Annals of Pure and Applied Logic, 155(**3**):194-224, 2008

RESOLUTION OVER LINEAR EQUATIONS AND MULTILINEAR PROOFS

RAN RAZ AND IDDO TZAMERET

Abstract. We develop and study the complexity of propositional proof systems of varying strength extending resolution by allowing it to operate with disjunctions of linear equations instead of clauses. We demonstrate polynomial-size refutations for hard tautologies like the pigeonhole principle, Tseitin graph tautologies and the clique-coloring tautologies in these proof systems. Using the (monotone) interpolation by a communication game technique we establish an exponential-size lower bound on refutations in a certain, considerably strong, fragment of resolution over linear equations, as well as a general polynomial upper bound on (non-monotone) interpolants in this fragment.

We then apply these results to extend and improve previous results on multilinear proofs (over fields of characteristic 0), as studied in [RT06]. Specifically, we show the following:

- Proofs operating with depth-3 multilinear formulas polynomially simulate a certain, considerably strong, fragment of resolution over linear equations.
- Proofs operating with depth-3 multilinear formulas admit polynomial-size refutations of the pigeonhole principle and Tseitin graph tautologies. The former improve over a previous result that established small multilinear proofs only for the *functional* pigeonhole principle. The latter are different than previous proofs, and apply to multilinear proofs of Tseitin mod p graph tautologies over any field of characteristic 0.

We conclude by connecting resolution over linear equations with extensions of the cutting planes proof system.

Contents

2000 *Mathematics Subject Classification.* 03F20, 68Q17, 68Q15.

Key words and phrases. Proof complexity, resolution, algebraic proof systems, multilinear proofs, cutting planes, feasible monotone interpolation.

The first author was supported by The Israel Science Foundation and The Minerva Foundation.
The second author was supported by The Israel Science Foundation (grant no. 250/05).

1. Introduction

This paper considers two kinds of proof systems. The first kind are extensions of resolution that operate with disjunctions of linear equations with integral coefficients instead of clauses. The second kind are algebraic proof systems operating with multilinear arithmetic formulas. Proofs in both kinds of systems establish the unsatisfiability of formulas in conjunctive normal form (CNF). We are primarily concerned with connections between these two families of proof systems and with extending and improving previous results on multilinear proofs.

The resolution system is a popular propositional proof system that establishes the unsatisfiability of CNF formulas (or equivalently, the truth of tautologies in disjunctive normal form) by operating with clauses (a clause is a disjunction of propositional variables and their negations). It is well known that resolution cannot provide small (that is, polynomial-size) proofs for many basic counting arguments. The most notable example of this are the strong exponential lower bounds on the resolution refutation size of the pigeonhole principle and its different variants (Haken [Hak85] was the first to establish such a lower bound; see also [Razb02] for a survey on the proof complexity of the pigeonhole principle). Due to the popularity of resolution both in practice, as the core of many automated theorem provers, and as a theoretical case-study in propositional proof complexity, it is natural to consider weak extensions of resolution that can overcome its inefficiency in providing proofs of counting arguments. The proof systems we present in this paper are extensions of resolution, of various strength, that are suited for this purpose.

Propositional proof systems of a different nature that also attracted much attention in proof complexity theory are *algebraic proof systems*, which are proof systems operating with (multivariate) polynomials over a field. In this paper, we are particularly interested in algebraic proof systems that operate with multilinear polynomials represented as multilinear arithmetic formulas, called by the generic name *multilinear proofs* (a polynomial is *multilinear* if the power of each variable in its monomials is at most one). The investigation into such proof systems was initiated in [RT06], and here we continue this line of research. This research is motivated on the one hand by the apparent considerable strength of such systems; and on the other hand, by the known super-polynomial size lower bounds on multilinear formulas computing certain important functions [Raz04, Raz06], combined with the general working assumption that establishing lower

bounds on the size of *objects* a proof system manipulates (in this case, multilinear formulas) is close to establishing lower bounds on the size of the *proofs* themselves.

The basic proof system we shall study is denoted R(lin). The proof-lines[1] in R(lin) proofs are disjunctions of linear equations with integral coefficients over the variables $\vec{x} = x_1, \ldots, x_n$. It turns out that (already proper subsystems of) R(lin) can handle very elegantly basic counting arguments. The following defines the R(lin) proof system. Given an initial CNF, we translate every clause $\bigvee_{i\in I} x_i \vee \bigvee_{j\in J} \neg x_j$ (where I are the indices of variables with positive polarities and J are the indices of variables with negative polarities) pertaining to the CNF, into the disjunction $\bigvee_{i\in I}(x_i = 1) \vee \bigvee_{j\in J}(x_j = 0)$. Let A and B be two disjunctions of linear equations, and let $\vec{a} \cdot \vec{x} = a_0$ and $\vec{b} \cdot \vec{x} = b_0$ be two linear equations (where $\vec{a}, \vec{b}$ are two vectors of n integral coefficients, and $\vec{a}\cdot\vec{x}$ is the scalar product $\sum_{i=1}^{n} a_i x_i$; and similarly for $\vec{b}\cdot\vec{x}$). The rules of inference belonging to R(lin) allow to derive $A \vee B \vee ((\vec{a} + \vec{b}) \cdot \vec{x} = a_0 + b_0)$ from $A \vee (\vec{a} \cdot \vec{x} = a_0)$ and $B \vee (\vec{b} \cdot \vec{x} = b_0)$ (or similarly, to derive $A \vee B \vee ((\vec{a} - \vec{b}) \cdot \vec{x} = a_0 - b_0)$ from $A \vee (\vec{a} \cdot \vec{x} = a_0)$ and $B \vee (\vec{b} \cdot \vec{x} = b_0)$). We can also simplify disjunctions by discarding (unsatisfiable) equations of the form $(0 = k)$, for $k \neq 0$. In addition, for every variable x_i, we shall add an axiom $(x_i = 0) \vee (x_i = 1)$, which forces x_i to take on only Boolean values. A derivation of the empty disjunction (which stands for FALSE) from the (translated) clauses of a CNF is called an *R(lin) refutation* of the given CNF. This way, every unsatisfiable CNF has an R(lin) refutation (this can be proved by a straightforward simulation of resolution by R(lin)).

The basic idea connecting resolution operating with disjunctions of linear equations and multilinear proofs is this: Whenever a disjunction of linear equations is simple enough — and specifically, when it is close to a symmetric function, in a manner made precise — then it can be represented by a small size and small depth multilinear arithmetic formula over fields of characteristic 0. This idea was already used (somewhat implicitly) in [RT06] to obtain polynomial-size multilinear proofs operating with depth-3 multilinear formulas of the functional pigeonhole principle (this principle is weaker than the pigeonhole principle). In the current paper we generalize previous results on multilinear proofs by fully using this idea: We show how to polynomially simulate with multilinear proofs, operating with small depth multilinear formulas, certain short proofs carried inside resolution over linear equations. This enables us to provide new polynomial-size multilinear proofs for certain hard tautologies, improving results from [RT06].

More specifically, we introduce a certain fragment of R(lin), which can be polynomially simulated by depth-3 multilinear proofs (that is, multilinear proofs operating with depth-3 multilinear formulas). On the one hand this fragment of resolution over linear equations already is sufficient to formalize in a transparent way basic counting arguments, and so it admits small proofs of the pigeonhole principle and the Tseitin mod p formulas (which yields some new upper bounds on multilinear proofs); and on the other hand we can use the (monotone) interpolation technique to establish an exponential-size lower bound on refutations in this fragment as well as demonstrating a general (non-monotone) polynomial upper bound on interpolants for this fragment. The possibility that multilinear proofs (possibly, operating with depth-3 multilinear formulas) possess the feasible monotone interpolation property (and hence, admit exponential-size lower bounds) remains open.

Another family of propositional proof systems we discuss in relation to the systems mentioned above are the *cutting planes* system and its extensions. The cutting planes proof system operates with linear *inequalities* with integral coefficients, and this system is very close to the extensions of resolution we present in this paper. In particular, the following simple observation can be used to polynomially simulate cutting planes proofs with polynomially bounded coefficients (and some

[1]Each element (usually a formula) of a proof-sequence is referred to as a *proof-line*.

of its extensions) inside resolution over linear equations: The truth value of a linear inequality $\vec{a} \cdot \vec{x} \geq a_0$ (where $\vec{a}$ is a vector of n integral coefficients and $\vec{x}$ is a vector of n *Boolean* variables) is equivalent to the truth value of the following disjunction of linear equalities:

$$(\vec{a} \cdot \vec{x} = a_0) \vee (\vec{a} \cdot \vec{x} = a_0 + 1) \vee \cdots \vee (\vec{a} \cdot \vec{x} = a_0 + k) \ ,$$

where $a_0 + k$ equals the sum of all positive coefficients in $\vec{a}$ (that is, $a_0 + k = \max_{\vec{x} \in \{0,1\}^n} (\vec{a} \cdot \vec{x})$).

Note on terminology. All the proof systems considered in this paper intend to prove the *unsatisfiability* over $0, 1$ values of collections of clauses (possibly, of translation of the clauses to disjunctions of linear equations). In other words, proofs in such proof systems intend to *refute* the collections of clauses, which is to validate their negation. Therefore, throughout this paper we shall sometime speak about refutations and proofs interchangeably, always intending refutations, unless otherwise stated.

1.1. **Comparison to Earlier Work.** To the best of our knowledge this paper is the first that investigates the complexity of resolution proofs operating with disjunctions of linear *equations*. Previous works considered extensions of resolution over linear *inequalities* augmented with the cutting planes inference rules (the resulting proof system denoted R(CP)). In full generality, we show that resolution over linear equations can polynomially simulate R(CP) when the coefficients in all the inequalities are polynomially bounded (however, the converse is not known to hold). On the other hand, we shall consider a certain fragment of resolution over linear equations, in which we do not even know how to polynomially simulate cutting planes proofs with polynomially bounded coefficients in inequalities (let alone R(CP) with polynomially bounded coefficients in inequalities). We now shortly discuss the previous work on R(CP) and related proof systems.

Extensions of resolution to disjunctions of linear *inequalities* were first considered by Krajíček [Kra98] who developed the proof systems LK(CP) and R(CP). The LK(CP) system is a first-order (Gentzen-style) sequent calculus that operates with linear inequalities instead of atomic formulas and augments the standard first-order sequent calculus inference rules with the cutting planes inference rules. The R(CP) proof system is essentially resolution over linear inequalities, that is, resolution that operates with disjunctions of linear inequalities instead of clauses.

The main motivation of [Kra98] is to extend the feasible interpolation technique and consequently the lower bounds results, from cutting planes and resolution to stronger proof systems. That paper establishes an exponential-size lower bound on a restricted version of R(CP) proofs, namely, when the number of inequalities in each proof-line is $O(n^{\varepsilon})$, where n is the number of variables of the initial formulas, ε is a small enough constant and the coefficients in the cutting planes inequalities are polynomially bounded.

Other papers considering extensions of resolution over linear inequalities are the more recent papers by Hirsch & Kojevnikov [HK06] and Kojevnikov [Koj07]. The first paper [HK06] considers combinations of resolution with LP (an incomplete subsystem of cutting planes based on simple linear programming reasoning), with the 'lift and project' proof system (L&P), and with the cutting planes proof system. That paper also illustrates polynomial-size refutations of the Tseitin mod 2 tautologies in all these extensions of resolution. The second paper [Koj07] deals with improving the parameters of the tree-like R(CP) lower-bounds obtained in [Kra98]. Also, on the more practical level, Hirsch et al. [HIK+05] have developed an experimental SAT-solver (that is, a software tool that receives a CNF formula and outputs a satisfying assignment if there is any; and outputs FALSE, otherwise) named *basolver* (which stands for mixed Boolean-Algebraic Solver). This SAT-solver solves CNF formulas (and also checks Boolean circuits for equivalence) by translating them first into systems of polynomial equations and disjunctions of *polynomial* equations, and then solving these systems by means of derivation rules in the spirit of the resolution derivation rules.

Whereas previous results concerned primarily with extending the cutting planes proof system, our foremost motivation is to extend and improve previous results on algebraic proof systems operating with multilinear formulas obtained in [RT06]. In that paper the concept of multilinear proofs was introduced and several basic results concerning multilinear proofs were proved. In particular, polynomial-size proofs of two important combinatorial principles were demonstrated: the functional pigeonhole principle and the Tseitin (mod p) graph tautologies. In the current paper we improve both these results.

As mentioned above, motivated by relations with multilinear proofs operating with depth-3 multilinear formulas, we shall consider a certain subsystem of resolution over linear equations. For this subsystem we apply twice the interpolation by a communication game technique. The first application is of the *non*-monotone version of the technique, and the second application is of the *monotone* version. Namely, the first application provides a general (non-monotone) interpolation theorem that demonstrates a polynomial (in the size of refutations) upper bound on interpolants; The proof uses the general method of transforming a refutation into a Karchmer-Wigderson communication game for two players, from which a Boolean circuit is then attainable. In particular, we shall apply the interpolation theorem of Krajíček from [Kra97]. The second application of the (monotone) interpolation by a communication game technique is implicit and proceeds by using the lower bound criterion of Bonet, Pitassi & Raz in [BPR97]. This criterion states that (semantic) proof systems (of a certain natural and standard kind) whose proof-lines (considered as Boolean functions) have low communication complexity cannot prove efficiently a certain tautology (namely, the clique-coloring tautologies).

