi_2)$ and $(\varphi_2 + \varphi_4)$ have multilinear formulas of depth $\mathrm{dp}(\Phi)$ and size $O(|\Phi|)$. Thus, by Claim 5.3, there is a depth-d fMC proof of

$$x_i \cdot (\varphi_3 - \varphi_2) + x_i \cdot (\varphi_2 + \varphi_4) = x_i \cdot \varphi_3 + x_i \cdot \varphi_4 \,, \tag{5.9}$$

from (5.8), where the size of the proof is $O(|\Phi|)$.

Now, multiply the Boolean axiom $\bar{x}_i \cdot x_i$ by $(\varphi_1 + \varphi_2)$, and add the result to (5.9). We obtain

$$\bar{x}_i \cdot x_i \cdot \varphi_1 + \bar{x}_i \cdot x_i \cdot \varphi_2 + x_i \cdot \varphi_3 + x_i \cdot \varphi_4 = \mathbf{M}[x_i \cdot p_i] \,. \tag{5.10}$$

Similar to (5.7), polynomial (5.10) can be written as depth d multilinear formula of size polynomial in $O(|\Phi|)$. □

Concluding the proof of Theorem 5.1. Recall that U is the sequence of multilinear formulas $\Phi_1, \ldots, \Phi_m$ (corresponding to the polynomials $\mathbf{M}[p_1], \ldots, \mathbf{M}[p_m]$).

Let p_j, p_k (for $j < k \in [m]$) be some instance of the PCR product rule in π. That is, the polynomial $p_k = x \cdot p_j$ is deduced from p_j, for some $x \in \{x_1, \ldots, x_n, \bar{x}_1, \ldots, \bar{x}_n\}$. We can assume that both Φ_j and Φ_k have plus gates at their root, and so by assumption both have depth at most d (if Φ_ℓ, for $\ell \in \{j, k\}$, has a product gate at the root, then by assumption the depth of the formula is at most $d-1$; hence, we can let Φ_ℓ be the formula $\Phi_\ell + 0$). Thus, by Lemma 5.4 there is a depth-d fMC proof of $\mathbf{M}[p_k] = \mathbf{M}[x \cdot p_j]$ from $\mathbf{M}[p_j]$ of size $O(|\Phi_j|)$. We denote this proof (sequence) by S_k. For all instances of the PCR product rule in π, replace the formula Φ_k in U with the proof sequence S_k, excluding the first formula of S_k (note that Φ_j the first formula of S_k, already appears in U). Let π' denote the new sequence of formulas obtained from U by this process.

Now, π' is easily seen to be a depth-d fMC refutation of Q: (i) π' ends with $\mathbf{M}[p_m] = \mathbf{M}[1] = 1$; (ii) Every arithmetic formula in π' is a multilinear formula of depth at most d; and (iii) For every formula Ψ_i in π', computing the polynomial q_i, either Ψ_i was added to π' as a formula in some proof sequence S_j (as defined above), or q_i is the result of applying $\mathbf{M}[\cdot]$ on some polynomial p_ℓ from π (that is, $q_i = \mathbf{M}[p_\ell]$ for some $\ell \in [m]$).

In the first case of (iii), Ψ_i is either an axiom of fMC, or a formula that was deduced by one of fMC's inference rules from preceding formulas in S_j.

In the second case of (iii), p_ℓ is either a (multilinear) polynomial from Q, or a Boolean axiom of PCR, or p_ℓ was deduced by one of the two PCR inference rules from some preceding polynomials in π. If p_ℓ is the Boolean axiom $x_j + \bar{x}_j - 1$ of PCR, for some $j \in [n]$, then $q_i = p_\ell$ is also a Boolean axiom of fMC. If p_ℓ is the Boolean axiom $x_j^2 - x_j$, for some $j \in [n]$, and so $q_i = 0$ (thus, q_i can be discarded from the proof; formally, the zero polynomial can be deduced from any polynomial, by the fMC inference rules).

In case p_ℓ was deduced by the PCR product rule from some preceding polynomial p_k ($k < \ell \in [m]$) in π, then by definition of S_ℓ, q_i stems from preceding polynomials in S_ℓ by an fMC inference rule. Moreover, instances of the PCR addition rule in π are transformed in π' into legal instances of the fMC addition rule, as $\mathbf{M}[\cdot]$ is easily seen to be a linear operator. □

6. Separations of Algebraic Proof Systems and Multilinear Circuit Lower Bounds

In this section we use Theorem 5.1 to link the separation of certain algebraic proof systems to the problem of proving multilinear arithmetic circuit lower bounds. Specifically, we show that if there is a set of multilinear polynomials Q, for which there exists an explicit polynomial-size refutation manipulating *general* arithmetic circuits (i.e., not necessarily multilinear), then proving a super-polynomial lower bound on the refutation size of Q in a proof system manipulating *multilinear* circuits implies a super-polynomial lower bound on the size of multilinear circuits computing an explicit polynomial (see Section 2.2.1 and Definition 2.1 for definitions of arithmetic circuits and multilinear circuits, respectively). In fact, this result can be generalized further (see remark after the proof of Theorem 6.4). We shall exploit the fact that we work with algebraic proof systems (like fMC) in which polynomials are represented by arbitrarily chosen arithmetic formulas (this fact enabled us to prove the general simulation result in Theorem 5.1).

The following defines algebraic proof systems that manipulate general and multilinear arithmetic circuits:

Definition 6.1. **(cMC, cPCR).**
(i) The cMC *(for Circuit Multilinear Calculus) proof system is identical to* fMC*, except that multilinear polynomials in* cMC *proof sequences are represented by multilinear circuits (instead of multilinear formulas). The* size *of a* cMC *proof π is defined to be the total sum of all the circuit-sizes in π.*
(ii) The cPCR *(for Circuit Polynomial Calculus with Resolution) proof system is identical to* PCR*, except that polynomials in* cPCR *proof sequences are represented by (general) arithmetic circuits (instead of explicit sums of monomials). The* size *of a* cPCR *proof π is defined to be the total sum of all the circuit-sizes in π.*

We now reiterate Theorem 5.1 where instead of dealing with depth d multilinear formulas we deal with multilinear circuits (not necessarily of constant-depth):

Proposition 6.2. *Fix a field $\mathbb{F}$ and let Q be an unsatisfiable set of multilinear polynomials from $\mathbb{F}[x_1,\ldots,x_n,\bar{x}_1,\ldots,\bar{x}_n]$. Let $\pi=(p_1,\ldots,p_m)$ be a* PCR *refutation of Q. For each $p_i\in\pi$ let Φ_i be a multilinear circuit for the polynomial $\mathbf{M}[p_i]$ and let S be the total size of all the multilinear circuits Φ_i. Then there is a* cMC *refutation of Q of size polynomial in S.*

Proof. First notice that any manipulation of arithmetic formulas presented in Section 3 is also applicable to multilinear circuits. Thus, by a straightforward inspection of the proof of Theorem 5.1, one can verify that when substituting the phrases 'depth d multilinear formulas' by 'multilinear circuits' and 'depth-d fMC' by 'cMC', Theorem 5.1 still holds. □

Corollary 6.3. *Fix a field $\mathbb{F}$ and let Q be an unsatisfiable set of multilinear polynomials from $\mathbb{F}[x_1,\ldots,x_n,\bar{x}_1,\ldots,\bar{x}_n]$. Let $\pi=(p_1,\ldots,p_m)$ be a* PCR *refutation of Q, with a polynomial (in n) number of steps.*[5] *Assume that every* cMC *refutation of Q is of super-polynomial size. Then there exists a polynomial $p_i\in\pi$ such that $\mathbf{M}[p_i]$ has no polynomial-size multilinear circuit.*

[5] Again (as in Section 5), we only require that the *number of steps* is polynomial. The *size* of the PCR proof might not be polynomially bounded.

PROOF. The statement follows immediately from Proposition 6.2, as if all polynomials $\mathbf{M}[p_i]$ (for all $1 \leq i \leq m$) had polynomial-size multilinear circuits, there would have been also a polynomial-size cMC refutation of Q, which contradicts the assumption. □

Recall from Definition 2.9 that a proof system P_1 has a super-polynomial (resp., exponential) gap over a proof system P_2 for a family $Q := \{Q_n\}_{n\in\mathbb{N}}$ of unsatisfiable sets of polynomials over a field, if there exist polynomial-size P_1 refutations of Q and every P_2 refutation of Q is of super-polynomial (resp., exponential) size. In this case we shall also say that Q *super-polynomially (resp., exponentially) separates* P_1 *from* P_2. In case we also have *explicit* polynomial-size proofs of Q in P_1 then we shall say that we have an *explicit super-polynomial (resp., exponential) separation of* P_1 *from* P_2 *for* Q. The term *explicit* here means that there is a Turing machine that for any given input-size n (given to the machine in unary representation) outputs the proof of Q_n in P_1 and runs in time polynomial in the size of the proof (similarly, we can speak about an explicit (family of) polynomials).