1.2. **Summary of Results.** This paper introduces and connects several new concepts and ideas with some known ones. It identifies new extensions of resolution operating with linear equations, and relates (a certain) such extension to multilinear proofs. The upper bounds for the pigeonhole principle and Tseitin mod p formulas in fragments of resolution over linear equations are new. By generalizing the machinery developed in [RT06], these upper bounds yield new and improved results concerning multilinear proofs. The lower bound for the clique-coloring formulas in a fragment of resolution over linear equations employs the standard monotone interpolation by a communication game technique, and specifically utilizes the theorem of Bonet, Pitassi & Raz from [BPR97]. The general (non-monotone) interpolation result for a fragment of resolution over linear equations employs the theorem of Krajíček from [Kra97]. The upper bound in (the stronger variant of – as described in the introduction) resolution over linear equations of the clique-coloring formulas follows that of Atserias, Bonet & Esteban [ABE02]. We now give a detailed outline of the results in this paper.

The proof systems. In Section 3 we formally define two extensions of resolution of decreasing strength allowing resolution to operate with disjunctions of linear equations. The size of a linear equation $a_1x_1 + \ldots + a_nx_n = a_0$ is the sum of all $a_0, \ldots, a_n$ written in *unary notation*. The size of a disjunction of linear equations is the total size of all linear equations in the disjunction. The size of a proof operating with disjunctions of linear equations is the total size of all the disjunctions in it.

R(lin): This is the stronger proof system (described in the introduction) that operates with disjunctions of linear equations with integer coefficients.

R^0(lin): This is a (provably proper) fragment of R(lin). It operates with disjunctions of (arbitrarily many) linear equations whose variables have constant coefficients, under the restriction that every disjunction can be partitioned into a constant number of sub-disjunctions, where each sub-disjunction either consists of linear equations that differ only in their free-terms or is a (translation of a) clause.

Note that any single linear *inequality* with Boolean variables can be represented by a disjunction of linear equations that differ only in their free-terms (see the example in the introduction section). So the $\mathrm{R}^0(\mathrm{lin})$ proof system is close to a proof system operating with disjunctions of constant number of linear inequalities (with constant integral coefficients). In fact, disjunctions of linear equations varying only in their free-terms, have more (expressive) strength than a single inequality. For instance, the PARITY function can be easily represented by a disjunction of linear equations, while it cannot be represented by a single linear inequality.

As already mentioned, the motivation to consider the restricted proof system $\mathrm{R}^0(\mathrm{lin})$ comes from its relation to multilinear proofs operating with depth-3 multilinear formulas (in short, depth-3 multilinear proofs): $\mathrm{R}^0(\mathrm{lin})$ corresponds roughly to the subsystem of $\mathrm{R}(\mathrm{lin})$ that we know how to simulate by depth-3 multilinear proofs via the technique in [RT06] (the technique is based on converting disjunctions of linear forms into symmetric polynomials, which are known to have small depth-3 multilinear formulas). This simulation is then applied in order to improve over known upper bounds for depth-3 multilinear proofs, as $\mathrm{R}^0(\mathrm{lin})$ is already sufficient to efficiently prove certain "hard tautologies". Moreover, we are able to establish an exponential lower bound on $\mathrm{R}^0(\mathrm{lin})$ refutations size (see below for both upper and lower bounds on $\mathrm{R}^0(\mathrm{lin})$ proofs). We also establish a super-polynomial separation of $\mathrm{R}(\mathrm{lin})$ from $\mathrm{R}^0(\mathrm{lin})$ (via the clique-coloring principle, for a certain choice of parameters; see below).

Short refutations. We demonstrate the following short refutations in $\mathrm{R}^0(\mathrm{lin})$ and $\mathrm{R}(\mathrm{lin})$:

(1) Polynomial-size refutations of the pigeonhole principle in $\mathrm{R}^0(\mathrm{lin})$;
(2) Polynomial-size refutations of Tseitin mod p graph formulas in $\mathrm{R}^0(\mathrm{lin})$;
(3) Polynomial-size refutations of the clique-coloring formulas in $\mathrm{R}(\mathrm{lin})$ (for certain parameters). The refutations here follow by direct simulation of the Res(2) refutations of clique-coloring formulas from [ABE02].

All the three families of formulas above are prominent "hard tautologies" in proof complexity literature, which means that strong size lower bounds on proofs in various proof systems are known for them (for the exact formulation of these families of formulas see Section 6).

Interpolation results. We provide a polynomial upper-bound on (non-monotone) interpolants corresponding to $\mathrm{R}^0(\mathrm{lin})$ refutations; Namely, we show that any $\mathrm{R}^0(\mathrm{lin})$-refutation of a given formula can be transformed into a (non-monotone) Boolean circuit computing the corresponding interpolant function of the formula (if there exists such a function), with at most a polynomial increase in size. We employ the general interpolation theorem of Krajíček [Kra97] for semantic proof systems.

Lower bounds. We provide the following exponential lower bound:

Theorem 1. $\mathrm{R}^0(\mathrm{lin})$ *does not have sub-exponential refutations for the clique-coloring formulas.*

This result is proved by applying a result of Bonet, Pitassi & Raz [BPR97]. The result in [BPR97] (implicitly) use the monotone interpolation by a communication game technique for establishing an exponential-size lower bound on refutations of general semantic proof systems operating with proof-lines of low communication complexity.

Applications to multilinear proofs. Multilinear proof systems are (semantic) refutation systems operating with multilinear polynomials over a fixed field, where every multilinear polynomial is represented by a multilinear arithmetic formula. In this paper we shall consider multilinear formulas over fields of characteristic 0 only. The *size* of a multilinear proof (that is, a proof in a multilinear proof system) is the total size of all multilinear formulas in the proof (for formal definitions concerning multilinear proofs see Section 9).

We shall first connect multilinear proofs with resolution over linear equations by the following result:

Theorem 2. *Multilinear proofs operating with depth-3 multilinear formulas over characteristic 0 polynomially-simulate* $\mathrm{R}^0(\mathrm{lin})$.

An immediate corollary of this theorem and the upper bounds in $\mathrm{R}^0(\mathrm{lin})$ described above are polynomial-size multilinear proofs for the pigeonhole principle and the Tseitin mod p formulas.

(1) Polynomial-size depth-3 multilinear refutations for the pigeonhole principle over fields of characteristic 0. This improves over [RT06] that shows a similar upper bound for a weaker principle, namely, the *functional* pigeonhole principle.
(2) Polynomial-size depth-3 multilinear refutations for the Tseitin mod p graph formulas over fields of characteristic 0. These refutations are different than those demonstrated in [RT06], and further they establish short multilinear refutations of the Tseitin mod p graph formulas over *any field of characteristic* 0 (the proof in [RT06] showed how to refute the Tseitin mod p formulas by multilinear refutations only over fields that contain a primitive pth root of unity).

Relations with cutting planes proofs. As mentioned in the introduction, a proof system combining resolution with cutting planes was presented by Krajíček in [Kra98]. The resulting system is denoted R(CP) (see Section 10 for a definition). When the coefficients in the linear inequalities inside R(CP) proofs are polynomially bounded, the resulting proof system is denoted R(CP*). We establish the following simulation result:

Theorem 3. R(lin) *polynomially simulates resolution over cutting planes inequalities with polynomially bounded coefficients* R(CP*).

We do not know if the converse also holds.

2. Notation and Background on Propositional Proof Systems

For a natural number n, we use $[n]$ to denote $\{1,\ldots,n\}$. For a vector of n (integral) coefficients $\vec{a}$ and a vector of n variables $\vec{x}$, we denote by $\vec{a}\cdot\vec{x}$ the scalar product $\sum_{i=1}^{n} a_i x_i$. If $\vec{b}$ is another vector (of length n), then $\vec{a}+\vec{b}$ denotes the addition of $\vec{a}$ and $\vec{b}$ as vectors, and $c\vec{a}$ (for an integer c) denotes the product of the scalar c with $\vec{a}$ (where, $-\vec{a}$ denotes $-1\vec{a}$). For two linear equations $L_1 : \vec{a}\cdot\vec{x} = a_0$ and $L_2 : \vec{b}\cdot\vec{x} = b_0$, their addition $(\vec{a}+\vec{b})\cdot\vec{x} = a_0+b_0$ is denoted L_1+L_2 (and their subtraction $(\vec{a}-\vec{b})\cdot\vec{x} = a_0-b_0$ is denoted L_1-L_2). For two Boolean assignments (identified as $0,1$ strings) $\alpha,\alpha' \in \{0,1\}^n$ we write $\alpha' \geq \alpha$ if $\alpha'_i \geq \alpha_i$, for all $i \in [n]$ (where α_i, α'_i are the ith bits of α and α', respectively).

We now recall some basic concepts on propositional proof systems. For background on algebraic proof systems (and specifically multilinear proofs) see Section 9.

Resolution. In order to put our work in context, we need to define the resolution refutation system.

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. A *CNF formula* is a conjunction of clauses. The *size of a clause* is the number of literals in it.

Resolution is a complete and sound proof system for unsatisfiable CNF formulas. Let C and D be two clauses containing neither x_i nor $\neg x_i$, the *resolution rule* allows one 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, and we also say that $C \vee x_i$ and $D \vee \neg x_i$ were *resolved over* x_i. The *weakening rule* allows to derive the clause $C \vee D$ from the clause C, for any two clauses C, D.

Definition 2.1 (**Resolution**)**.** A *resolution proof of the clause D from a CNF formula K* is a sequence of clauses $D_1, D_2, \ldots, D_\ell$, such that: (1) each clause D_j is either a clause of K or

a resolvent of two previous clauses in the sequence or derived by the weakening rule from a previous clause in the sequence; (2) the last clause $D_\ell = D$. The *size* of a resolution proof is the sum of all the sizes of the clauses in it. A *resolution refutation* of a CNF formula K is a resolution proof of the empty clause $\square$ from K (the empty clause stands for FALSE; that is, the empty clause has no satisfying assignments).

A proof in resolution (or any of its extensions) is also called a *derivation* or a *proof-sequence*. Each sequence-element in a proof-sequence is also called a *proof-line*. A proof-sequence containing the proof-lines $D_1, \ldots, D_\ell$ is also said to be a *derivation of* $D_1, \ldots, D_\ell$.

Cook-Reckhow proof systems. Following [CR79], a *Cook-Reckhow proof system* is a polynomial-time algorithm A that receives a Boolean formula F (for instance, 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. The *completeness* of a (Cook-Reckhow) proof system (with respect to the set of all unsatisfiable Boolean formulas; or for a subset of it, e.g. the set of unsatisfiable CNF formulas) stands for the fact that every unsatisfiable formula F has a string π ("the refutation of F") so that $A(F, \pi) = 1$. The *soundness* of a (Cook-Reckhow) proof system stands for the fact that every formula F so that $A(F, \pi) = 1$ for some string π is unsatisfiable (in other words, no satisfiable formula has a refutation).

For instance, resolution is a Cook-Reckhow proof system, since it is complete and sound for the set of unsatisfiable CNF formulas, and given a CNF formula F and a string π it is easy to check in polynomial-time (in *both* F *and* π) whether π constitutes a resolution refutation of F.

We shall also consider proof systems that are not necessarily (that is, not known to be) Cook-Reckhow proof systems. Specifically, multilinear proof systems (over large enough fields) meet the requirements in the definition of Cook-Reckhow proof systems, *except* that the condition on A above is relaxed: we allow A to be in *probabilistic* polynomial-time **BPP** (which is not known to be equal to deterministic polynomial-time).

Polynomial simulations of proof systems. When comparing the strength of different proof systems we shall confine ourselves to CNF formulas only. That is, we consider propositional proof systems as proof systems for the set of unsatisfiable CNF formulas. For that purpose, if a proof system does not operate with clauses directly, then we fix a (direct) translation from clauses to the objects operated by the proof system. This is done for both resolution over linear equations (which operate with disjunctions of linear equations) and its fragments, and also for multilinear proofs (which operate with multilinear polynomials, represented as multilinear formulas); see for example Subsection 3.1 for such a direct translation.

Definition 2.2. Let $\mathcal{P}_1, \mathcal{P}_2$ be two proof systems for the set of unsatisfiable CNF formulas (we identify a CNF formula with its corresponding translation, as discussed above). We say that $\mathcal{P}_2$ *polynomially simulates* $\mathcal{P}_1$ if given a $\mathcal{P}_1$ refutation π of a CNF formula F, then there exists a refutation of F in $\mathcal{P}_2$ of size polynomial in the size of π. In case $\mathcal{P}_2$ polynomially simulates $\mathcal{P}_1$ while $\mathcal{P}_1$ does not polynomially simulates $\mathcal{P}_2$ we say that $\mathcal{P}_2$ is *strictly stronger* than $\mathcal{P}_1$.

3. Resolution over Linear Equations and its Subsystems

The proof systems we consider in this section are extensions of resolution. Proof-lines in resolution are clauses. Instead of this, the extensions of resolution we consider here operate with disjunctions of linear equations with integral coefficients. For this section we use the convention that all the formal variables in the propositional proof systems considered are taken from the set $X := \{x_1, \ldots, x_n\}$.

3.1. Disjunctions of Linear Equations.

For L a linear equation $a_1x_1 + \ldots + a_nx_n = a_0$, the right hand side a_0 is called the *free-term* of L and the left hand side $a_1x_1 + \ldots + a_nx_n$ is called the

linear form of L (the linear form can be 0). A *disjunction of linear equations* is of the following general form:

$$\left(a_1^{(1)}x_1 + \ldots + a_n^{(1)}x_n = a_0^{(1)}\right) \vee \cdots \vee \left(a_1^{(t)}x_1 + \ldots + a_n^{(t)}x_n = a_0^{(t)}\right), \tag{1}$$

where $t \geq 0$ and the coefficients $a_i^{(j)}$ are integers (for all $0 \leq i \leq n,\ 1 \leq j \leq t$). We discard duplicate linear equations from a disjunction of linear equations. The semantics of such a disjunction is the natural one: We say that an assignment of integral values to the variables $x_1, ..., x_n$ *satisfies* (1) if and only if there exists $j \in [t]$ so that the equation $a_1^{(j)}x_1 + \ldots + a_n^{(j)}x_n = a_0^{(j)}$ holds under the given assignment.

The symbol $\models$ denotes the *semantic implication* relation, that is, for every collection $D_1, \ldots, D_m$ of disjunctions of linear equations,

$$D_1, \ldots, D_m \models D_0$$

means that every assignment of $0, 1$ values that satisfies all $D_1, \ldots, D_m$ also satisfies D_0.[2] In this case we also say that $D_1, \ldots, D_m$ *semantically imply* D_0.

The *size of a linear equation* $a_1x_1 + \ldots + a_nx_n = a_0$ is $\sum_{i=0}^{n} |a_i|$, i.e., the sum of the bit sizes of all a_i written in *unary* notation. Accordingly, the *size of the linear form* $a_1x_1 + \ldots + a_nx_n$ is $\sum_{i=1}^{n} |a_i|$. The *size of a disjunction of linear equations* is the total size of all linear equations in it.

Since all linear equations considered in this paper are of integral coefficients, we shall speak of *linear equations* when we actually mean linear equations with integral coefficients. Similar to resolution, the *empty disjunction* is unsatisfiable and stands for the truth value FALSE.

Translation of clauses. As described in the introduction, we can translate any CNF formula to a collection of disjunctions of linear equations in a direct manner: Every clause $\bigvee_{i\in I} x_i \vee \bigvee_{j\in J} \neg x_j$ (where I and J are sets of indices of variables) pertaining to the CNF is translated into the disjunction $\bigvee_{i\in I}(x_i = 1) \vee \bigvee_{j\in J}(x_j = 0)$. For a clause D we denote by $\widetilde{D}$ its translation into a disjunction of linear equations. It is easy to verify that any Boolean assignment to the variables $x_1, \ldots, x_n$ satisfies a clause D if and only if it satisfies $\widetilde{D}$ (where TRUE is treated as 1 and FALSE as 0).

3.2. **Resolution over Linear Equations – R(lin).** Defined below is our basic proof system R(lin) that enables resolution to reason with disjunctions of linear equations. As we wish to reason about Boolean variables we augment the system with the axioms $(x_i = 0) \vee (x_i = 1)$, for all $i \in [n]$, called the *Boolean axioms*.

Definition 3.1 (R(lin)). Let $K := \{K_1, \ldots, K_m\}$ be a collection of disjunctions of linear equations. An *R(lin)-proof from K of a disjunction of linear equations D* is a finite sequence $\pi = (D_1, ..., D_\ell)$ of disjunctions of linear equations, such that $D_\ell = D$ and for every $i \in [\ell]$, either $D_i = K_j$ for some $j \in [m]$, or D_i is a Boolean axiom $(x_h = 0) \vee (x_h = 1)$ for some $h \in [n]$, or D_i was deduced by one of the following R(lin)-inference rules, using D_j, D_k for some $j, k < i$:

Resolution: Let A, B be two disjunctions[3] of linear equations and let L_1, L_2 be two linear equations.

From $A \vee L_1$ and $B \vee L_2$ derive $A \vee B \vee (L_1 + L_2)$.

Similarly, from $A \vee L_1$ and $B \vee L_2$ derive $A \vee B \vee (L_1 - L_2)$.

[2] Alternatively, we can consider assignments of any integral values (instead of only Boolean values) to the variables in $D_1, \ldots, D_m$, stipulating that the collection $D_1, \ldots, D_m$ contains all disjunctions of the form $(x_j = 0) \vee (x_j = 1)$ for all the variables $x_j \in X$ (these formulas force any satisfying assignment to give only $0, 1$ values to the variables).

Weakening: From a disjunction of linear equations A derive $A \vee L$, where L is an arbitrary linear equation over X.
Simplification: From $A \vee (0 = k)$ derive A, where A is a disjunction of linear equations and $k \neq 0$.

An *R(lin) refutation of* a collection of disjunctions of linear equations K is a proof of the empty disjunction from K. The *size* of an R(lin)-proof π is the total size of all the disjunctions of linear equations in π, denoted $|\pi|$.

Similar to resolution, in case $A \vee B \vee (L_1 + L_2)$ is derived from $A \vee L_1$ and $B \vee L_2$ by the resolution rule, we say that $A \vee L_1$ and $B \vee L_2$ were *resolved over* L_1 *and* L_2, respectively, and we call $A \vee B \vee (L_1 + L_2)$ the *resolvent* of $A \vee L_1$ and $B \vee L_2$ (and similarly, when $A \vee B \vee (L_1 - L_2)$ is derived from $A \vee L_1$ and $B \vee L_2$ by the resolution rule; we use the same terminology for both addition and subtraction, and it should be clear from the context which operation is actually applied). We also describe such an application of the resolution rule by saying that L_1 *was added (resp., subtracted) to (resp. from)* L_2 *in* $A \vee L_1$ *and* $B \vee L_2$.

In light of the direct translation between CNF formulas and collections of disjunctions of linear equations (described in the previous subsection), we can consider R(lin) to be a proof system for the set of unsatisfiable CNF formulas:

Proposition 1. *The R(lin) refutation system is a sound and complete Cook-Reckhow (see Section 2) refutation system for unsatisfiable CNF formulas (translated into unsatisfiable collection of disjunctions of linear equations).*

Proof: Completeness of R(lin) (for the set of unsatisfiable CNF formulas) stems from a straightforward simulation of resolution, as we now show.

Claim 1. R(lin) polynomially simulates resolution.

Proof of claim: Proceed by induction on the length of the resolution refutation to show that any resolution derivation of a clause A can be translated with only a linear increase in size into an R(lin) derivation of the corresponding disjunction of linear equations $\widetilde{A}$ (see the previous subsection for the definition of $\widetilde{A}$).

The base case: An initial clause A is translated into its corresponding disjunction of linear equations $\widetilde{A}$.

The induction step: If a resolution clause $A \vee B$ was derived by the resolution rule from $A \vee x_i$ and $B \vee \neg x_i$, then in R(lin) we subtract $(x_i = 0)$ from $(x_i = 1)$ in $\widetilde{B} \vee (x_i = 0)$ and $\widetilde{A} \vee (x_i = 1)$, respectively, to obtain $\widetilde{A} \vee \widetilde{B} \vee (0 = 1)$. Then, using the Simplification rule, we can cut-off $(0 = 1)$ from $\widetilde{A} \vee \widetilde{B} \vee (0 = 1)$, and arrive at $\widetilde{A} \vee \widetilde{B}$.

If a clause $A \vee B$ was derived in resolution from A by the Weakening rule, then we derive $\widetilde{A} \vee \widetilde{B}$ from $\widetilde{A}$ by the Weakening rule in R(lin). ∎

Soundness of R(lin) stems from the soundness of the inference rules (which means that: If D was derived from C, B by the R(lin) resolution rule then any assignment that satisfies both C and B also satisfies D; and if D was derived from C by either the Weakening rule or the Simplification rule, then any assignment that satisfies C also satisfies D).

The R(lin) proof system is a Cook-Reckhow proof system, as it is easy to verify in polynomial-time whether an R(lin) proof-line is inferred, by an application of one of R(lin)'s inference rules, from a previous proof-line (or proof-lines). Thus, any sequence of disjunctions of linear equations,

[3]Possibly the empty disjunction. This remark also applies to the inference rules below.

can be checked in polynomial-time (in the size of the sequence) to decide whether or not it is a legitimate R(lin) proof-sequence. ∎

In Section 5 we shall see that a stronger notion of completeness (that is, implicational completeness) holds for R(lin) and its subsystems.

3.3. **Fragment of Resolution over Linear Equations – $\mathbf{R^0(lin)}$.** Here we consider a restriction of R(lin), denoted R^0(lin). As discussed in the introduction section, R^0(lin) is roughly the fragment of R(lin) we know how to polynomially simulate with depth-3 multilinear proofs.

By results established in the sequel (Sections 6.3 and 8) R(lin) is *strictly stronger* than R^0(lin), which means that R(lin) polynomially simulates R^0(lin), while the converse does not hold.

R^0(lin) operates with disjunctions of (arbitrarily many) linear equations with constant coefficients (excluding the free terms), under the following restriction: Every disjunction can be partitioned into a constant number of sub-disjunctions, where each sub-disjunction either consists of linear equations that differ only in their free-terms or is a (translation of a) clause.

As mentioned in the introduction, every linear *inequality* with Boolean variables can be represented by a disjunction of linear equations that differ only in their free-terms. So the R^0(lin) proof system resembles, to some extent, a proof system operating with disjunctions of constant number of linear inequalities with constant integral coefficients (on the other hand, it is probable that R^0(lin) is stronger than such a proof system, as a disjunction of linear equations that differ only in their free terms is [expressively] stronger than a linear inequality: the former can define the PARITY function while the latter cannot).

Example of an R^0(lin)*-line:*

$$(x_1 + \ldots + x_\ell = 1) \vee \cdots \vee (x_1 + \ldots + x_\ell = \ell) \vee (x_{\ell+1} = 1) \vee \cdots \vee (x_n = 1),$$

for some $1 \leq \ell \leq n$. The next section contains other concrete (and natural) examples of R^0(lin)-lines.

Let us define formally what it means to be an R^0(lin) proof-line, that is, a proof-line inside an R^0(lin) proof, called R^0(lin)*-line*:

Definition 3.2 ($\mathbf{R^0}$(lin)-line)**.** Let D be a disjunction of linear equations whose variables have constant integer coefficients (the free-terms are unbounded). Assume D can be partitioned into a constant number k of sub-disjunctions $D_1, \ldots, D_k$, where each D_i either consists of (an unbounded) disjunction of linear equations that differ only in their free-terms, or is a translation of a clause (as defined in Subsection 3.1). Then the disjunction D is called an R^0(lin)*-line*.

Thus, any R^0(lin)-line is of the following general form:

$$\bigvee_{i \in I_1} \left(\vec{a}^{(1)} \cdot \vec{x} = \ell_i^{(1)} \right) \vee \cdots \vee \bigvee_{i \in I_k} \left(\vec{a}^{(k)} \cdot \vec{x} = \ell_i^{(k)} \right) \vee \bigvee_{j \in J} (x_j = b_j) \,, \tag{2}$$

where k and all $a_r^{(t)}$ (for $r \in [n]$ and $t \in [k]$) are integer constants and $b_j \in \{0, 1\}$ (for all $j \in J$) (and $I_1, \ldots, I_k, J$ are unbounded sets of indices). Note that a disjunction of clauses can be combined into a single clause. Hence, without loss of generality we can assume that in any R^0(lin)-line only a single (translation of a) clause occurs. This is depicted in (2) (where in addition we have ignored in (2) the possibility that the single clause obtained by combining several clauses contains $x_j \vee \neg x_j$, for some $j \in [n]$).

Definition 3.3 ($\mathbf{R^0}$(lin))**.** The R^0(lin) proof system is a restriction of the R(lin) proof system in which each proof-line is an R^0(lin)-line (as in Definition 3.2).

For a completeness proof of $\mathrm{R}^0(\mathrm{lin})$ see Section 5.[4]

4. Reasoning and Counting inside R(lin) and its Subsystems

In this section we illustrate a simple way to reason by case-analysis inside R(lin) and its subsystems. This kind of reasoning will simplify the presentation of proofs inside R(lin) (and $\mathrm{R}^0(\mathrm{lin})$) in the sequel (essentially, a similar – though weaker – kind of reasoning is applicable already in resolution). We will then demonstrate efficient and transparent proofs for simple counting arguments that will also facilitate us in the sequel.