THEOREM 6.4. *Fix a field* $\mathbb{F}$ *and let* Q *be an unsatisfiable set of multilinear polynomials from* $\mathbb{F}[x_1, \ldots, x_n, \bar{x}_1, \ldots, \bar{x}_n]$. *Assume that there is an explicit super-polynomial (resp., exponential) size separation of* cPCR *from* cMC *for* Q. *Then there exists an explicit multilinear polynomial* g *with no polynomial-size (resp., sub-exponential size) multilinear circuit.*

PROOF. Let $(\Theta_1, \ldots, \Theta_m)$ be the explicit polynomial-size cPCR proof sequence of Q (every Θ_i is an arithmetic circuit). For all $1 \leq i \leq m$ let p_i be the polynomial computed by Θ_i (in other words, we can view $(p_1, \ldots, p_m)$ as a PCR refutation of Q). By assumption m is polynomially bounded by n and any cMC refutation of Q (of any depth) is of super-polynomial size (resp., exponential size) in n. Thus, by Corollary 6.3 there exists an $1 \leq i \leq m$ such that $\mathbf{M}[p_i]$ has no polynomial-size (resp., sub-exponential size) multilinear circuit.

Let $z_1, \ldots, z_m$ be new variables and consider the polynomial $g := \sum_{j=1}^{m} z_j \cdot \mathbf{M}[p_j]$. Then g has no polynomial-size in n (resp., sub-exponential size in n) multilinear circuit (over $\mathbb{F}$) (as if there was such a multilinear circuit Ψ computing g, we could have obtained a polynomial-size in n (resp., sub-exponential size in n) multilinear circuit for $\mathbf{M}[p_i]$ by substituting every occurrence of z_j in Ψ, for $j \neq i$, by 0, and every occurrence of z_i in Ψ by 1). (Note that the number of variables in g is polynomially bounded by n, since m is polynomially bounded by n.) □

Remark. Theorem 6.4 can be generalized further for any (reasonably defined) pair of arithmetic circuit-classes $\mathcal{C}_1, \mathcal{C}_2$ that are at least as strong as multilinear formulas (that is, when considering $\mathcal{C}_1, \mathcal{C}_2$ instead of arithmetic circuits and multilinear circuits, respectively).

Specifically, denote by $\mathcal{C}_1$PCR the proof system that is similar to PCR, where polynomials are represented by $\mathcal{C}_1$-circuits (instead of explicit sums of monomials); and denote by $\mathcal{C}_2$MC the proof system that is similar to fMC, where multilinear polynomials are represented by $\mathcal{C}_2$-circuits (instead of multilinear formulas). It is not hard to see that if $\mathcal{C}_2$ is any (reasonably defined) arithmetic circuit-class that contains the class of multilinear formulas, then Proposition 6.2 is still valid when one considers $\mathcal{C}_2$-circuits and $\mathcal{C}_2$MC refutations instead of multilinear circuits and cMC refutations, respectively.

Thus, the same reasoning that was described above (i.e., in Corollary 6.3 and Theorem 6.4) implies that if there is an explicit super-polynomial (resp., exponential) size separation of $\mathcal{C}_1$PCR from $\mathcal{C}_2$MC for some unsatisfiable set of multilinear polynomials Q, then there exists an explicit multilinear polynomial g with no polynomial-size (resp., sub-exponential size) $\mathcal{C}_2$-circuit.

(In a similar manner, we can speak of $\mathcal{C}_1$PCR versus $\mathcal{C}_2$PCR refutations instead of $\mathcal{C}_1$PCR versus $\mathcal{C}_2$MC refutations. We thus obtain that an explicit super-polynomial (resp., exponential) size separation of $\mathcal{C}_1$PCR from $\mathcal{C}_2$PCR for some unsatisfiable set of polynomials Q implies the existence of an explicit polynomial g (*not necessarily multilinear*) with no polynomial-size (resp., sub-exponential size) $\mathcal{C}_2$-circuit.)

7. The Functional Pigeonhole Principle

In this section we show that there is a polynomial size depth-3 fMC refutation of the Functional Pigeonhole Principle over fields of characteristic 0.

The Functional Pigeonhole Principle FPHP_n^m with m pigeons and $n < m$ holes states that there is no injection from m pigeons to n holes. As a propositional formula it is usually formulated as follows:

$$\left(\bigwedge_{i\in[m]} \bigvee_{k\in[n]} x_{i,k} \wedge \bigwedge_{i\in[m]} \bigwedge_{k<l\in[n]} (\neg x_{i,k} \vee \neg x_{i,\ell}) \right) \longrightarrow \bigvee_{i<j\in[m]} \bigvee_{k\in[n]} (x_{i,k} \wedge x_{j,k}) \,, \tag{7.1}$$

where each propositional variable $x_{i,j}$ designates that the pigeon i is mapped to the hole j. It is clear that if $m > n$ then FPHP_n^m is a tautology. The negation of (7.1), formulated as an unsatisfiable CNF formula, consists of the following clauses:

$$\begin{array}{ll} \forall i \in [m], & x_{i,1} \vee \ldots \vee x_{i,n} \\ \forall i \in [m]\, \forall k < \ell \in [n], & \neg x_{i,k} \vee \neg x_{i,\ell} \\ \forall i < j \in [m]\, \forall k \in [n], & \neg x_{i,k} \vee \neg x_{j,k} \end{array} \tag{7.2}$$

The term *functional* (in Functional Pigeonhole Principle) comes from the second set of polynomials, that force each pigeon $i \in [m]$ to be mapped to at most one hole. The three sets of clauses in (7.2) translate via Definition 2.5 to the following set of polynomials, denoted $\neg\text{FPHP}_n^m$:

$$\begin{array}{lll} \text{Pigeons :} & \forall i \in [m], & \bar{x}_{i,1} \cdots \bar{x}_{i,n} \\ \text{Functional :} & \forall i \in [m]\, \forall k < \ell \in [n], & x_{i,k} \cdot x_{i,\ell} \\ \text{Holes :} & \forall i < j \in [m]\, \forall k \in [n], & x_{i,k} \cdot x_{j,k} \end{array} \tag{7.3}$$

Haken (1985) showed an exponential lower bound on the size of Resolution refutations of the Functional Pigeonhole Principle (where the number of holes is n and the number of pigeons is $n + 1$).

Pitassi *et al.* (1993) and independently Krajíček *et al.* (1995) showed exponential lower bounds on the size of proofs of the Functional Pigeonhole Principle in bounded depth Frege (again, where the number of holes is n and number of pigeons is $n + 1$).

Razborov (1998) and subsequently Impagliazzo *et al.* (1999) showed exponential lower bounds on the size (and degree) of PC-refutations of a different *low degree* version of the Functional Pigeonhole Principle. In this low degree version the Pigeons polynomials of (7.3) are replaced by $1 - (x_{i,1} + \ldots + x_{i,n})$, for all $i \in [m]$. This low degree version is not a translation of a CNF formula. Our upper bound is also applicable to this low-degree version of the Functional Pigeonhole Principle (however, this would not yield a separation of fMC from PC and PCR as we consider all proof systems in this paper as proof systems for (polynomial translations of) CNF formulas).

Grigoriev & Hirsch (2003) showed a polynomial size refutation of the Pigeonhole Principle (i.e., polynomials (7.3) without the Functional axioms) for a formal (see Section 2.3.5) propositional proof system, denoted $\mathcal{F}$-$\mathcal{PC}$, that manipulates general arithmetic formulas. Their refutation works with general arithmetic formulas of constant-depth.

The main theorem of this section is:

THEOREM 7.4. *Let $\mathbb{F}$ be a field of characteristic 0 and let m, n be natural numbers such that $m > n$. Then depth-3 fMC over $\mathbb{F}$ has a refutation of $\neg\text{FPHP}_n^m$ of size polynomial in n.*

By the discussion above we have:

COROLLARY 7.5. *Depth-3 fMC over fields of characteristic 0 has an exponential gap over Resolution and bounded-depth Frege for the Functional Pigeonhole Principle.*

Since depth-3 fMC polynomially simulates Resolution then by Corollary 7.5 depth-3 fMC is strictly stronger than Resolution. In Section 8 we shall see another example of an exponential gap of depth-3 fMC over Resolution, as well as over PC and PCR (over any field having a primitive p-th root of unity, for some p). This will be proved via Tseitin's graph tautologies.

Remark. Note that in order to prove Theorem 7.4, it is enough to show that there is a depth-3 fMC proof of $\neg\text{FPHP}_n^m$ of size polynomial in the number of *pigeons* m. The reason is that for any $m > n$, $\neg\text{FPHP}_n^{n+1}$ is an unsatisfiable subset of $\neg\text{FPHP}_n^m$. Thus, a refutation of $\neg\text{FPHP}_n^{n+1}$ of size polynomial in the number of *pigeons* $n+1$ is also a refutation of $\neg\text{FPHP}_n^m$.

The rest of this section is devoted to prove Theorem 7.4.

For all $k \in [n]$, let us fix the following abbreviation:

$$y_k := x_{1,k} + \ldots + x_{m,k}\,. \tag{7.6}$$

This means that the variables y_k are *not formal variables* of fMC, but rather a shorthand, i.e., in the actual proofs the variables y_k are replaced by the righthand side of (7.6). Throughout this section we use the variables y_k only according to this abbreviation.