4.1. Basic Reasoning inside R(lin) and its Subsystems.

Given K a collection of disjunctions of linear equations $\{K_1, \ldots, K_m\}$ and C a disjunction of linear equations, denote by $K \vee C$ the collection $\{K_1 \vee C, \ldots, K_m \vee C\}$. Recall that the formal variables in our proof system are $x_1, \ldots, x_n$.

Lemma 4. *Let K be a collection of disjunctions of linear equations, and let z abbreviate some linear form with integer coefficients. Let $E_1, \ldots, E_\ell$ be ℓ disjunctions of linear equations. Assume that for all $i \in [\ell]$ there is an R(lin) derivation of E_i from $z = a_i$ and K with size at most s where $a_1, \ldots, a_\ell$ are distinct integers. Then, there is an R(lin) proof of $\bigvee_{i=1}^{\ell} E_i$ from K and $(z = a_1) \vee \cdots \vee (z = a_\ell)$, with size polynomial in s and ℓ.*

Proof: Denote by D the disjunction $(z = a_1) \vee \cdots \vee (z = a_\ell)$ and by π_i the R(lin) proof of E_i from K and $z = a_i$ (with size at most s), for all $i \in [\ell]$. It is easy to verify that for all $i \in [\ell]$ the sequence $\pi_i \vee \bigvee_{j \in [\ell] \setminus \{i\}} (z = a_j)$ is an R(lin) proof of $E_i \vee \bigvee_{j \in [\ell] \setminus \{i\}} (z = a_j)$ from K and D. So overall, given D and K as premises, there is an R(lin) derivation of size polynomial in s and ℓ of the following collection of disjunctions of linear equations:

$$E_1 \vee \bigvee_{j \in [\ell] \setminus \{1\}} (z = a_j), \ldots, E_\ell \vee \bigvee_{j \in [\ell] \setminus \{\ell\}} (z = a_j)\,. \tag{3}$$

We now use the Resolution rule to cut-off all the equations $(z = a_i)$ inside all the disjunctions in (3). Formally, we prove that for every $1 \leq k \leq \ell$ there is a polynomial-size (in s and ℓ) R(lin) derivation from (3) of

$$E_1 \vee \cdots \vee E_k \vee \bigvee_{j \in [\ell] \setminus [k]} (z = a_j)\,, \tag{4}$$

and so putting $k = \ell$, will conclude the proof of the lemma.

We proceed by induction on k. The base case for $k = 1$ is immediate (from (3)). For the induction case, assume that for some $1 \leq k < \ell$ we already have an R(lin) proof of (4), with size polynomial in s and ℓ.

Consider the line

$$E_{k+1} \vee \bigvee_{j \in [\ell] \setminus \{k+1\}} (z = a_j)\,. \tag{5}$$

We can now cut-off the disjunctions $\bigvee_{j \in [\ell] \setminus [k]} (z = a_j)$ and $\bigvee_{j \in [\ell] \setminus \{k+1\}} (z = a_j)$ from (4) and (5), respectively, using the Resolution rule (since the a_j's in (4) and in (5) are disjoint). We will demonstrate this derivation in some detail now, in order to exemplify a proof carried inside R(lin). We shall be less formal sometime in the sequel.

[4]The simulation of resolution inside R(lin) (in the proof of Proposition 1) is carried on with each R(lin) proof-line being in fact a translation of a clause, and hence, an $\mathrm{R}^0(\mathrm{lin})$-line (notice that the Boolean axioms of R(lin) are $\mathrm{R}^0(\mathrm{lin})$-lines). This already implies that $\mathrm{R}^0(\mathrm{lin})$ is a complete refutation system for the set of unsatisfiable CNF formulas. In section 5 we give a proof of a stronger notion of completeness for $\mathrm{R}^0(\mathrm{lin})$.

Resolve (4) with (5) over $(z = a_{k+1})$ and $(z = a_1)$, respectively, to obtain

$$(0 = a_1 - a_{k+1}) \vee E_1 \vee \cdots \vee E_k \vee E_{k+1} \vee \bigvee_{j \in [\ell] \setminus \{1,k+1\}} (z = a_j)\,. \tag{6}$$

Since $a_1 \neq a_{k+1}$, we can use the Simplification rule to cut-off $(0 = a_1 - a_{k+1})$ from (6), and we arrive at

$$E_1 \vee \cdots \vee E_k \vee E_{k+1} \vee \bigvee_{j \in [\ell] \setminus \{1,k+1\}} (z = a_j)\,. \tag{7}$$

Now, similarly, resolve (4) with (7) over $(z = a_{k+1})$ and $(z = a_2)$, respectively, and use Simplification to obtain

$$E_1 \vee \cdots \vee E_k \vee E_{k+1} \vee \bigvee_{j \in [\ell] \setminus \{1,2,k+1\}} (z = a_j)\,.$$

Continue in a similar manner until you arrive at

$$E_1 \vee \cdots \vee E_k \vee E_{k+1} \vee \bigvee_{j \in [\ell] \setminus \{1,2,\ldots,k,k+1\}} (z = a_j)\,,$$

which is precisely what we need. ∎

Under the appropriate conditions, Lemma 4 also holds for $\mathrm{R}^0(\mathrm{lin})$ proofs. This is stated in the following lemma.

Lemma 5. *Let K be a collection of disjunctions of linear equations, and let z abbreviate a linear form with integer coefficients. Let $E_1, \ldots, E_\ell$ be ℓ disjunctions of linear equations. Assume that for all $i \in [\ell]$ there is an $\mathrm{R}^0(\mathrm{lin})$ derivation of E_i from $z = a_i$ and K with size at most s, where the a_i's are distinct integers. Then, assuming $\bigvee_{i=1}^{\ell} E_i$ is an $\mathrm{R}^0(\mathrm{lin})$-line, there is an $\mathrm{R}^0(\mathrm{lin})$ proof of $\bigvee_{i=1}^{\ell} E_i$ from K and $(z = a_1) \vee \cdots \vee (z = a_\ell)$, with size polynomial in s and ℓ.*

Proof: It can be verified by simple inspection that, under the conditions spelled out in the statement of the lemma, each proof-line in the R(lin) derivations in the proof of Lemma 4 is actually an $\mathrm{R}^0(\mathrm{lin})$-line.[5] ∎

Abbreviations. Lemmas 4 and 5 will sometime facilitate us to proceed inside R(lin) and $\mathrm{R}^0(\mathrm{lin})$ with a slightly less formal manner. For example, the situation in Lemma 4 above can be depicted by saying that "if $z = a_i$ implies E_i (with a polynomial-size proof) for all $i \in [\ell]$, then $\bigvee_{i=1}^{\ell}(z = a_i)$ implies $\bigvee_{i=1}^{\ell} E_i$ (with a polynomial-size proof)".

In case $\bigvee_{i=1}^{\ell}(z = a_i)$ above is just the *Boolean axiom* $(x_i = 0) \vee (x_i = 1)$, for some $i \in [n]$, and $x_i = 0$ implies E_0 and $x_i = 1$ implies E_1 (both with polynomial-size proofs), then to simplify the writing we shall sometime not mention the Boolean axiom at all. For example, the latter situation can be depicted by saying that "if $x_i = 0$ implies E_0 with a polynomial-size proof and $x_i = 1$ implies E_1 with a polynomial-size proof, then we can derive $E_0 \vee E_1$ with a polynomial-size proof".

4.2. **Basic Counting inside R(lin) and R^0(lin).** In this subsection we illustrate how to efficiently prove several basic counting arguments inside R(lin) and $\mathrm{R}^0(\mathrm{lin})$. This will facilitate us in showing short proofs for hard tautologies in the sequel. In accordance with the last paragraph in the previous subsection, we shall carry the proofs inside R(lin) and $\mathrm{R}^0(\mathrm{lin})$ with a slightly less rigor.

[5]Note that when the proofs of E_i from $z = a_i$, for all $i \in [\ell]$, are all done inside $\mathrm{R}^0(\mathrm{lin})$, then the linear form z ought to have *constant* coefficients.

Lemma 6. *Let z_1 abbreviate $\vec{a}\cdot\vec{x}$ and z_2 abbreviate $\vec{b}\cdot\vec{x}$. Let D_1 be $\bigvee_{\alpha\in\mathcal{A}}(z_1=\alpha)$ and let D_2 be $\bigvee_{\beta\in\mathcal{B}}(z_2=\beta)$, where $\mathcal{A},\mathcal{B}$ are two (finite) sets of integers. Then there is a polynomial-size (in the size of D_1, D_2) R(lin) proof from D_1, D_2 of:*

$$\bigvee_{\alpha\in\mathcal{A},\beta\in\mathcal{B}}(z_1+z_2=\alpha+\beta)\ . \tag{8}$$

Moreover, if $\vec{a}$ and $\vec{b}$ consist of constant *integers (which means that D_1, D_2 are $\mathrm{R}^0(\mathrm{lin})$-lines), then there is a polynomial-size (in the size of D_1, D_2) $\mathrm{R}^0(\mathrm{lin})$ proof of (8) from D_1, D_2.*

Proof: Denote the elements of $\mathcal{A}$ by $\alpha_1,\ldots,\alpha_k$. In case $z_1=\alpha_i$, for some $i\in[k]$ then we can add $z_1=\alpha_i$ to every equation in $\bigvee_{\beta\in\mathcal{B}}(z_2=\beta)$ to get $\bigvee_{\beta\in\mathcal{B}}(z_1+z_2=\alpha_i+\beta)$. Therefore, there exist k R(lin) proofs, each with polynomial-size (in $|D_1|$ and $|D_2|$), of

$$\bigvee_{\beta\in\mathcal{B}}(z_1+z_2=\alpha_1+\beta)\,,\ \bigvee_{\beta\in\mathcal{B}}(z_1+z_2=\alpha_2+\beta)\,,\quad\ldots\quad,\bigvee_{\beta\in\mathcal{B}}(z_1+z_2=\alpha_k+\beta)$$

from $z_1=\alpha_1$, $z_1=\alpha_2$,...,$z_1=\alpha_k$, respectively.

Thus, by Lemma 4, we can derive

$$\bigvee_{\alpha\in\mathcal{A},\beta\in\mathcal{B}}(z_1+z_2=\alpha+\beta) \tag{9}$$

from D_1 and D_2 in a polynomial-size (in $|D_1|$ and $|D_2|$) R(lin)-proof. This concludes the first part of the lemma.

Assume that $\vec{a}$ and $\vec{b}$ consist of constant coefficients only. Then by inspecting the R(lin)-proof of (9) from D_1 and D_2 demonstrated above (and by using Lemma 5 instead of Lemma 4), one can verify that this proof is in fact carried inside $\mathrm{R}^0(\mathrm{lin})$. ∎

An immediate corollary of Lemma 6 is the efficient formalization in R(lin) of the following obvious counting argument: If a linear form equals some value in the interval (of integer numbers) $[a_0,a_1]$ and another linear form equals some value in $[b_0,b_1]$ (for some $a_0\le a_1$ and $b_0\le b_1$), then their addition equals some value in $[a_0+b_0,a_1+b_1]$. More formally:

Corollary 7. *Let z_1 abbreviate $\vec{a}\cdot\vec{x}$ and z_2 abbreviate $\vec{b}\cdot\vec{x}$. Let D_1 be $(z_1=a_0)\vee(z_1=a_0+1)\ldots\vee(z_1=a_1)$, and let D_2 be $(z_2=b_0)\vee(z_2=b_0+1)\ldots\vee(z_2=b_1)$. Then there is a polynomial-size (in the size of D_1, D_2) R(lin) proof from D_1, D_2 of*

$$(z_1+z_2=a_0+b_0)\vee(z_1+z_2=a_0+b_0+1)\vee\ldots\vee(z_1+z_2=a_1+b_1)\ . \tag{10}$$

Moreover, if $\vec{a}$ and $\vec{b}$ consist of constant *integers (which means that D_1, D_2 are $\mathrm{R}^0(\mathrm{lin})$-lines), then there is a polynomial-size (in the size of D_1, D_2) $\mathrm{R}^0(\mathrm{lin})$ proofs of (10) from D_1, D_2.*

Lemma 8. *Let $\vec{a}\cdot\vec{x}$ be a linear form with n variables, and let $\mathcal{A}:=\{\vec{a}\cdot\vec{x}\mid\vec{x}\in\{0,1\}^n\}$ be the set of all possible values of $\vec{a}\cdot\vec{x}$ over Boolean assignments to $\vec{x}$. Then there is a polynomial-size, in the size of the linear form $\vec{a}\cdot\vec{x}$,[6] R(lin) proof of*

$$\bigvee_{\alpha\in\mathcal{A}}(\vec{a}\cdot\vec{x}=\alpha)\,. \tag{11}$$

Moreover, if the coefficients in $\vec{a}$ are constants, then there is a polynomial-size (in the size of $\vec{a}\cdot\vec{x}$) $\mathrm{R}^0(\mathrm{lin})$ proof of (11).

[6]Recall that the size of $\vec{a}\cdot\vec{x}$ is $\sum_{i=1}^n|a_i|$, that is, the size of the unary representation of $\vec{a}$.

Proof: Without loss of generality, assume that all the coefficients in $\vec{a}$ are nonzero. Consider the Boolean axiom $(x_1 = 0) \vee (x_1 = 1)$ and the (first) coefficient a_1 from $\vec{a}$. Assume that $a_1 \geq 1$. Add $(x_1 = 0)$ to itself a_1 times, and arrive at $(a_1 x_1 = 0) \vee (x_1 = 1)$. Then, in the resulted line, add $(x_1 = 1)$ to itself a_1 times, until the following is reached:

$$(a_1 x_1 = 0) \vee (a_1 x_1 = a_1)\,.$$

Similarly, in case $a_1 \leq -1$ we can subtract ($|a_1| + 1$ many times) $(x_1 = 0)$ from itself in $(x_1 = 0) \vee (x_1 = 1)$, and then subtract ($|a_1| + 1$ many times) $(x_1 = 1)$ from itself in the resulted line.

In the same manner, we can derive the disjunctions: $(a_2 x_2 = 0) \vee (a_2 x_2 = a_2), \ldots, (a_n x_n = 0) \vee (a_n x_n = a_n)$.

Consider $(a_1 x_1 = 0) \vee (a_1 x_1 = a_1)$ and $(a_2 x_2 = 0) \vee (a_2 x_2 = a_2)$. From these two lines, by Lemma 6, there is a polynomial-size in $|a_1| + |a_2|$ derivation of:

$$(a_1 x_1 + a_2 x_2 = 0) \vee (a_1 x_1 + a_2 x_2 = a_1) \vee (a_1 x_1 + a_2 x_2 = a_2) \vee (a_1 x_1 + a_2 x_2 = a_1 + a_2)\,. \quad (12)$$

In a similar fashion, now consider $(a_3 x_3 = 0) \vee (a_3 x_3 = a_3)$ and apply again Lemma 6, to obtain

$$\bigvee_{\alpha \in \mathcal{A}'} (a_1 x_1 + a_2 x_2 + a_3 x_3 = \alpha)\,, \quad (13)$$

where $\mathcal{A}'$ are all possible values to $a_1 x_1 + a_2 x_2 + a_3 x_3$ over Boolean assignments to x_1, x_2, x_3. The derivation of (13) is of size polynomial in $|a_1| + |a_2| + |a_3|$.

Continue to consider, successively, all other lines $(a_4 x_4 = 0) \vee (a_4 x_4 = a_4), \ldots, (a_n x_n = 0) \vee (a_n x_n = a_n)$, and apply the same reasoning. Each step uses a derivation of size at most polynomial in $\sum_{i=1}^{n} |a_i|$. And so overall we reach the desired line (11), with a derivation of size polynomial in the size of $\vec{a} \cdot \vec{x}$. This concludes the first part of the lemma.

Assume that $\vec{a}$ consists of constant coefficients only. Then by inspecting the R(lin)-proof demonstrated above (and by using the second part of Lemma 6), one can see that this proof is in fact carried inside $\mathrm{R}^0(\mathrm{lin})$. ∎

Lemma 9. *There is a polynomial-size (in n)* $\mathrm{R}^0(\mathrm{lin})$ *proof from*

$$(x_1 = 1) \vee \cdots \vee (x_n = 1) \quad (14)$$

of

$$(x_1 + \ldots + x_n = 1) \vee \cdots \vee (x_1 + \ldots + x_n = n)\,. \quad (15)$$

Proof: We show that for every $i \in [n]$, there is a polynomial-size (in n) $\mathrm{R}^0(\mathrm{lin})$ proof from $(x_i = 1)$ of $(x_1 + \ldots + x_n = 1) \vee \cdots \vee (x_1 + \ldots + x_n = n)$. This concludes the proof since, by Lemma 5, we then can derive from (14) (with a polynomial-size (in n) $\mathrm{R}^0(\mathrm{lin})$ proof) the disjunction (14) in which each $(x_i = 1)$ (for all $i \in [n]$) is replace by $(x_1 + \ldots + x_n = 1) \vee \cdots \vee (x_1 + \ldots + x_n = n)$, which is precisely the disjunction (15) (note that (15) is an $\mathrm{R}^0(\mathrm{lin})$-line).

Claim 2. For every $i \in [n]$, there is a a polynomial-size (in n) $\mathrm{R}^0(\mathrm{lin})$ proof from $(x_i = 1)$ of $(x_1 + \ldots + x_n = 1) \vee \cdots \vee (x_1 + \ldots + x_n = n)$.

Proof of claim: By Lemma 8, for every $i \in [n]$ there is a polynomial-size (in n) $\mathrm{R}^0(\mathrm{lin})$ proof (using only the Boolean axioms) of

$$(x_1 + \ldots + x_{i-1} + x_{i+1} + \ldots + x_n = 0) \vee \cdots \vee (x_1 + \ldots + x_{i-1} + x_{i+1} + \ldots + x_n = n - 1)\,. \quad (16)$$

Now add successively $(x_i = 1)$ to every equation in (16) (note that this can be done in $\mathrm{R}^0(\mathrm{lin})$). We obtain precisely $(x_1 + \ldots + x_n = 1) \vee \cdots \vee (x_1 + \ldots + x_n = n)$. ∎ ∎

Lemma 10. *There is a polynomial-size (in n)* $\mathrm{R}^0(\mathrm{lin})$ *proof of* $(x_1+\ldots+x_n=0)\vee(x_1+\ldots+x_n=1)$ *from the collection of disjunctions consisting of* $(x_i=0)\vee(x_j=0)$*, for all* $1\leq i<j\leq n$*.*

Proof: We proceed by induction on n. The base case for $n=1$ is immediate from the Boolean axiom $(x_1=0)\vee(x_1=1)$. Assume we already have a polynomial-size proof of

$$(x_1+\ldots+x_n=0)\vee(x_1+\ldots+x_n=1). \tag{17}$$

If $x_{n+1}=0$ we add $x_{n+1}=0$ to both of the equations in (17), and reach:

$$(x_1+\ldots+x_{n+1}=0)\vee(x_1+\ldots+x_{n+1}=1). \tag{18}$$

Otherwise, $x_{n+1}=1$, and so we can cut-off $(x_{n+1}=0)$ in all the initial disjunctions $(x_i=0)\vee(x_{n+1}=0)$, for all $1\leq i\leq n$. We thus obtain $(x_1=0),\ldots,(x_n=0)$. Adding together $(x_1=0),\ldots,(x_n=0)$ and $(x_{n+1}=1)$ we arrive at

$$(x_1+\ldots+x_{n+1}=1)\,. \tag{19}$$

So overall, either (18) holds or (19) holds; and so (using Lemma 5) we arrive at the disjunction of (19) and (18), which is precisely (18). ∎

5. Implicational Completeness of R(lin) and its Subsystems

In this section we provide a proof of the implicational completeness of R(lin) and its subsystems. We shall need this property in the sequel (see Section 6.2). The implicational completeness of a proof system is a stronger property than mere completeness. Essentially, a system is implicationally complete if whenever something is *semantically* implied by a set of initial premises, then it is also *derivable* from the initial premises. In contrast to this, mere completeness means that any tautology (or in case of a refutation system, any unsatisfiable set of initial premises) has a proof in the system (respectively, a refutation in the system). As a consequence, the proof of implicational completeness in this section establishes an alternative completeness proof to that obtained via simulating resolution (see Proposition 1). Note that we are not concerned in this section with the size of the proofs, but only with their existence.

Recall the definition of the semantic implication relation $\models$ from Section 3.1. Formally, we say that R(lin) is *implicationally complete* if for every collection of disjunctions of linear equations $D_0, D_1, \ldots, D_m$, it holds that $D_1, \ldots, D_m \models D_0$ implies that there is an R(lin) proof of D_0 from $D_1, \ldots, D_m$.

Theorem 11. *R(lin) is implicationally complete.*

Proof: We proceed by induction on n, the number of variables $x_1, \ldots, x_n$ in $D_0, D_1, \ldots, D_m$.

The base case $n=0$. We need to show that $D_1, \ldots, D_m \models D_0$ implies that there is an R(lin) proof of D_0 from $D_1, \ldots, D_m$, where all D_i's (for $0\leq i\leq m$) have no variables but only constants. This means that each D_i is a disjunction of equations of the form $(0=a_0)$ for some integer a_0 (if a linear equation have no variables, then the left hand side of this equation must be 0; see Section 3.1).

There are two cases to consider. In the first case D_0 *is satisfiable*. Since D_0 has no variables, this means precisely that D_0 is the equation $(0=0)$. Thus, D_0 can be derived easily from any axiom in R(lin) (for instance, by subtracting each equation in $(x_1=0)\vee(x_1=1)$ from itself, to reach $(0=0)\vee(0=0)$, which is equal to $(0=0)$, since we discard duplicate equations inside disjunctions).

In the second case D_0 *is unsatisfiable*. Thus, since $D_1, \ldots, D_m \models D_0$, there is no assignment satisfying all $D_1, \ldots, D_m$. Hence, there must be at least one unsatisfiable disjunction D_i in

$D_1, \ldots, D_m$ (as a disjunction with no variables is either tautological or unsatisfiable). Such an unsatisfiable D_i is a disjunction of zero or more unsatisfiable equations of the form $(0 = a_0)$, for some integer $a_0 \neq 0$. We can then use Simplification to cut-off all the unsatisfiable equations in D_i to reach the empty disjunction. By the Weakening rule, we can now derive D_0 from the empty disjunction.

The induction step. Assume that the theorem holds for disjunctions with n variables. Let the underlying variables of $D_0, D_1, \ldots, D_m$ be $x_1, \ldots, x_{n+1}$, and assume that

$$D_1, \ldots, D_m \models D_0 \,. \tag{20}$$

We write the disjunction D_0 as:

$$\bigvee_{j=1}^{t} \left(\sum_{i=1}^{n} a_i^{(j)} x_i + a_{n+1}^{(j)} x_{n+1} = a_0^{(j)} \right) , \tag{21}$$

where the $a_i^{(j)}$'s are integer coefficients. We need to show that there is an R(lin) proof of D_0 from $D_1, \ldots, D_m$.

Let D be a disjunction of linear equations, let x_i be a variable and let $b \in \{0, 1\}$. We shall denote by $D{\restriction}_{x_i=b}$ the disjunction D, where in every equation in D the variable x_i is substituted by b, and the constant terms in the left hand sides of all resulting equations (after substituting b for x_i) switch sides (and change signs, obviously) to the right hand sides of the equations (we have to switch sides of constant terms, as by definition linear equations in R(lin) proofs have all constant terms appearing only on the right hand sides of equations).

We now reason (slightly) informally inside R(lin) (as illustrated in Section 4.1). Fix some $b \in \{0, 1\}$, and assume that $x_{n+1} = b$. Then, from $D_1, \ldots, D_m$ we can derive (inside R(lin)):

$$D_1{\restriction}_{x_{n+1}=b}, \ldots, D_m{\restriction}_{x_{n+1}=b} \ . \tag{22}$$

The only variables occurring in (22) are $x_1, \ldots, x_n$. From assumption (20) we clearly have $D_1{\restriction}_{x_{n+1}=b}, \ldots, D_m{\restriction}_{x_{n+1}=b} \models D_0{\restriction}_{x_{n+1}=b}$. And so by the induction hypothesis there is an R(lin) derivation of $D_0{\restriction}_{x_{n+1}=b}$ from $D_1{\restriction}_{x_{n+1}=b}, \ldots, D_m{\restriction}_{x_{n+1}=b}$. So overall, assuming that $x_{n+1} = b$, there is an R(lin) derivation of $D_0{\restriction}_{x_{n+1}=b}$ from $D_1, \ldots, D_m$.

We now consider the two possible cases: $x_{n+1} = 0$ and $x_{n+1} = 1$.

In case $x_{n+1} = 0$, by the above discussion, we can derive $D_0{\restriction}_{x_{n+1}=0}$ from $D_1, \ldots, D_m$. For every $j \in [t]$, add successively ($a_{n+1}^{(j)}$ times) the equation $x_{n+1} = 0$ to the jth equation in $D_0{\restriction}_{x_{n+1}=0}$ (see (21)). We thus obtain precisely D_0.

In case $x_{n+1} = 1$, again, by the above discussion, we can derive $D_0{\restriction}_{x_{n+1}=1}$ from $D_1, \ldots, D_m$. For every $j \in [t]$, add successively ($a_{n+1}^{(j)}$ times) the equation $x_{n+1} = 1$ to the jth equation in $D_0{\restriction}_{x_{n+1}=1}$ (recall that we switch sides of constant terms in every linear equation after the substitution of x_{n+1} by 1 is performed in $D_0{\restriction}_{x_{n+1}=1}$). Again, we obtain precisely D_0. ∎

By inspecting the proof of Theorem 11, it is possible to verify that if all the disjunctions $D_0, , \ldots, D_m$ are R^0(lin)-lines (see Definition 3.2), then the proof of D_0 in R(lin) uses only R^0(lin)-lines as well. Therefore, we have:

Corollary 12. R^0(lin) *is implicationally complete.*

Remark 1. Corollary 12 states that any R^0(lin)-line that is semantically implied by a set of initial R^0(lin)-lines, is in fact derivable in R^0(lin) from the initial R^0(lin)-lines. On the other hand, it is possible that a certain proof of the same R^0(lin)-line inside R(lin) will be significantly shorter than the proof inside R^0(lin). Indeed, we shall see in Section 8 that for certain CNF formulas R(lin) has a super-polynomial speed-up over R^0(lin).