LEMMA 7.7. *Let the variables y_k (for all $k \in [n]$) be the abbreviations as defined in (7.6). Then there is a polynomial size (in m) depth-3 fMC proof from $\neg\text{FPHP}_n^m$ of*

$$m - (y_1 + \ldots + y_n)\,. \tag{7.8}$$

PROOF. For all pigeons $i \in [m]$, replace one by one each variable $\bar{x}_{i,j}$, for $j \in [n]$, in the Pigeons axioms of (7.3) with $(1 - x_{i,j})$. We arrive at:

$$(1 - x_{i,1}) \cdots (1 - x_{i,n})\,. \tag{7.9}$$

By Proposition 3.6, this can be done with a depth-3 fMC proof of size polynomial in n. Polynomial (7.9) is equal to:

$$\sum_{J \subseteq [n]} (-1)^{|J|} \prod_{j \in J} x_{i,j}$$

(where we define $\prod_{j\in\emptyset} x_{i,j} := 1$), which is equal to the following depth 3 multilinear formula of *polynomial size in* n:

$$1-(x_{i,1}+\ldots+x_{i,n})+\underbrace{\sum_{k=1}^{n-1}\sum_{\ell=k+1}^{n} x_{i,k}\times x_{i,\ell}\times\prod_{j>\ell}(1-x_{i,j})}_{(\star)}\,. \tag{7.10}$$

Note that the term $(\star)$ is a sum of $O(n^2)$ polynomials, where each polynomial in this sum has a depth 2 multilinear formula of size $O(n)$ and computes some product of a Functional axiom $x_{i,k}\cdot x_{i,\ell}$ (for $k<\ell$). Thus, there is a polynomial size in n depth-3 fMC proof of $(\star)$ from $\neg\text{FPHP}_n^m$. Therefore, for every pigeon $i\in[m]$ we can subtract the term $(\star)$ from equation (7.10) in order to get:

$$1-(x_{i,1}+\ldots+x_{i,n})\,. \tag{7.11}$$

We thus get:

$$\begin{aligned} &1-(x_{1,1}+\ldots+x_{1,n})\\ &\vdots\\ &1-(x_{m,1}+\ldots+x_{m,n}) \end{aligned} \tag{7.12}$$

Since for all $k\in[m]$, $y_k = x_{1,k}+\ldots+x_{m,k}$, summing all (7.12) polynomials together and rearranging the terms, we get:

$$m-(y_1+\ldots+y_n)\,. \qquad \square$$

By Lemma 7.7, in order to prove Theorem 7.4 it remains to show a short depth-3 fMC refutation of (7.8) and $\neg\text{FPHP}_n^m$. In light of Theorem 5.1, it is sufficient to follow the following two steps:

Step 1: Show a PCR refutation π of $m-(y_1+\ldots+y_n)$ and $\neg\text{FPHP}_n^m$, where the number of steps in π (i.e., the number of proof lines) is polynomial in m (note that we do not speak about the *size* (i.e., the number of monomials) of the PCR refutation).

Step 2: Show that for each polynomial $p\in\pi$ from Step 1, there is a polynomial-size (in m) depth 3 multilinear formula for $\mathbf{M}[p]$ with a plus gate at the root (over fields of characteristic 0).

7.1. Step 1: a PCR refutation of $m-(y_1+\ldots+y_n)$ and $\neg\text{FPHP}_n^m$. For $1\le i\le n$, define the abbreviation

$$G_i := \sum_{k=1}^{i} y_k\,.$$

Hence, $m-(y_1+\ldots+y_n)$ can be written as

$$m-G_n\,. \tag{7.13}$$

In order to refute (7.13) (and $\neg\text{FPHP}_n^m$), we shall show a PCR proof from $\neg\text{FPHP}_n^m$ of

$$G_n\cdot(G_n-1)\cdots(G_n-n)\,. \tag{7.14}$$

Using (7.13), we will be able to substitute each occurrence of G_n in (7.14) with m. We arrive at $m!/(m-n-1)!>0$. By multiplying $m!/(m-n-1)!$ with its inverse in the field we arrive at 1, which is the terminal polynomial.

We make use of the following claim:

CLAIM 7.15. *For any $1 \leq k \leq n$, there is a* PCR *proof of $y_k^2 - y_k$ from $\neg\text{FPHP}_n^m$ of a polynomial number of steps.*

PROOF. Add together all Holes axioms pertaining to hole k and multiply the result by 2, we get:

$$\sum_{i \neq j \in [m]} x_{i,k} \cdot x_{j,k} \ . \tag{7.16}$$

Add together all PCR Boolean axioms of the form $x_{i,k}^2 - x_{i,k}$, for all $1 \leq i \leq m$:

$$\sum_{i=1}^{m} (x_{i,k}^2 - x_{i,k}) \ . \tag{7.17}$$

Adding (7.16) with (7.17), we get:

$$\begin{aligned}
&\sum_{i=1}^{m} (x_{i,k}^2 - x_{i,k}) + \sum_{i \neq j} x_{i,k} \cdot x_{j,k} \\
&= x_{1,k}^2 + \ldots + x_{m,k}^2 + \sum_{i \neq j} x_{i,k} \cdot x_{j,k} - (x_{1,k} + \ldots + x_{m,k}) \\
&= (x_{1,k} + \ldots + x_{m,k})^2 - (x_{1,k} + \ldots + x_{m,k}) \\
&= y_k^2 - y_k
\end{aligned}$$

□

7.1.1. A PCR proof of $G_n \cdot (G_n - 1) \cdots (G_n - n)$.. We shall need the following claim:

CLAIM 7.18. *For all $1 \leq i \leq n$ and all $0 \leq r \leq i$ and any polynomial q there are* PCR *proofs having polynomially (in m) many steps of*

$$q \cdot (G_i - r) \cdot (G_i - r - 1) \cdots (G_i - i)$$

from the polynomial q.

Remark. The existence of such a PCR proof having polynomially many *steps* is clear, since the proofs start from some polynomial q and derive a product of it $q \cdot q_0$ (for $q_0 = (G_i - r) \cdot (G_i - r - 1) \cdots (G_i - i)$), where also q_0 has a formula of size polynomial (in m) (note that if q_1 is a polynomial and Φ is a formula computing another polynomial q_2, then $q_1 \cdot q_2$ can be proved in PCR from q_1, with polynomially in $|\Phi|$ many *steps* (this can be shown simply by induction on the structure of Φ)). *Our goal, however, is to describe the* PCR *proof explicitly,* in order to show later (in Step 2) that every polynomial p in the proof sequence has small corresponding multilinear formula for $\mathbf{M}[p]$.

PROOF. Apply the following proof sequence (we skip here, and in the sequel, obvious proof sequences, e.g., a sequence of additions is described in one line; we also describe in one line a sequence of proof lines of the same form, e.g., line 2. Thus, line numbers come for convenience only,

and do not represent necessarily the actual positions of the lines in the proof sequence):

1. q — hypothesis
2. $q \cdot x_{j,k}$, for all $j \in [m], k \in [i]$ — product of (1)
3. $q \cdot G_i$ — add together all polynomials from (2)
4. $q \cdot (G_i - r)$ — subtract $r \cdot q$ (a scalar product of (1)) from (3)
5. $q \cdot (G_i - r) \cdot x_{j,k}$, for all $j \in [m], k \in [i]$ — product of (4)
6. $q \cdot (G_i - r) \cdot G_i$ — add together all polynomials from (5)
7. $q \cdot (G_i - r) \cdot (G_i - r - 1)$ — subtract $q \cdot (G_i - r) \cdot (r + 1)$ (a scalar product of (4)) from (6)

Continue in the same manner to reach $q \cdot (G_i - r) \cdot (G_i - r - 1) \cdots (G_i - i)$. □

The PCR proof of $G_n \cdot (G_n - 1) \cdots (G_n - n)$ is described by induction:

The base case is to show a PCR proof of $G_1 \cdot (G_1 - 1) = y_1 \cdot (y_1 - 1) = y_1^2 - y_1$, which follows from Claim 7.15.

The induction step is to show a PCR proof of $G_{i+1} \cdot (G_{i+1} - 1) \cdots (G_{i+1} - i - 1)$ from $G_i \cdot (G_i - 1) \cdots (G_i - i)$. This is shown in the following lemma (see a close proof in Grigoriev & Hirsch (2003)).