6. Short Proofs for Hard Tautologies

In this section we show that $\mathrm{R}^0(\mathrm{lin})$ is already enough to admit small proofs for "hard" counting principles like the pigeonhole principle and the Tseitin graph formulas for constant degree graphs. On the other hand, as we shall see in Section 8, $\mathrm{R}^0(\mathrm{lin})$ inherits the same weakness that cutting planes proofs have with respect to the clique-coloring tautologies. Nevertheless, we can efficiently prove the clique-coloring principle in (the stronger system) $\mathrm{R}(\mathrm{lin})$, but not by using $\mathrm{R}(\mathrm{lin})$ "ability to count", rather by using its (straightforward) ability to simulate $\mathrm{Res}(2)$ proofs (that is, resolution proofs extended to operate with 2-DNF formulas, instead of clauses).

6.1. The Pigeonhole Principle Tautologies in $\mathrm{R}^0(\mathrm{lin})$.

This subsection illustrates polynomial-size $\mathrm{R}^0(\mathrm{lin})$ proofs of the pigeonhole principle. This will allow us to establish polynomial-size multilinear proofs operating with depth-3 multilinear formulas of the pigeonhole principle (in Section 9).

The *m to n pigeonhole principle* states that m pigeons cannot be mapped one-to-one into $n < m$ holes. The negation of the pigeonhole principle, denoted $\neg\mathrm{PHP}^m_n$, is formulated as an unsatisfiable CNF formula as follows (where clauses are translated to disjunctions of linear equations):

Definition 6.1. The $\neg\mathrm{PHP}^m_n$ is the following set of clauses:

(1) Pigeons axioms: $(x_{i,1} = 1) \vee \cdots \vee (x_{i,n} = 1)$, for all $1 \leq i \leq m$;
(2) Holes axioms: $(x_{i,k} = 0) \vee (x_{j,k} = 0)$, for all $1 \leq i < j \leq m$ and for all $1 \leq k \leq n$.

The intended meaning of each propositional variable $x_{i,j}$ is that the ith pigeon is mapped to the jth hole.

We now describe a polynomial-size in n refutation of $\neg\mathrm{PHP}^m_n$ inside $\mathrm{R}^0(\mathrm{lin})$. For this purpose it is sufficient to prove a polynomial-size refutation of the pigeonhole principle when the number of pigeons m equals $n+1$ (because the set of clauses pertaining to $\neg\mathrm{PHP}^{n+1}_n$ is already contained in the set of clauses pertaining to $\neg\mathrm{PHP}^m_n$, for any $m > n$). Thus, we fix $m = n + 1$. In this subsection we shall say a proof in $\mathrm{R}^0(\mathrm{lin})$ is of *polynomial-size*, always intending *polynomial-size in n* (unless otherwise stated).

By Lemma 9, for all $i \in [m]$ we can derive from the Pigeon axiom (for the ith pigeon):

$$(x_{i,1} + \ldots + x_{i,n} = 1) \vee \cdots \vee (x_{i,1} + \ldots + x_{i,n} = n) \tag{23}$$

with a polynomial-size $\mathrm{R}^0(\mathrm{lin})$ proof.

By Lemma 10, from the Hole axioms we can derive, with a polynomial-size $\mathrm{R}^0(\mathrm{lin})$ proof

$$(x_{1,j} + \ldots + x_{m,j} = 0) \vee (x_{1,j} + \ldots + x_{m,j} = 1), \tag{24}$$

for all $j \in [n]$.

Let S abbreviate the sum of all formal variables $x_{i,j}$. In other words,

$$S := \sum_{i \in [m], j \in [n]} x_{i,j}\,.$$

Lemma 13. *There is a polynomial-size* $\mathrm{R}^0(\mathrm{lin})$ *proof from (23) (for all $i \in [m]$) of*

$$(S = m) \vee (S = m + 1) \cdots \vee (S = m \cdot n).$$

Proof: For every $i \in [m]$ fix the abbreviation $z_i := x_{i,1} + \ldots + x_{i,n}$. Thus, by (23) we have $(z_i = 1) \vee \cdots \vee (z_i = n)$.

Consider $(z_1 = 1) \vee \cdots \vee (z_1 = n)$ and $(z_2 = 1) \vee \cdots \vee (z_2 = n)$. By Corollary 7, we can derive from these two lines

$$(z_1 + z_2 = 2) \vee (z_1 + z_2 = 3) \vee \cdots \vee (z_1 + z_2 = 2n) \tag{25}$$

with a polynomial-size $\mathrm{R}^0(\mathrm{lin})$ proof.

Now, consider $(z_3 = 1) \vee \cdots \vee (z_3 = n)$ and (25). By Corollary 7 again, from these two lines we can derive with a polynomial-size $\mathrm{R}^0(\mathrm{lin})$ proof:

$$(z_1 + z_2 + z_3 = 3) \vee (z_1 + z_2 + z_3 = 4) \vee \cdots \vee (z_1 + z_2 + z_3 = 3n)\,. \tag{26}$$

Continuing in the same way, we eventually arrive at

$$(z_1 + \ldots + z_m = m) \vee (z_1 + \ldots + z_m = m + 1) \vee \cdots \vee (z_1 + \ldots + z_m = m \cdot n)\,,$$

which concludes the proof, since S equals $z_1 + \ldots + z_m$. ∎

Lemma 14. *There is a polynomial-size* $\mathrm{R}^0(\mathrm{lin})$ *proof from (24) of*

$$(S = 0) \vee \cdots \vee (S = n).$$

Proof: For all $j \in [n]$, fix the abbreviation $y_j := x_{1,j} + \ldots + x_{m,j}$. Thus, by (24) we have $(y_j = 0) \vee (y_j = 1)$, for all $j \in [n]$. Now the proof is similar to the proof of Lemma 8, except that here single variables are abbreviations of linear forms.

If $y_1 = 0$ then we can add y_1 to the two sums in $(y_2 = 0) \vee (y_2 = 1)$, and reach $(y_1 + y_2 = 0) \vee (y_1 + y_2 = 1)$ and if $y_1 = 1$ we can do the same and reach $(y_1 + y_2 = 1) \vee (y_1 + y_2 = 2)$. So, by Lemma 5, we can derive with a polynomial-size $\mathrm{R}^0(\mathrm{lin})$ proof

$$(y_1 + y_2 = 0) \vee (y_1 + y_2 = 1) \vee (y_1 + y_2 = 2)\,. \tag{27}$$

Now, we consider the three cases in (27): $y_1 + y_2 = 0$ or $y_1 + y_2 = 1$ or $y_1 + y_2 = 2$, and the clause $(y_3 = 0) \vee (y_3 = 1)$. We arrive in a similar manner at $(y_1 + y_2 + y_3 = 0) \vee \cdots \vee (y_1 + y_2 + y_3 = 3)$. We continue in the same way until we arrive at $(S = 0) \vee \cdots \vee (S = n)$. ∎

Theorem 15. *There is a polynomial-size* $\mathrm{R}^0(\mathrm{lin})$ *refutation of the* m *to* n *pigeonhole principle* $\neg PHP_n^m$.

Proof: By Lemmas 13 and 14 above, all we need is to show a polynomial-size refutation of $(S = m) \vee \cdots \vee (S = m \cdot n)$ and $(S = 0) \vee \cdots \vee (S = n)$.

Since $n < m$, for all $0 \leq k \leq n$, if $S = k$ then using the Resolution and Simplification rules we can cut-off all the sums in $(S = m) \vee \cdots \vee (S = m \cdot n)$ and arrive at the empty clause. Thus, by Lemma 5, there is a polynomial-size $\mathrm{R}^0(\mathrm{lin})$ proof of the empty clause from $(S = 0) \vee \cdots \vee (S = n)$ and $(S = m) \vee \cdots \vee (S = m \cdot n)$. ∎

6.2. **Tseitin mod p Tautologies in $\mathrm{R}^0(\mathrm{lin})$.** This subsection establishes polynomial-size $\mathrm{R}^0(\mathrm{lin})$ proofs of Tseitin graph tautologies (for constant degree graphs). This will allow us (in Section 9) to extend the multilinear proofs of the Tseitin mod p tautologies to any field of characteristic 0 (the proofs in [RT06] required working over a field containing a primitive pth root of unity when proving the Tseitin mod p tautologies; for more details see Section 9).

Tseitin mod p tautologies (introduced in [BGIP01]) are generalizations of the (original, mod 2) Tseitin graph tautologies (introduced in [Tse68]). To build the intuition for the generalized version, we start by describing the (original) Tseitin mod 2 principle. 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.

As mentioned above, the Tseitin mod 2 principle was generalized by Buss *et al.* [BGIP01] to obtain the Tseitin mod p principle. 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 $n \equiv 1 \pmod{p}$). We shall see in what follows, that this simple counting argument can be carried on in a natural (and efficient) way already inside $\mathrm{R}^0(\mathrm{lin})$.

As an unsatisfiable propositional formula (in CNF form) the negation of 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 the edge e has an associated value i. The following are the clauses of the Tseitin mod p CNF formula (as translated to disjunctions of linear equations).

Definition 6.2 (**Tseitin mod p formulas** ($\neg\textsc{Tseitin}_{G,p}$))**.** 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 $v \in V$, denote the edges in E' coming out of v by $e[v, 1], \ldots, e[v, r]$ and define the following set of (translation of) clauses:

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

The Tseitin mod p formula, denoted $\neg\textsc{Tseitin}_{G,p}$, consists of the following (translation) of clauses:

1. $\bigvee_{i=0}^{p-1} (x_{e,i} = 1)$, for all $e \in E'$
(expresses that every edge is assigned at least one value from $0, \ldots, p-1$);
2. $(x_{e,i} = 0) \vee (x_{e,j} = 0)$, 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. $(x_{e,i} = 1) \vee (x_{\bar{e},p-i} = 0)$ and $(x_{e,i} = 0) \vee (x_{\bar{e},p-i} = 1)$,[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).

[7] If $i = 0$ then $x_{\bar{e},p-i}$ denotes $x_{\bar{e},0}$.

Note that for every edge $e \in E'$, the polynomials of (1,2) in Definition 6.2, combined with the Boolean axioms of $\mathrm{R}^0(\mathrm{lin})$, force any collection of edge-variables $x_{e,0}, \ldots, x_{e,p-1}$ to contain exactly one $i \in \{0, \ldots, p-1\}$ so that $x_{e,i} = 1$. 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 disjunctions of (1,2) and the disjunctions 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).

Until the rest of this subsection we fix an integer $p \geq 2$ and a connected undirected r-regular graph $G = (V, E)$ with n vertices and no double edges, such that $n \equiv 1 \mod p$ and r is a constant. As in Definition 6.2, we 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. We now proceed to refute $\neg\textsc{Tseitin}_{G,p}$ inside $\mathrm{R}^0(\mathrm{lin})$ with a polynomial-size (in n) refutation.

Given a vertex $v \in V$, and the edges in E' coming out of v, denoted $e[v,1], \ldots, e[v,r]$, define the following abbreviation:

$$\alpha_v := \sum_{j=1}^{r} \sum_{i=0}^{p-1} i \cdot x_{e[v,j],i} \,. \tag{28}$$

Lemma 16. *Let $v \in V$ be any vertex in G'. Then there is a constant-size $\mathrm{R}^0(\mathrm{lin})$ proof from $\neg\textsc{Tseitin}_{G,p}$ of the following disjunction:*

$$\bigvee_{\ell=0}^{r-1} (\alpha_v = 1 + \ell \cdot p) \,. \tag{29}$$

Proof: Let $T_v \subseteq \neg\textsc{Tseitin}_{G,p}$ be the set of all disjunctions of the form (1,2,4) from Definition 6.2 that contain only variables pertaining to vertex v (that is, all the variables $x_{e,i}$, where $e \in E'$ is an edge coming out of v, and $i \in \{0, \ldots, p-1\}$).

Claim 3. T_v semantically implies (29), that is:[8]

$$T_v \models \bigvee_{\ell=0}^{r-1} (\alpha_v = 1 + \ell \cdot p) \,.$$

Proof of claim: Let σ be an assignment of $0, 1$ values to the variables in T_v that satisfies both the disjunctions of (1,2) and the disjunctions of $\mathrm{MOD}_{p,1}(v)$ in Definition 6.2. As mentioned above (the comment after Definition 6.2), such a 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. This means precisely that $\alpha_v = 1 \mod p$ under the assignment σ. Thus, there exists a nonnegative integer k, such that $\alpha_v = 1 + kp$ under σ.

It remains to show that $k \leq r - 1$ (and so the only possible values that α_v can get under σ are $1, 1+p, 1+2p, \ldots, 1+(r-1)p$). Note that because σ gives the value 1 to only one variable from $x_{e[v,j],0}, \ldots, x_{e[v,j],p-1}$ (for every $j \in [r]$), then the maximal value that α_v can have under σ is $r(p-1)$. Thus, $1 + kp \leq rp - r$ and so $k \leq r - 1$. ∎

From Claim 3 and from the implicational completeness of $\mathrm{R}^0(\mathrm{lin})$ (Corollary 12), there exists an $\mathrm{R}^0(\mathrm{lin})$ derivation of (29) from T_v. It remains to show that this derivation is of constant-size.

Since the degree r of G' and the modulus p are both constants, both T_v and (29) have constant number of variables and constant coefficients (including the free-terms). Thus, there is a constant-size $\mathrm{R}^0(\mathrm{lin})$ derivation of (29) from T_v. ∎

[8] Recall that we only consider assignments of $0, 1$ values to variables when considering the semantic implication relation $\models$.

Lemma 17. *There is a polynomial-size (in n)* $\mathrm{R}^0(\mathrm{lin})$ *derivation from* $\neg\textsc{Tseitin}_{G,p}$ *of the following disjunction:*

$$\bigvee_{\ell=0}^{(r-1)\cdot n}\left(\sum_{v\in V}\alpha_v = n+\ell\cdot p\right).$$

Proof: Simply add successively all the equations pertaining to disjunctions (29), for all vertices $v\in V$. Formally, we show that for every subset of vertices $\mathcal{V}\subseteq V$, with $|\mathcal{V}|=k$, there is a polynomial-size (in n) $\mathrm{R}^0(\mathrm{lin})$ derivation from $\neg\textsc{Tseitin}_{G,p}$ of

$$\bigvee_{\ell=0}^{(r-1)\cdot k}\left(\sum_{v\in \mathcal{V}}\alpha_v = k+\ell\cdot p\right), \tag{30}$$

and so putting $\mathcal{V}=V$, will conclude the proof.

We proceed by induction on the size of $\mathcal{V}$. The base case, $|\mathcal{V}|=1$, is immediate from Lemma 16.

Assume that we already derived (30) with a polynomial-size (in n) $\mathrm{R}^0(\mathrm{lin})$ proof, for some $\mathcal{V}\subset V$, such that $|\mathcal{V}|=k<n$. Let $u\in V\setminus\mathcal{V}$. By Lemma 16, we can derive

$$\bigvee_{\ell=0}^{r-1}(\alpha_u = 1+\ell\cdot p) \tag{31}$$

from $\neg\textsc{Tseitin}_{G,p}$ with a constant-size proof. Now, by Lemma 6, each linear equation in (31) can be added to each linear equation in (30), with a polynomial-size (in n) $\mathrm{R}^0(\mathrm{lin})$ proof. This results in the following disjunction:

$$\bigvee_{\ell=0}^{(r-1)\cdot(k+1)}\left(\sum_{v\in \mathcal{V}\cup\{u\}}\alpha_v = k+1+\ell\cdot p\right),$$

which is precisely what we need to conclude the induction step. ∎

Lemma 18. *Let $e,\bar{e}$ be any pair of opposite directed edges in G' and let $i\in\{0,\ldots,p-1\}$. Let $T_e\subseteq\neg\textsc{Tseitin}_{G,p}$ be the set of all disjunctions of the form (1,2,3) from Definition 6.2 that contain only variables pertaining to edges $e,\bar{e}$ (that is, all the variables $x_{e,j},x_{\bar{e},j}$, for all $j\in\{0,\ldots,p-1\}$). Then, there is a constant-size* $\mathrm{R}^0(\mathrm{lin})$ *proof from T_e of the following disjunction:*

$$(i\cdot x_{e,i}+(p-i)\cdot x_{\bar{e},p-i}=0)\vee(i\cdot x_{e,i}+(p-i)\cdot x_{\bar{e},p-i}=p)\,. \tag{32}$$

Proof: First note that T_e semantically implies

$$(x_{e,i}+x_{\bar{e},p-i}=0)\vee(x_{e,i}+x_{\bar{e},p-i}=2)\,. \tag{33}$$

The number of variables in T_e and (33) is constant. Hence, there is a constant-size $\mathrm{R}^0(\mathrm{lin})$-proof of (32) from T_e. Also note that

$$\begin{aligned}(x_{e,i}+x_{\bar{e},p-i}=0)\vee(x_{e,i}+x_{\bar{e},p-i}=2)\models\\ (i\cdot x_{e,i}+(p-i)\cdot x_{\bar{e},p-i}=0)\vee(i\cdot x_{e,i}+(p-i)\cdot x_{\bar{e},p-i}=p)\,.\end{aligned} \tag{34}$$

Therefore, there is also an $\mathrm{R}^0(\mathrm{lin})$-proof of constant-size from T_e of the lower line in (34). ∎

We are now ready to complete the polynomial-size $\mathrm{R}^0(\mathrm{lin})$ refutation of $\neg\textsc{Tseitin}_{G,p}$. Using the two prior lemmas, the refutation idea is simple, as we now explain. Observe that

$$\sum_{v\in V}\alpha_v = \sum_{\substack{\{e,\bar{e}\}\subseteq E'\\ i\in\{0,\ldots,p-1\}}}(i\cdot x_{e,i}+(p-i)\cdot x_{\bar{e},p-i})\,, \tag{35}$$

where by $\{e, \bar{e}\} \subseteq E'$ we mean that $e, \bar{e}$ is pair of opposite directed edges in G'.

Derive by Lemma 17 the disjunction

$$\bigvee_{\ell=0}^{(r-1)\cdot n} \left(\sum_{v \in V} \alpha_v = n + \ell \cdot p \right) . \tag{36}$$

This disjunction expresses the fact that $\sum_{v \in V} \alpha_v = 1 \mod p$ (since $n = 1 \mod p$). On the other hand, using Lemma 18, we can "sum together" all the equations (32) (for all $\{e, \bar{e}\} \subseteq E'$ and all $i \in \{0, \ldots, p-1\}$), to obtain a disjunction expressing the statement that

$$\sum_{\substack{\{e,\bar{e}\} \subseteq E' \\ i \in \{0,\ldots,p-1\}}} (i \cdot x_{e,i} + (p-i) \cdot x_{\bar{e},p-i}) = 0 \mod p .$$

By Equation (35), we then obtain the desired contradiction. This idea is formalized in the proof of the following theorem:

Theorem 19. *Let $G = (V, E)$ be an r-regular graph with n vertices, where r is a constant. Fix some modulus p. Then, there are polynomial-size (in n) $\mathrm{R}^0(\mathrm{lin})$ refutations of $\neg\textsc{Tseitin}_{G,p}$.*

Proof: First, use Lemma 17 to derive

$$\bigvee_{\ell=0}^{(r-1)\cdot n} \left(\sum_{v \in V} \alpha_v = n + \ell \cdot p \right) . \tag{37}$$

Second, use Lemma 18 to derive

$$(i \cdot x_{e,i} + (p-i) \cdot x_{\bar{e},p-i} = p) \vee (i \cdot x_{e,i} + (p-i) \cdot x_{\bar{e},p-i} = 0) , \tag{38}$$

for every pair of opposite directed edges in $G' = (V, E')$ (as in Definition 6.2) and every residue $i \in \{0, \ldots, p-1\}$.

We now reason inside $\mathrm{R}^0(\mathrm{lin})$. Pick a pair of opposite directed edges $e, \bar{e}$ and a residue $i \in \{0, \ldots, p-1\}$. If $i \cdot x_{e,i} + (p-i) \cdot x_{\bar{e},p-i} = 0$, then subtract this equation successively from every equation in (37). We thus obtain a new disjunction, similar to that of (37), but which does not contain the $x_{e,i}$ and $x_{\bar{e},p-i}$ variables, and with the same free-terms.

Otherwise, $i \cdot x_{e,i} + (p-i) \cdot x_{\bar{e},p-i} = p$, then subtract this equation successively from every equation in (37). Again, we obtain a new disjunction, similar to that of (37), but which does not contain the $x_{e,i}$ and $x_{\bar{e},p-i}$ variables, and such that p is subtracted from every free-term in every equation. Since, by assumption, $n \equiv 1 \mod p$, the free-terms in every equation are (still) equal $1 \mod p$.

So overall, in both cases ($i \cdot x_{e,i} + (p-i) \cdot x_{\bar{e},p-i} = 0$ and $i \cdot x_{e,i} + (p-i) \cdot x_{\bar{e},p-i} = p$) we obtained a new disjunction with all the free-terms in equations equal $1 \mod p$.

We now continue the same process for every pair $e, \bar{e}$ of opposite directed edges in G' and every residue i. Eventually, we discard all the variables $x_{e,i}$ in the equations, for every $e \in E'$ and $i \in \{0, \ldots, p-1\}$, while all the free-terms in every equation remain to be equal $1 \mod p$. Therefore, we arrive at a disjunction of equations of the form $(0 = \gamma)$ for some $\gamma = 1 \mod p$. By using the Simplification rule we can cut-off all such equations, and arrive finally at the empty disjunction. ∎

6.3. **The Clique-Coloring Principle in R(lin).** In this section we observe that there are polynomial-size R(lin) proofs of the clique-coloring principle (for certain, weak, parameters). This implies, in particular, that R(lin) does not possess the feasible monotone interpolation property (see more details on the interpolation method in Section 7).

Atserias, Bonet & Esteban [ABE02] demonstrated polynomial-size Res(2) refutations of the clique-coloring formulas (for certain weak parameters; Theorem 20). Thus, it is sufficient to show that R(lin) polynomially-simulates Res(2) proofs (Proposition 2). This can be shown in a straightforward manner. As noted in the first paragraph of Section 6, because the proofs of the clique-coloring formula we discuss here only follow the proofs inside Res(2), then in fact these proofs do not take any advantage of the capacity "to count" inside R(lin) (this capacity is exemplified, for instance, in Section 4.2).

We start with the clique-coloring formulas (these formulas will also be used in Section 8). These formulas express the clique-coloring principle that has been widely used in the proof complexity literature (cf., [BPR97], [Pud97], [Kra97], [Kra98], [ABE02], [Kra07]). This principle is based on the following basic combinatorial idea. Let $G = (V, E)$ be an undirected graph with n vertices and let $k' < k$ be two integers. Then, one of the following must hold:

(i): The graph G does not contain a *clique with k vertices*;

(ii): The graph G is not a *complete k'-partite graph*. In other words, there is no way to partition G into k' subgraphs $G_1, \ldots, G_{k'}$, such that every G_i is an independent set, and for all $i \neq j \in [k']$, all the vertices in G_i are connected by edges (in E) to all the vertices in G_j.

Obviously, if Item (ii) above is false (that is, if G is a complete k'-partite graph), then there exists a k'-coloring of the vertices of G; hence the name *clique-coloring* for the principle.

The propositional formulation of the (negation of the) clique-coloring principle is as follows. Each variable $p_{i,j}$, for all $i \neq j \in [n]$, is an indicator variable for the fact that there is an edge in G between vertex i and vertex j. Each variable $q_{\ell,i}$, for all $\ell \in [k]$ and all $i \in [n]$, is an indicator variable for the fact that the vertex i in G is the ℓth vertex in the k-clique. Each variable $r_{\ell,i}$, for all $\ell \in [k']$ and all $i \in [n]$, is an indicator variable for the fact that the vertex i in G pertains to the independent set G_ℓ.

Definition 6.3. The negation of the clique-coloring principle consists of the following unsatisfiable collection of clauses (as translated to disjunctions of linear equations), denoted $\neg\text{CLIQUE}^n_{k,k'}$:

(i) $(q_{\ell,1} = 1) \vee \cdots \vee (q_{\ell,n} = 1)$, for all $\ell \in [k]$
(expresses that there exists at least one vertex in G which constitutes the ℓth vertex of the k-clique);

(ii) $(q_{\ell,i} = 0) \vee (q_{\ell,j} = 0)$, for all $i \neq j \in [n]$, $\ell \in [k]$
(expresses that there exists at most one vertex in G which constitutes the ℓth vertex of the k-clique);

(iii) $(q_{\ell,i} = 0) \vee (q_{\ell',i} = 0)$, for all $i \in [n]$, $\ell \neq \ell' \in [k]$
(expresses that the ith vertex of G cannot be both the ℓth and the ℓ'th vertex of the k-clique);

(iv) $(q_{\ell,i} = 0) \vee (q_{\ell',j} = 0) \vee (p_{i,j} = 1)$, for all $\ell \neq \ell' \in [k], i \neq j \in [n]$
(expresses that if both the vertices i and j in G are in the k-clique, then there is an edge in G between i and j);

(v) $(r_{1,i} = 1) \vee \cdots \vee (r_{k',i} = 1)$, for all $i \in [n]$
(expresses that every vertex of G pertains to at least one independent set);

(vi) $(r_{\ell,i} = 0) \vee (r_{\ell',i} = 0)$, for all $\ell \neq \ell \in [k'], i \in [n]$
(expresses that every vertex of G pertains to at most one independent set);

(vii) $(p_{i,j} = 0) \vee (r_{t,i} = 0) \vee (r_{t,j} = 0)$, for all $i \neq j \in [n], t \in [k']$
(expresses that if there is an edge between vertex i and j in G, then i and j cannot be in the same independent set);

Remark 2. Our formulation of the clique-coloring formulas above is similar to the one used by [BPR97], except that we consider also the $p_{i,j}$ variables (we added the (iv) clauses and changed accordingly the (vii) clauses). This is done for the sake of clarity of the contradiction itself, and also to make it clear that the formulas are in the appropriate form required by the interpolation method (see Section 7 for details on the interpolation method). By resolving over the $p_{i,j}$ variables in (iv) and (vii), one can obtain precisely the collection of clauses in [BPR97].