LEMMA 7.19. *For every $1 \leq i < n$, there is a* PCR *proof of $G_{i+1} \cdot (G_{i+1} - 1) \cdots (G_{i+1} - i - 1)$ from $G_i \cdot (G_i - 1) \cdots (G_i - i)$ and $\neg\text{FPHP}_n^m$ having polynomially (in m) many steps.*

PROOF. Apply the following PCR proof sequence:

1. $y_{i+1} \cdot (y_{i+1} - 1)$ — by Claim 7.15
2. $G_i \cdot (G_i - 1) \cdots (G_i - i)$ — hypothesis
3. $G_i \cdot (G_i - 1) \cdots (G_i - i) \cdot x_{j,i+1}$, for all $j \in [m]$ — product of (2)
4. $G_i \cdot (G_i - 1) \cdots (G_i - i) \cdot y_{i+1}$ — add together all polynomials from (3)
5. $G_i \cdot (G_i - 1) \cdots (G_i - i) \cdot (y_{i+1} - 1)$ — subtract (2) from (4)

Now use (1) to substitute G_i in (5) by $G_i + y_{i+1} = G_{i+1}$:

6. $(G_i - 1) \cdots (G_i - i) \cdot y_{i+1} \cdot (y_{i+1} - 1)$ — by (1) and Claim 7.18
7. $(G_i + y_{i+1}) \cdot (G_i - 1) \cdots (G_i - i) \cdot (y_{i+1} - 1) = (G_{i+1}) \cdot (G_i - 1) \cdots (G_i - i) \cdot (y_{i+1} - 1)$ — add (5) and (6)

Continue in a similar manner to substitute all G_i's in (7) by G_{i+1}, to reach:

8. $(G_{i+1}) \cdot (G_{i+1} - 1) \cdots (G_{i+1} - i) \cdot (y_{i+1} - 1)$.

By multiplying (2) with y_{i+1} (in a similar manner as above) we get $G_i \cdot (G_i - 1) \cdots (G_i - i) \cdot y_{i+1}$, and by substituting in it, as was demonstrated above, all G_i by $G_i + (y_{i+1} - 1) = G_{i+1} - 1$ we arrive at:

9. $(G_{i+1} - 1) \cdot (G_{i+1} - 2) \cdots (G_{i+1} - i - 1) \cdot y_{i+1}$

Multiplying (8) by $(G_{i+1} - i - 1)$ and (9) by G_{i+1} (both by Claim 7.18), and subtracting the two resulting polynomials we arrive finally at:

$$G_{i+1} \cdot (G_{i+1} - 1) \cdots (G_{i+1} - i - 1).$$

□

7.1.2. Concluding the PCR refutation. Assume we have already the following polynomials inside a PCR proof sequence (due to Section 7.1.1):

$$m - G_n \,, \tag{7.20}$$

and

$$G_n \cdot (G_n - 1) \cdots (G_n - n) \,. \tag{7.21}$$

We shall substitute each occurrence of G_n in (7.21) by m. Specifically, apply the following PCR proof sequence:

1. $(m - G_n) \cdot (G_n - 1) \cdots (G_n - n)$ product of (7.20) (by Claim 7.18)
2. $m \cdot (G_n - 1) \cdots (G_n - n)$ add (1) and (7.21)
3. $m \cdot (m - G_n) \cdot (G_n - 2) \cdots (G_n - n)$ product of (7.20) (by Claim 7.18)
4. $m \cdot (m - 1) \cdot (G_n - 2) \cdots (G_n - n)$ add (3) and (2)

...

Continuing in the same manner, we arrive finally at

$$m \cdot (m - 1) \cdots (m - n) = m!/(m - n - 1)! > 0 \,. \tag{7.22}$$

By multiplying (7.22) by the inverse of $m!/(m - n - 1)!$ in the field, we get the polynomial 1.

7.2. Step 2: multilinearization of polynomials. In this section we show that for every polynomial p occurring in the PCR proof described in Step 1, the corresponding multilinear polynomial $\mathbf{M}[p]$ has a polynomial-size (in m) depth 3 multilinear formula (over fields of characteristic 0).

7.2.1. Symmetric polynomials. A *renaming* of the variables in X is a permutation $\sigma \in S_\ell$ (the symmetric group on $[\ell]$) such that x_i is mapped to $x_{\sigma(i)}$ for every $1 \leq i \leq \ell$.

Definition 7.23. *Given a set of variables $X = \{x_1, \ldots, x_\ell\}$, a symmetric polynomial f over X is a polynomial in (all the variables of) X such that renaming of variables does not change the (formal) polynomial.*

For example, $1 + x_1 + x_2 + x_1 \cdot x_2^3 + x_2 \cdot x_1^3$ is a symmetric polynomial over $X = \{x_1, x_2\}$.

The following theorem is due to M. Ben-Or (cf. Theorem 5.1 in Shpilka & Wigderson (2001)):

Theorem 7.24. **(Ben-Or)** *Let $\mathbb{F}$ be a field of characteristic 0 and let X be a set of ℓ variables $\{x_1, \ldots, x_\ell\}$. For any multilinear symmetric polynomial over X (over the field $\mathbb{F}$) there is a polynomial-size (in ℓ) depth 3 multilinear formula. Moreover, this formula is a leveled multilinear formula with a plus gate at the root.*

We need the following simple properties (given without a proof):

PROPOSITION 7.25. *Fix a field $\mathbb{F}$ and let X be a finite set of variables.*

- *If p, q are two symmetric polynomials over X, then the product $p \cdot q$ is also a symmetric polynomial over X;*
- *If p is a symmetric polynomial over X, then $\mathbf{M}[p]$ is a multilinear symmetric polynomial over X.*

From Theorem 7.24 and Proposition 7.25 we get:

COROLLARY 7.26. *Let $\mathbb{F}$ be a field of characteristic 0 and let X be a set of ℓ variables. If p is a product of (one or more) symmetric polynomials over X (over the field $\mathbb{F}$), then $\mathbf{M}[p]$ has a depth 3 multilinear formula of size polynomial (in ℓ), with a plus gate at the root.*

We shall also need the following more general proposition, which might be interesting by itself:

PROPOSITION 7.27. *Let $\mathbb{F}$ be a field of characteristic 0. For a constant c, let $X_1, \ldots, X_c$ be c finite sets of variables (not necessarily disjoint), where $\Sigma_{i=1}^{c} |X_i| = \ell$. Let $f_1, \ldots, f_c$ be c symmetric polynomials over $X_1, \ldots, X_c$ (over the field $\mathbb{F}$), respectively. Then, there is a depth 3 multilinear formula for $\mathbf{M}[f_1 \cdots f_c]$ of size polynomial (in ℓ), with a plus gate at the root.*

Remark. Note the difference between Corollary 7.26 and Proposition 7.27. Corollary 7.26 speaks about a (finite) *unbounded* product of symmetric polynomials over the *same* set of variables. On the other hand, Proposition 7.27 speaks about a product of *constant* number of symmetric polynomials over *different* (but not necessarily disjoint) sets of variables.

PROOF. We shall need the following two basic claims (given without a proof).

CLAIM 7.28. *Let X be a set of ℓ variables $x_1, \ldots, x_\ell$, and let p_1, p_2 be two multilinear polynomials over X such that for all $0,1$ assignments to $x_1, \ldots, x_\ell$, $p_1(x_1, \ldots, x_\ell) = p_2(x_1, \ldots, x_\ell)$. Then $p_1 = p_2$ as* formal polynomials.

Since symmetric polynomials are invariant under renaming of variables then restricted to $0,1$ assignments the values of symmetric polynomials are determined only by the *number of* 1*'s* in their input variables. Formally, if p is a symmetric polynomial of degree d from $\mathbb{F}[X]$ in ℓ variables, then there is a polynomial h of degree at most d in one variable, such that for $0,1$ assignments to $x_1, \ldots, x_\ell$, $p(x_1, \ldots, x_\ell) = h(x_1 + \ldots + x_\ell)$. Hence, if we let $Y_1, \ldots, Y_m$ be pairwise disjoint subsets of $X = \{x_1, \ldots, x_\ell\}$, such that $\biguplus_{i=1}^{m} Y_i = X$, then we have the following:

CLAIM 7.29. *Let p be a symmetric polynomial from $\mathbb{F}[X]$ of degree d, then there is a polynomial h of degree at most d in m variables, such that for all $0,1$ assignments to $x_1, \ldots, x_\ell$,*

$$p(x_1, \ldots, x_\ell) = h\left(\sum_{x_i \in Y_1} x_i, \ldots, \sum_{x_i \in Y_m} x_i \right).$$

We are now ready to prove Proposition 7.27. Let $\mathbf{X} := \bigcup_{i=1}^{c} X_i$. Let $m := 2^c$ and partition $\mathbf{X}$ into at most m disjoint subsets as follows. For every $J \subseteq [c]$, let $X_J := \bigcap_{i \in J} X_i \setminus \bigcup_{i \in [c] \setminus J} X_i$ and define the abbreviation

$$z_J := \sum_{x_i \in X_J} x_i \,. \tag{7.30}$$

(This way, the variables in X_i are exactly the variables that occur in all z_J, such that $i \in J \subseteq [c]$.) Let $J_1, \ldots, J_m$ be all the subsets of $[c]$, and let z_k denote z_{J_k}, for every $1 \leq k \leq m$.

We clearly have,

$$\mathbf{M}\left[\prod_{i=1}^{c} f_i\right] = \mathbf{M}\left[\prod_{i=1}^{c} \mathbf{M}[f_i]\right] \,. \tag{7.31}$$

By Proposition 7.25, for all $1 \leq i \leq c$, $\mathbf{M}[f_i]$ is a (multilinear) symmetric polynomial. Thus, by Claim 7.29, for all $1 \leq i \leq c$ there exists a polynomial $g_i(y_1, \ldots, y_m)$ of degree at most ℓ (with at most m variables), such that $\mathbf{M}[f_i] = g_i(z_1, \ldots, z_m)$ for all assignments of $0, 1$ values *to the variables in* X_i (note that $g_i(y_1, \ldots, y_m)$ is not necessarily a multilinear polynomial in the y_j's).[6]

Hence, by (7.31) for all assignments of $0, 1$ to the variables in $\mathbf{X}$,

$$\mathbf{M}\left[\prod_{i=1}^{c} f_i\right] = \mathbf{M}\left[\prod_{i=1}^{c} g_i(z_1, \ldots, z_m)\right] \tag{7.32}$$

(note that the multilinearization operator $\mathbf{M}[\cdot]$ in the right hand side of (7.32) operates on the polynomial $\prod_i^c g_i(z_1, \ldots, z_m)$ considered as a polynomial in the $\mathbf{X}$ variables).

Therefore, by Claim 7.28, the two sides of (7.32) are equal as *formal* polynomials (over $\mathbf{X}$). Since c and m are constants, $\prod_i^c g_i(y_1, \ldots, y_m)$ can be written as a sum of polynomially many monomials in the variables $y_1, \ldots, y_m$. Thus, when substituting z_j's for y_j's (for all $1 \leq j \leq m$), $\prod_i^c g_i(z_1, \ldots, z_m)$ can be written as a sum of polynomially many products of the form $\prod_{j=1}^{m} z_j^{e_j}$ (where the e_j's stand for some non-negative integers). Hence, by linearity of $\mathbf{M}[\cdot]$, the right hand side of (7.32) can be written as a sum of polynomially many terms of the form $\mathbf{M}\left[\prod_{j=1}^{m} z_j^{e_j}\right]$. It remains only to prove the following:

Claim 7.33. *Every polynomial of the form* $\mathbf{M}\left[\prod_{j=1}^{m} z_j^{e_j}\right]$ *(where the* e_j*'s stand for some non-negative integers) has a depth* 3 *multilinear formula (in the variables in* $\mathbf{X}$*) of size polynomial in* ℓ *and a plus gate at the root .*

Proof. Since the sets of variables that occur in each of the z_j's are pairwise disjoint, $\mathbf{M}\left[\prod_{j=1}^{m} z_j^{e_j}\right] = \prod_{j=1}^{m} \mathbf{M}\left[z_j^{e_j}\right]$. For every $1 \leq j \leq m$, $z_j^{e_j}$ is a product of symmetric polynomials (in (not necessarily all) the variables in $\mathbf{X}$). Thus, by Corollary 7.26, $\mathbf{M}\left[z_j^{e_j}\right]$ can be written as a sum of polynomially (in ℓ) many products of linear polynomials (in other words, a polynomial-size leveled depth 3 multilinear formula with a plus gate at the root). Since m is a constant, $\prod_{j=1}^{m} \mathbf{M}\left[z_j^{e_j}\right]$ can be written as a sum of polynomially many terms, where each term is a product of (polynomially many) linear polynomials over *disjoint sets of variables*. In other words, we have reached a polynomial-size (in ℓ) depth 3 multilinear formula. □

[6] For any $1 \leq k \leq m$, the variable y_k actually occurs in g_i if and only if $i \in J_k$. The other variables y_k are still indicated for ease of writing.

This completes the proof of Proposition 7.27. □

7.2.2. Concluding Step 2. It is important to note that all the polynomials that appeared in Step 1, are polynomials in the formal variables $x_{i,j}, \bar{x}_{i,j}$, for $i \in [m], j \in [n]$, only.

For $1 \leq k \leq n$ define,

$$X_k := \{x_{i,j} \mid i \in [m], j \in [k]\}.$$

Notice that the variables in X_k are exactly the variables that occur in $y_1, \ldots, y_k$ (according to the abbreviation $y_k := x_{1,k} + \ldots + x_{m,k}$).

The following lemma concludes Step 2, and thus the proof of Theorem 7.4:

LEMMA 7.34. *For every polynomial p in the* PCR *refutation of Step 1, the corresponding multilinear polynomial $\mathbf{M}[p]$ has a multilinear formula of size polynomial in m and depth at most 3 (when the underlying field is of characteristic 0); and further, if the depth of the multilinear formula is 3 then it has a plus gate at its root.*

PROOF. **Case 1**: p is an axiom of $\neg\text{FPHP}_n^m$ or a Boolean axiom of PCR.

Notice that all the polynomials of $\neg\text{FPHP}_n^m$ (7.3) are multilinear, and have polynomial-size (in $n < m$) multilinear formulas of depth 1. The multilinear polynomials corresponding to the Boolean axioms of PCR have constant size depth ≤ 1 multilinear formulas: $\mathbf{M}\left[x_i^2 - x_i\right] = 0$ and $x_i + \bar{x}_i - 1$ is already multilinear.

Case 2: p is a polynomial in the PCR proof sequence of $y_i^2 - y_i$ (for $1 \leq i \leq n$) (Claim 7.15).

Note that (7.16) is already multilinear and has size $O(m^2)$ and depth 1 multilinear formula. Further, applying $\mathbf{M}[\cdot]$ on (7.17) yields the zero polynomial.

Case 3: p is a polynomial in the PCR proof of $G_n \cdot (G_n - 1) \cdots (G_n - n)$ (Section 7.1.1).

It is sufficient to consider the polynomials that occur in the PCR proof sequences described in the proofs of Claim 7.18 and Lemma 7.19. It is straightforward to verify that every such polynomial is a product of a constant number of symmetric polynomials (over different, but not necessarily disjoint, sets of variables). Thus, by Proposition 7.27, it follows that $\mathbf{M}[p]$ has depth 3 multilinear formula of size polynomial in m and a plus gate at the root.

For example, consider a typical proof line: line 7 of the PCR proof sequence presented in the proof of Lemma 7.19:

$$(G_{i+1}) \cdot (G_i - 1) \cdots (G_i - i) \cdot (y_{i+1} - 1).$$

Note that G_{i+1} is a symmetric polynomial over X_{i+1}; and $(G_i - 1) \cdots (G_i - i)$ is a symmetric polynomial over X_i (as a product of symmetric polynomials over X_i); and $y_{i+1} - 1$ is a symmetric polynomial over $\{x_{1,i+1}, \ldots, x_{m,i+1}\}$.

Case 4: p is a polynomial in the PCR proof sequence in Section 7.1.2.

It is straightforward to verify that p is a symmetric polynomial over X_n, as a product of symmetric polynomials over X_n. Thus, by Corollary 7.26, the lemma holds for p. □

8. Tseitin's Graph Tautologies

In this section we show an exponential gap of depth-3 fMC over Resolution, PC, PCR and bounded-depth Frege for certain Tseitin's graph tautologies (over any characteristic). Specifically, we show that for any p the Tseitin mod p formula (see Definition 8.1) has a polynomial-size depth-3 fMC refutation over any field of characteristic $q \nmid p$ that includes a primitive p-th root of unity.

We shall consider the generalization of the Tseitin's graph tautologies, given in Buss *et al.* (2001). This generalization can be formulated as a CNF formula, which in turn can be reduced to a more convenient form (we observe that this reduction is efficiently provable in depth-2 fMC). Given this latter form, the refutations of the generalized Tseitin formulas follow in a rather straightforward manner, and we show that such refutations can be done efficiently in depth-3 fMC.

It is worth noting that Grigoriev & Hirsch (2003) have shown a polynomial-size constant depth refutation of the Tseitin mod 2 principle in a formal proof system manipulating general arithmetic formulas of constant-depth (denoted $\mathcal{F}$-NS).

Preparatory to the generalized Tseitin principle we start by describing the (original) Tseitin mod 2 principle (cf. Tseitin (1968)). Let $G = (V, E)$ be a connected undirected graph with an *odd* number of vertices n. The Tseitin mod 2 tautology states that there is no sub-graph $G' = (V, E')$, where $E' \subseteq E$, so that for *every* vertex $v \in V$, the number of edges from E' incident to v is odd. This statement is valid, since otherwise, summing the degrees of all the vertices in G' would amount to an odd number (since n is odd), whereas this sum also counts every edge in E' twice, and so is even.

The Tseitin mod 2 principle can be generalized to obtain the Tseitin mod p principle, as was suggested in Buss *et al.* (2001). Let $p \geq 2$ be some fixed integer and let $G = (V, E)$ be a connected undirected r-regular graph with n vertices and no double edges. Let $G' = (V, E')$ be the corresponding *directed* graph that results from G by replacing every (undirected) edge in G with two opposite directed edges. Assume that $n \equiv 1 \pmod{p}$. Then the Tseitin mod p principle states that there is no way to assign to every edge in E' a value from $\{0, \ldots, p-1\}$, so that:

(i) For every pair of opposite directed edges $e, \bar{e}$ in E', with assigned values a, b, respectively, $a + b \equiv 0 \pmod{p}$; and

(ii) For every vertex v in V, the sum of the values assigned to the edges in E' coming out of v is congruent to 1 (mod p).

The Tseitin mod p principle is valid, since if we sum the values assigned to all edges of E' in pairs we obtain 0 (mod p) (by (i)), where summing them by vertices we arrive at a total value of 1 (mod p) (by (ii) and since $n \equiv 1 \pmod{p}$).

As a propositional formula (in CNF form) the Tseitin mod p principle is formulated by assigning a variable $x_{e,i}$ for every edge $e \in E'$ and every residue i modulo p. The variable $x_{e,i}$ is an indicator variable for the fact that edge e has an associated value i. The following are the clauses of the Tseitin mod p CNF formula, as translated to polynomials (we call it the Tseitin mod p *formula* to emphasize that it is a translation of a CNF formula). (To be consistent with Buss *et al.* (2001) we use the notation $\mathrm{BTS}_{G,p}$ which stands for 'Boolean Tseitin mod p'.)

DEFINITION 8.1. **(Tseitin mod p formula ($\mathrm{BTS}_{G,p}$)).** *Let $p \geq 2$ be some fixed integer and let $G = (V, E)$ be a connected undirected r-regular graph with n vertices and no double edges, and assume that $n \equiv 1 \pmod{p}$. Let $G' = (V, E')$ be the corresponding directed graph that results from G by replacing every (undirected) edge in G with two opposite directed edges.*

Given a vertex $v \in V$, let the edges in E' coming out of v be $e_{v,1}, \ldots, e_{v,r}$ and define the following set of polynomials:

$$\mathrm{MOD}_{p,1}(v) := \left\{ \prod_{k=1}^{r} x_{e_{v,k},i_k} \,\middle|\, i_1, \ldots, i_r \in \{0, \ldots, p-1\} \text{ and } \sum_{k=1}^{r} i_k \not\equiv 1 \mod p \right\}.$$

The Tseitin mod p *formula, denoted* $\mathrm{BTS}_{G,p}$*, consists of the following multilinear polynomials, where each polynomial is easily seen to be a translation of a clause (via Definition 2.5):*

1. $\prod_{i=0}^{p-1} \bar{x}_{e,i}$ *, for all* $e \in E'$
   (expresses that every edge is assigned at least one value from $0, \ldots, p-1$*);*
2. $x_{e,i} \cdot x_{e,j}$ *, for all* $i \neq j \in \{0, \ldots, p-1\}$ *and all* $e \in E'$
   (expresses that every edge is assigned at most one value from $0, \ldots, p-1$*);*
3. $\bar{x}_{e,i} \cdot x_{\bar{e},p-i}$ *and* $x_{e,i} \cdot \bar{x}_{\bar{e},p-i}$,[7]
   for all two opposite directed edges $e, \bar{e} \in E'$ *and all* $i \in \{0, \ldots, p-1\}$
   (expresses condition (i) of the Tseitin mod p principle above);
4. $\mathrm{MOD}_{p,1}(v)$ *, for all* $v \in V$
   (expresses condition (ii) of the Tseitin mod p principle above).

Note that for every edge $e \in E'$, the polynomials of (1,2) in Definition 8.1, combined with the Boolean axioms of fMC, force any collection of edge-variables $x_{e,0}, \ldots, x_{e,p-1}$ to have exactly one true variable $x_{e,i}$, for some $i \in \{0, \ldots, p-1\}$. Also, it is easy to verify that, given a vertex $v \in V$, any assignment σ of $0, 1$ values (to the relevant variables) satisfies both the clauses of (1,2) and the clauses of $\mathrm{MOD}_{p,1}(v)$ if and only if σ corresponds to an assignment of values from $\{0, \ldots, p-1\}$ to the edges coming out of v that sums up to 1 (mod p).

DEFINITION 8.2. *Let $G = (V, E)$ be an undirected graph, and let $\epsilon > 0$. The graph G has* expansion ϵ *if for any subset $S \subseteq V$ of vertices with $|S| \leq |V|/2$, $|N(S)| \geq (1 + \epsilon)|S|$, where $N(S)$ is the set of all vertices from V incident to vertices in S.*

THEOREM 8.3. **(Buss *et al.* (2001))** *Let $q \geq 2$ be a prime such that $q \nmid p$ and let $\mathbb{F}$ be a field of characteristic q. Let G be an r-regular graph with n vertices and expansion $\epsilon > 0$. Then, any* PCR*-refutation (over $\mathbb{F}$) of* $\mathrm{BTS}_{G,p}$ *requires degree $\Omega(n)$.*

It can be proved that there exist *constants* r, $\epsilon > 0$ and an infinite family of r-regular graphs $\{G_i\}_{i=1}^{\infty}$, such that every G_i has ϵ expansion and n_i vertices, and n_i tends to infinity as i tends to infinity (cf. Alon (1986)). Thus, for each G_i pertaining to such a family, the corresponding set

[7] If $i = 0$ then $x_{\bar{e},p-i}$ and $\bar{x}_{\bar{e},p-i}$ denote $x_{\bar{e},0}$ and $\bar{x}_{\bar{e},0}$, respectively.

$\mathrm{BTS}_{G_i,p}$ contains only linear in n_i many polynomials (since for each vertex v in G_i, $\mathrm{MOD}_{p,1}(v)$ defines $< p^r$ many polynomials). Notice also that every polynomial in $\mathrm{BTS}_{G_i,p}$ is a multilinear monomial and has a constant number of variables. So we conclude that the total number of variables in $\mathrm{BTS}_{G_i,p}$ is linear in n_i and the total number of monomials in $\mathrm{BTS}_{G_i,p}$ is also linear in n_i.

By Theorem 8.3, $\mathrm{BTS}_{G_i,p}$ has a linear in n_i *degree* lower bound. By the previous paragraph, this means that $\mathrm{BTS}_{G_i,p}$ has a linear in the total number of variables degree lower bound. By the size-degree tradeoff proved in Impagliazzo *et al.* (1999), a linear (in the number of variables) lower bound on the degree of PCR refutations implies an exponential (in the number of variables) lower bound on the *size* of PCR refutations (this tradeoff was proved for PC (Corollary 6.3 in Impagliazzo *et al.* (1999)), but it is also valid for PCR as was observed in Alekhnovich *et al.* (2002)). Therefore, we have:

COROLLARY 8.4. *Let $q \geq 2$ be a prime such that $q \nmid p$ and let $\mathbb{F}$ be a field of characteristic q. For infinitely many n, there is a graph G with n vertices, such that the Tseitin mod p formula $\mathrm{BTS}_{G,p}$ has polynomial-size (i.e., it has polynomially in n many monomials), and any* PCR *refutation over $\mathbb{F}$ of $\mathrm{BTS}_{G,p}$ has size exponential in n (i.e., the refutation has exponentially many monomials in n).*

We shall show now that if the field $\mathbb{F}$ in Corollary 8.4 contains a primitive p-th root of unity, then for any G there is a *polynomial-size* depth-3 fMC refutation of $\mathrm{BTS}_{G,p}$ (over $\mathbb{F}$). For this purpose, we first transform the Tseitin mod p formula into the following multiplicative version (cf. Buss *et al.* (2001)):

DEFINITION 8.5. **(multiplicative Tseitin mod p ($\mathrm{TS}_{G,p}$)).** *Let $\mathbb{F}$ be a field of characteristic $q \nmid p$ having a primitive p-th root of unity, denoted by ω (that is, $\omega \neq 1$ and p is the smallest positive integer such that $\omega^p = 1$). Let $G' = (V, E')$ be the graph corresponding to G as in Definition 8.1. Define the abbreviation $y_e := \sum_{i=0}^{p-1} x_{e,i} \cdot \omega^i$ for every edge $e \in E'$. The* multiplicative Tseitin mod p, *denoted $\mathrm{TS}_{G,p}$, is the following set of* multilinear *polynomials over $\mathbb{F}$:*

1. $y_e \cdot y_{\bar{e}} - 1$, *for all two opposite directed edges $e, \bar{e} \in E'$;*
2. $\prod_{j=1}^{r} y_{e_j} - \omega$, *for all $v \in V$, where $e_1, \ldots, e_r$ are the edges coming out of v.*

We emphasize that the formal variables of $\mathrm{TS}_{G,p}$ are $x_{e,i}$ for all $e \in E'$ and $i \in \{0, \ldots, p-1\}$. (In Buss *et al.* (2001) $\mathrm{TS}_{G,p}$ also included the polynomials $y_e^p - 1$ for all edges $e \in E'$. We shall not need these polynomials for the upper bound.)

Notice that every Boolean assignment to the $x_{e,i}$ variables (where $e \in E'$ and $i \in \{0, \ldots, p-1\}$) that satisfies the polynomials in lines (1,2) and line (3) in $\mathrm{BTS}_{G,p}$, also satisfies the polynomials in line (1) in $\mathrm{TS}_{G,p}$. Indeed, let ρ be a Boolean assignment that satisfies the polynomials in lines (1,2) and line (3) in $\mathrm{BTS}_{G,p}$. Then, by lines (1,2) in $\mathrm{BTS}_{G,p}$ there is exactly one variable $x_{e,i}$ from the variables $x_{e,0}, \ldots, x_{e,p-1}$ in y_e that is set to 1 by ρ, and similarly there is exactly one variable $x_{\bar{e},j}$ from the variables $x_{\bar{e},0}, \ldots, x_{\bar{e},p-1}$ in $y_{\bar{e}}$ that is set to 1 in ρ. Thus, under the assignment ρ, $y_e = \omega^i$ and $y_{\bar{e}} = \omega^j$. By line (3) in $\mathrm{BTS}_{G,p}$ we have that $i + j = 0 \pmod p$, and so $y_e \cdot y_{\bar{e}} = \omega^i \cdot \omega^j = 1$ under the assignment ρ. Similar reasoning shows that every Boolean assignment to the $x_{e,i}$ variables (where $e \in E'$ and $i \in \{0, \ldots, p-1\}$) that satisfies the polynomials in lines (1,2) and line (4) in $\mathrm{BTS}_{G,p}$, also satisfies the polynomials in line (2) in $\mathrm{TS}_{G,p}$.

The previous paragraph shows that $\mathrm{BTS}_{G,p}$ semantically implies $\mathrm{TS}_{G,p}$. In fact, by Buss *et al.* (2001) there is a PCR-proof of $\mathrm{TS}_{G,p}$ from $\mathrm{BTS}_{G,p}$, such that all the polynomials in the proof are of degree at most pr:

LEMMA 8.6. **(Buss *et al.* (2001))** *For any r-regular graph G, and for any field $\mathbb{F}$ of characteristic $q \nmid p$ that includes a primitive p-th root of unity, there is a* PCR*-proof (over $\mathbb{F}$) of $\mathrm{TS}_{G,p}$ from $\mathrm{BTS}_{G,p}$, where the degrees of the polynomials in the proof are at most pr.*[8]

COROLLARY 8.7. *For any r-regular graph G with n vertices, and for any field $\mathbb{F}$ of characteristic $q \nmid p$ that includes a primitive p-th root of unity, there is a depth-2* fMC *(over $\mathbb{F}$) proof of $\mathrm{TS}_{G,p}$ from $\mathrm{BTS}_{G,p}$ of size polynomial in n, assuming r is a constant.*

PROOF. Since r and p are constants, then by Lemma 8.6 there is a PCR-proof of $\mathrm{TS}_{G,p}$ from $\mathrm{BTS}_{G,p}$ of constant-degree. The results of Clegg *et al.* (1996) imply that any constant-degree PCR-proof can be transformed into a polynomial-size (in the number of variables) PCR proof. The number of (formal) variables in $\mathrm{BTS}_{G,p}$ (and, hence $\mathrm{TS}_{G,p}$) is $2prn$, in other words, polynomial in n. Thus, there is a PCR-proof of $\mathrm{TS}_{G,p}$ from $\mathrm{BTS}_{G,p}$ of size polynomial in n. By Corollary 5.2, there is also such a depth-2 fMC proof of size polynomial in n. □

The following is the main theorem of this section:

THEOREM 8.8. *Let $\mathbb{F}$ be a field of characteristic $q \nmid p$ that includes a primitive p-th root of unity. Let G be an r-regular graph with n vertices. Then, there is a depth-3* fMC *polynomial-size (in n) refutation of $\mathrm{BTS}_{G,p}$ over $\mathbb{F}$.*

Proof. By Corollary 8.7, we first derive the polynomials of $\mathrm{TS}_{G,p}$ from $\mathrm{BTS}_{G,p}$, with a depth-2 fMC proof of size polynomial in n.

Given $\mathrm{TS}_{G,p}$, the refutation idea is straightforward: Recall that we interpret a polynomial t in an fMC proof sequence as the equation $t = 0$. Thus, the first axiom of $\mathrm{TS}_{G,p}$ interprets as $y_e \cdot y_{\bar{e}} = 1$, and the second axiom interprets as $\prod_{i=1}^{r} y_{e_i} = \omega$. Therefore, the multiplication of all polynomials $\prod_{i=1}^{r} y_{e_i}$, for all $v \in V$, equals 1, by the first axiom. On the other hand, by the second axiom, this multiplication equals $\omega^n = \omega$ (since $n \equiv 1 \pmod{p}$). So we reached $\omega = 1$, a contradiction.

More formally, the depth-3 fMC refutation goes as follows. For any $v \in V$, denote by E'_v the set of edges from E' that come out of v. Let v_0 be some vertex in V. Apply the following depth-3 fMC proof sequence:

1. $\displaystyle\prod_{e \in E'_{v_0}} y_e - \omega$ — hypothesis
2. $\displaystyle\left(\prod_{e \in E'_{v_0}} y_e - \omega\right) \cdot \prod_{v \in V \setminus \{v_0\}} \prod_{e \in E'_v} y_e = \prod_{e \in E'} y_e - \omega \cdot \prod_{v \in V \setminus \{v_0\}} \prod_{e \in E'_v} y_e$ — product of (1)

Now, choose a different vertex $v_1 \neq v_0$ from V.

3. $\displaystyle\prod_{e \in E'_{v_1}} y_e - \omega$ — hypothesis

[8] This result was proved in Buss *et al.* (2001) for PC, when one replaces in $\mathrm{TS}_{G,p}$ and $\mathrm{BTS}_{G,p}$ every occurrence of $\bar{x}_i$ (for any $1 \leq i \leq n$) by $1 - x_i$. Lemma 8.6 clearly stems from this.

4. $\left(\prod_{e\in E'_{v_1}} y_e - \omega\right)\cdot\omega\cdot\prod_{v\in V\setminus\{v_0,v_1\}}\prod_{e\in E'_v} y_e$

$= \omega\cdot\prod_{v\in V\setminus\{v_0\}}\prod_{e\in E'_v} y_e - \omega^2\cdot\prod_{v\in V\setminus\{v_0,v_1\}}\prod_{e\in E'_v} y_e$ product of (3)

5. $\prod_{e\in E'} y_e - \omega^2\cdot\prod_{v\in V\setminus\{v_0,v_1\}}\prod_{e\in E'_v} y_e$ add (2) and (4)

Continuing in the same manner for all other vertices $v \in V$, we arrive at $\prod_{e\in E'} y_e - \omega^n$, which equals

$$6.\quad \prod_{e\in E'} y_e - \omega\,,$$

over $\mathbb{F}$, since $n \equiv 1 \pmod p$.

We now substitute by 1 each product $y_e \cdot y_{\bar{e}}$ in (6), for any two opposite directed edges $e, \bar{e} \in E'$. Specifically, choose a pair of opposite directed edges $e_0, \bar{e}_0 \in E'$.

7. $y_{e_0}\cdot y_{\bar{e}_0} - 1$ hypothesis

8. $(y_{e_0}\cdot y_{\bar{e}_0} - 1)\cdot\prod_{e\in E'\setminus\{e_0,\bar{e}_0\}} y_e = \prod_{e\in E'} y_e - \prod_{e\in E'\setminus\{e_0,\bar{e}_0\}} y_e$ product of (7)

In the same manner, let $e_1, \bar{e}_1 \in E'$ be another pair of opposite directed edges. We can multiply $y_{e_1}\cdot y_{\bar{e}_1} - 1$ by $\prod_{e\in E'\setminus\{e_0,\bar{e}_0,e_1,\bar{e}_1\}} y_e$ and reach $\prod_{e\in E'\setminus\{e_0,\bar{e}_0\}} y_e - \prod_{e\in E'\setminus\{e_0,\bar{e}_0,e_1,\bar{e}_1\}} y_e$. Adding this to (8) yields $\prod_{e\in E'} y_e - \prod_{e\in E'\setminus\{e_0,\bar{e}_0,e_1,\bar{e}_1\}} y_e$. Continuing this process for all other pairs of opposite directed edges from E', we arrive finally at

$$9.\quad \prod_{e\in E'} y_e - 1\,.$$

Subtracting (9) from (6) we reach $1-\omega$. Since, $\omega \neq 1$, then $1-\omega$ has an inverse in the field, so by multiplying $1-\omega$ by its inverse we arrive at the terminal polynomial 1.

It is easy to verify that when replacing every variable y_e with its corresponding polynomial $\sum_{i=0}^{p-1} x_{e,i}\cdot\omega^i$ (which constitutes a depth 1 multilinear formula with a plus gate at the root), every polynomial in the above proof sequence can be written as a depth 3 multilinear formula of polynomial-size (in n) with a plus gate at the root. □

By Corollary 8.4 and Theorem 8.8, we conclude that:

Corollary 8.9. *Over fields of any characteristic q that include a primitive p-th root of unity, where $q \nmid p$, depth-3* fMC *has an exponential gap over* PC *and* PCR *for Tseitin mod p formulas (when the underlying graphs are appropriately expanding).*

It is known that the Tseitin mod 2 formulas have only exponential-size refutations in Resolution (again, when the underlying graphs are appropriately expanding; see Ben-Sasson & Wigderson (1999); Urquhart (1987)). Moreover, Ben-Sasson (2002) proved an exponential lower bound on bounded-depth Frege proof-size of such Tseitin mod 2 formulas. Therefore, by Theorem 8.8:

COROLLARY 8.10. *Over fields of characteristic different than* 2 *depth-*3 fMC *has exponential gap over Resolution and bounded-depth Frege for Tseitin mod* 2 *formulas (when the underlying graphs are appropriately expanding).*

Notice that the refutation of the Tseitin mod p formula described in the proof of Theorem 8.8 uses only depth 3 multilinear formulas, with *constant number of product gates* (in other words, the root is a plus gate with a constant fan-in), or depth 2 multilinear formulas (by Corollary 8.7). Dvir & Shpilka (2005) (Theorem 6.10) showed a deterministic polynomial-time algorithm for deciding the identity of such formulas – i.e., a polynomial-time algorithm that receives two (leveled) multilinear formulas of depth 3 with a constant fan-in plus gate at the root, and answers whether the two formulas compute the same polynomial. Thus, depth-3 fMC proof systems for which all depth 3 multilinear formulas appearing in proofs have a constant fan-in plus gate at the root constitute *formal* propositional proof systems (see Section 2.3.5) (note that these proof systems can manipulate *any kind* of depth 2 or depth 1 multilinear formulas; we only restrict the way depth 3 multilinear formulas appear). Therefore, by Corollaries 8.9 and 8.10, we have the following:

COROLLARY 8.11. *Depth-*3 *fMC proof systems for which all depth* 3 *multilinear formulas appearing in proofs have a constant fan-in plus gate at the root, are sound and complete* formal *proof systems. Moreover, these formal proof systems are strictly stronger than* PC, PCR *and Resolution, and have an exponential gap over bounded-depth Frege (for Tseitin mod* 2 *formulas, when the underlying field has characteristic different than* 2 *and the underlying graphs are appropriately expanding).*

Acknowledgements

The first author would like to thank Toni Pitassi for very helpful conversations. The second author would like to thank Nachum Dershowitz for very helpful conversations. The first author was supported by The Israel Science Foundation and The Minerva Foundation. The work of the second author was carried out as part of a Ph.D. research at Tel Aviv University and was supported in part by The Israel Science Foundation (grant no. 250/05).

References

S. AARONSON (2004). Multilinear formulas and skepticism of quantum computing. In *Proceedings of the 36th Annual ACM Symposium on the Theory of Computing*, 118–127. ACM Press, Chicago, IL.

MICHAEL ALEKHNOVICH, ELI BEN-SASSON, ALEXANDER A. RAZBOROV & AVI WIGDERSON (2000). Pseudorandom generators in propositional proof complexity. In *Proceedings of the 41st Annual Symposium on Foundations of Computer Science (Redondo Beach, CA, 2000)*, 43–53. IEEE Comput. Soc. Press, Los Alamitos, CA.

MICHAEL ALEKHNOVICH, ELI BEN-SASSON, ALEXANDER A. RAZBOROV & AVI WIGDERSON (2002). Space complexity in propositional calculus. *SIAM J. Comput.* **31**(4), 1184–1211 (electronic). ISSN 1095-7111.

MICHAEL ALEKHNOVICH & ALEXANDER A. RAZBOROV (2001). Lower bounds for polynomial calculus: non-binomial case. In *Proceedings of the 42nd IEEE Symposium on Foundations of Computer Science (Las Vegas, NV, 2001)*, 190–199. IEEE Computer Soc., Los Alamitos, CA.

NOGA ALON (1986). Eigenvalues and expanders. *Combinatorica* **6**, 83–96.

Paul Beame, Russell Impagliazzo, Jan Krajíček, Toniann Pitassi & Pavel Pudlák (1996). Lower bounds on Hilbert's Nullstellensatz and propositional proofs. *Proc. London Math. Soc. (3)* **73**(1), 1–26. ISSN 0024-6115.

Paul Beame & Toniann Pitassi (1998). Propositional proof complexity: past, present, and future. *Bull. Eur. Assoc. Theor. Comput. Sci. EATCS* (65), 66–89. ISSN 0252-9742.

Eli Ben-Sasson (2002). Hard examples for the bounded depth Frege proof system. *Comput. Complexity* **11**(3-4), 109–136. ISSN 1016-3328.

Eli Ben-Sasson & Russell Impagliazzo (1999). Random CNF's are hard for the polynomial calculus. In *Proceedings of the IEEE 40th Annual Symposium on Foundations of Computer Science (New York, 1999)*, 415–421. IEEE Computer Soc., Los Alamitos, CA.

Eli Ben-Sasson & Avi Wigderson (1999). Short proofs are narrow—resolution made simple. In *Proceedings of the 31th Annual ACM Symposium on the Theory of Computing (Atlanta, GA, 1999)*, 517–526 (electronic). ACM, New York.

Sam Buss, Dima Grigoriev, Russell Impagliazzo & Toniann Pitassi (2001). Linear gaps between degrees for the polynomial calculus modulo distinct primes. *J. Comput. System Sci.* **62**(2), 267–289. ISSN 0022-0000. Special issue on the 14th Annual IEEE Conference on Computational Complexity (Atlanta, GA, 1999).

Sam Buss, Russell Impagliazzo, Jan Krajíček, Pavel Pudlák, Alexander A. Razborov & Jiří Sgall (1996/97). Proof complexity in algebraic systems and bounded depth Frege systems with modular counting. *Comput. Complexity* **6**(3), 256–298. ISSN 1016-3328.

Matthew Clegg, Jeffery Edmonds & Russell Impagliazzo (1996). Using the Groebner basis algorithm to find proofs of unsatisfiability. In *Proceedings of the 28th Annual ACM Symposium on the Theory of Computing (Philadelphia, PA, 1996)*, 174–183. ACM, New York.

Stephen A. Cook & Robert A. Reckhow (1979). The relative efficiency of propositional proof systems. *J. Symbolic Logic* **44**(1), 36–50. ISSN 0022-4812.

Zeev Dvir & Amir Shpilka (2005). Locally Decodable Codes with 2 queries and Polynomial Identity Testing for depth 3 circuits. In *Proceedings of the 37th annual ACM symposium on Theory of computing (Baltimore, MD)*, 592–601.

Dima Grigoriev & Edward A. Hirsch (2003). Algebraic proof systems over formulas. *Theoret. Comput. Sci.* **303**(1), 83–102. ISSN 0304-3975. Logic and complexity in computer science (Créteil, 2001).

Armin Haken (1985). The intractability of resolution. *Theoret. Comput. Sci.* **39**(2-3), 297–308. ISSN 0304-3975.

Russell Impagliazzo, Pavel Pudlák & Jiří Sgall (1999). Lower bounds for the polynomial calculus and the Gröbner basis algorithm. *Comput. Complexity* **8**(2), 127–144. ISSN 1016-3328.

Neeraj Kayal & Nitin Saxena (2006). Polynomial identity testing for depth 3 circuits. In *Proceedings of the 21th Annual IEEE Conference on Computational Complexity (Prague)*, 9–17.

Jan Krajíček, Pavel Pudlák & Alan Woods (1995). An exponential lower bound to the size of bounded depth Frege proofs of the pigeonhole principle. *Random Structures Algorithms* **7**(1), 15–39. ISSN 1042-9832.

Toniann Pitassi (1997). Algebraic propositional proof systems. In *Descriptive complexity and finite models (Princeton, NJ, 1996)*, volume 31 of *DIMACS Ser. Discrete Math. Theoret. Comput. Sci.*, 215–244. Amer. Math. Soc., Providence, RI.

Toniann Pitassi, Paul Beame & Russell Impagliazzo (1993). Exponential lower bounds for the pigeonhole principle. *Comput. Complexity* **3**(2), 97–140. ISSN 1016-3328.

Ran Raz (2004a). Multi-linear formulas for permanent and determinant are of super-polynomial size. In *Proceedings of the 36th Annual ACM Symposium on the Theory of Computing*, 633–641. ACM, Chicago, IL.

Ran Raz (2004b). Multilinear-$\mathrm{NC}_1 \neq$ Multilinear-NC_2. In *Proceedings of the IEEE 45th Annual Symposium on Foundations of Computer Science*, 344–351. Rome.

Ran Raz & Amir Shpilka (2004). Deterministic polynomial identity testing in non-commutative models. In *Proceedings of the 19th Annual IEEE Conference on Computational Complexity (Amherst, MA)*, 215–222.

Alexander A. Razborov (1998). Lower bounds for the polynomial calculus. *Comput. Complexity* **7**(4), 291–324. ISSN 1016-3328.

Alexander A. Razborov (2002-2003). Pseudorandom generators hard for k-DNF resolution and polynomial calculus resolution. *Manuscript* .

Jacob T. Schwartz (1980). Fast probabilistic algorithms for verification of polynomial identities. *Journal of the ACM* **27**(4), 701–717.

Amir Shpilka & Avi Wigderson (2001). Depth-3 arithmetic circuits over fields of characteristic zero. *Comput. Complexity* **10**, 1–27.

G. C. Tseitin (1968). *On the complexity of derivations in propositional calculus*. Studies in constructive mathematics and mathematical logic Part II. Consultants Bureau, New-York-London.

Alasdair Urquhart (1987). Hard examples for resolution. *Journal of the ACM* **34**(1), 209–219.

Richard Zippel (1979). Probabilistic algorithms for sparse polynomials. In *Proceedings of the International Symposiumon on Symbolic and Algebraic Computation*. Springer-Verlag.