Atserias, Bonet & Esteban [ABE02] demonstrated polynomial-size (in n) Res(2) refutations of $\neg\text{CLIQUE}^n_{k,k'}$, when $k = \sqrt{n}$ and $k' = (\log n)^2/8 \log\log n$. These are rather weak parameters, but they suffice to establish the fact that Res(2) does not possess the feasible monotone interpolation property.

The Res(2) proof system (also called 2-*DNF resolution*), first considered in [Kra01], is resolution extended to operate with 2-DNF formulas, defined as follows.

A 2-*term* is a conjunction of up to two literals. A 2-DNF is a disjunction of 2-terms. The size of a 2-term is the number of literals in it (that is, either 1 or 2). The *size of a* 2-*DNF* is the total size of all the 2-terms in it.

Definition 6.4 (Res(2))**.** A *Res(2) proof of a* 2-*DNF* D *from a collection* K *of* 2-*DNFs* is a sequence of 2-DNFs $D_1, D_2, \ldots, D_s$, such that $D_s = D$, and every D_j is either from K or was derived from previous line(s) in the sequence by the following inference rules:

Cut: Let A, B be two 2-DNFs.
From $A \vee \bigwedge_{i=1}^{2} l_i$ and $B \vee \bigvee_{i=1}^{2} \neg l_i$ derive $A \vee B$, where the l_i's are (not necessarily distinct) literals (and $\neg l_i$ is the negation of the literal l_i).

AND-introduction: Let A, B be two 2-DNFs and l_1, l_2 two literals.
From $A \vee l_1$ and $B \vee l_2$ derive $A \vee B \vee \bigwedge_{i=1}^{2} l_i$.

Weakening: From a 2-DNF A derive $A \vee \bigwedge_{i=1}^{2} l_i$, where the l_i's are (not necessarily distinct) literals.

A Res(2) *refutation* of a collection of 2-DNFs K is a Res(2) proof of the empty disjunction $\square$ from K (the empty disjunction stands for FALSE). The *size* of a Res(2) proof is the total size of all the 2-DNFs in it.

Given a collection K of 2-DNFs we translate it into a collection of disjunctions of linear equations via the following translation scheme. For a literal l, denote by $\widehat{l}$ the translation that maps a variable x_i into x_i, and $\neg x_i$ into $1 - x_i$. A 2-term $l_1 \wedge l_2$ is first transformed into the equation $\widehat{l_1} + \widehat{l_2} = 2$, and then moving the free-terms in the left hand side of $\widehat{l_1} + \widehat{l_2} = 2$ (in case there are such free-terms) to the right hand side; So that the final translation of $l_1 \wedge l_2$ has only a single free-term in the right hand side. A disjunction of 2-terms (that is, a 2-DNF) $D = \bigvee_{i \in I}(l_{i,1} \wedge l_{i,2})$ is translated into the disjunction of the translations of the 2-terms, denoted by $\widehat{D}$. It is clear that every assignment satisfies a 2-DNF D if and only if it satisfies $\widehat{D}$.

Proposition 2. *R(lin) polynomially simulates Res(2). In other words, if* π *is a Res(2) proof of* D *from a collection of* 2-*DNFs* $K_1, \ldots, K_t$*, then there is an R(lin) proof of* $\widehat{D}$ *from* $\widehat{K}_1, \ldots, \widehat{K}_t$ *whose size is polynomial in the size of* π*.*

The proof of Proposition 2 proceeds by induction on the length (that is, the number of proof-lines) in the Res(2) proof. This is pretty straightforward and similar to the simulation of resolution by R(lin), as illustrated in the proof of Proposition 1. We omit the details.

Theorem 20 ([ABE02])**.** *Let* $k = \sqrt{n}$ *and* $k' = (\log n)^2/8 \log\log n$*. Then* $\neg\text{CLIQUE}^n_{k,k'}$ *has Res(2) refutations of size polynomial in* n*.*

Thus, Proposition 2 yields the following:

Corollary 21. *Let k, k' be as in Theorem 20. Then $\neg\text{CLIQUE}^n_{k,k'}$ has R(lin) refutations of size polynomial in n.*

The following corollary is important (we refer the reader to Section A in the Appendix for the necessary relevant definitions concerning the *feasible monotone interpolation property* and to Section 7 for explanation and definitions concerning the general [non-monotone] interpolation method).

Corollary 22. R(lin) *does not possess the feasible monotone interpolation property.*

Remark 3. The proof of $\neg\text{CLIQUE}^n_{k,k'}$ inside Res(2) demonstrated in [ABE02] (and hence, also the corresponding proof inside R(lin)) proceeds along the following lines. First reduce $\neg\text{CLIQUE}^n_{k,k'}$ to the k to k' pigeonhole principle. For the appropriate values of the parameters k and k' — and specifically, for the values in Theorem 20 — there is a short *resolution* proof of the k to k' pigeonhole principle (this was shown by Buss & Pitassi [BP97]); (this resolution proof is polynomial in the number of pigeons k, but not in the number of holes k', which is exponentially smaller than k).[9] Therefore, in order to conclude the refutation of $\neg\text{CLIQUE}^n_{k,k'}$ inside Res(2) (or inside R(lin)), it suffices to simulate the short resolution refutation of the k to k' pigeonhole principle. It is important to emphasize this point: After reducing, inside R(lin), $\neg\text{CLIQUE}^n_{k,k'}$ to the pigeonhole principle, one simulates the *resolution* refutation of the pigeonhole principle, and this has nothing to do with the small-size R^0(lin) refutations of the pigeonhole principle demonstrated in Section 6.1. This is because, the reduction (inside R(lin)) of $\neg\text{CLIQUE}^n_{k,k'}$ to the k to k' pigeonhole principle, results in a *substitution instance* of the pigeonhole principle formulas; in other words, the reduction results in a collection of disjunctions that are similar to the pigeonhole principle disjunctions *where each original pigeonhole principle variable is substituted by some big formula* (and, in particular, these disjunctions are not R^0(lin)-lines at all). (Note that R^0(lin) does not admit short proofs of the clique-coloring formulas as we show in Section 8.)

7. Interpolation Results for R^0(lin)

In this section we study the applicability of the feasible (non-monotone) interpolation technique to R^0(lin) refutations. In particular, we show that R^0(lin) admits a polynomial (in terms of the R^0(lin)-proofs) upper bound on the (non-monotone) circuit-size of interpolants. In the next section we shall give a polynomial upper bound on the *monotone* circuit-size of interpolants, but only in the case that the interpolant corresponds to the clique-coloring formulas (whereas, in this section we are interested in the general case; that is, upper bounding circuit-size of interpolants corresponding to any formula [of the prescribed type; see below]). First, we shortly describe the feasible interpolation method and explain how this method can be applied to obtain (sometime, conditional) lower bounds on proof size. Explicit usage of the interpolation method in proof complexity goes back to [Kra94].

Let $A_i(\vec{p},\vec{q})$, $i \in I$, and $B_j(\vec{p},\vec{r})$, $j \in J$, (I and J are sets of indices) be a collection of formulas (for instance, a collection of disjunctions of linear equations) in the displayed variables only. Denote by $A(\vec{p},\vec{q})$ the conjunction of all $A_i(\vec{p},\vec{q})$, $i \in I$, and by $B(\vec{p},\vec{r})$, the conjunction of all $B_j(\vec{p},\vec{r})$, $j \in J$. Assume that $\vec{p},\vec{q},\vec{r}$ are pairwise disjoint sets of distinct variables, and that there is no assignment that satisfies both $A(\vec{p},\vec{q})$ and $B(\vec{p},\vec{r})$. Fix an assignment $\vec{\alpha}$ to the variables in $\vec{p}$. The $\vec{p}$ variables are the *only common variables* of the A_i's and the B_j's. Therefore, either $A(\vec{\alpha},\vec{q})$ is unsatisfiable or $B(\vec{\alpha},\vec{r})$ is unsatisfiable.

[9] Whenever $k \geq 2k'$ the k to k' pigeonhole principle is referred to as the *weak pigeonhole principle*.

The interpolation technique transforms a refutation of $A(\vec{p},\vec{q}) \wedge B(\vec{p},\vec{r})$, in some proof system, into a circuit (usually a Boolean circuit) that outputs 1 for those assignments $\vec{\alpha}$ (for $\vec{p}$) for which $A(\vec{\alpha},\vec{q})$ is unsatisfiable, and outputs 0 for those assignments $\vec{\alpha}$ for which $B(\vec{\alpha},\vec{r})$ is unsatisfiable (the two cases are not necessarily exclusive, so if both cases hold for an assignment, the circuit can output either that the first case holds or that the second case holds). In other words, given a refutation of $A(\vec{p},\vec{q}) \wedge B(\vec{p},\vec{r})$, we construct a circuit $C(\vec{p})$, called *the interpolant*, such that

$$\begin{array}{lll} C(\vec{\alpha}) = 1 & \Longrightarrow & A(\vec{\alpha},\vec{q}) \text{ is unsatisfiable, and} \\ C(\vec{\alpha}) = 0 & \Longrightarrow & B(\vec{\alpha},\vec{r}) \text{ is unsatisfiable.} \end{array} \tag{39}$$

(Note that if U denotes the set of those assignments $\vec{\alpha}$ for which $A(\vec{\alpha},\vec{q})$ is *satisfiable*, and V denotes the set of those assignments $\vec{\alpha}$ for which $B(\vec{\alpha},\vec{r})$ is *satisfiable*, then U and V are disjoint [since $A(\vec{p},\vec{q}) \wedge B(\vec{p},\vec{r})$ is unsatisfiable], and $C(\vec{p})$ separates U from V; see Definition 7.2 below.)

Assume that for a proof system $\mathcal{P}$ the transformation from refutations of $A(\vec{p},\vec{q}), B(\vec{p},\vec{r})$ into the corresponding interpolant circuit $C(\vec{p})$ results in a circuit whose size is polynomial in the size of the refutation. Then, an exponential lower bound on circuits for which (39) holds, implies an exponential lower bound on $\mathcal{P}$-refutations of $A(\vec{p},\vec{q}), B(\vec{p},\vec{r})$.

7.1. **Interpolation for Semantic Refutations.** We now lay out the basic concepts needed to formally describe the feasible interpolation technique. We use the general notion of *semantic refutations* (which generalizes any standard propositional refutation system). We shall use a close terminology to that in [Kra97].

Definition 7.1 (Semantic refutation)**.** Let N be a fixed natural number and let $E_1,\ldots,E_k \subseteq \{0,1\}^N$, where $\bigcap_{i=1}^k E_i = \emptyset$. A *semantic refutation* from $E_1,\ldots,E_k$ is a sequence $D_1,\ldots,D_m \subseteq \{0,1\}^N$ with $D_m = \emptyset$ and such that for every $i \in [m]$, D_i is either one of the E_j's or is deduced from two previous D_j, D_ℓ, $1 \leq j,\ell < i$, by the following *semantic inference rule*:

- From $A, B \subseteq \{0,1\}^N$ deduce any C, such that $C \supseteq (A \cap B)$.

Observe that any standard propositional refutation (with inference rules that derive from at most two proof-lines, a third line) can be regarded as a semantic refutation: just substitute each refutation-line by the set of its satisfying assignments; and by the soundness of the inference rules applied in the refutation, it is clear that each refutation-line (considered as the set of assignments that satisfy it) is deduced by the semantic inference rule from previous refutation-lines.

Definition 7.2 (Separating circuit)**.** Let $\mathcal{U},\mathcal{V} \subseteq \{0,1\}^n$, where $\mathcal{U} \cap \mathcal{V} = \emptyset$, be two disjoint sets. A Boolean circuit C with n input variables is said to *separate $\mathcal{U}$ from $\mathcal{V}$* if $C(\vec{x}) = 1$ for every $\vec{x} \in \mathcal{U}$, and $C(\vec{x}) = 0$ for every $\vec{x} \in \mathcal{V}$. In this case we also say that $\mathcal{U}$ and $\mathcal{V}$ are *separated by C*.

Convention: In what follows we sometime identify a Boolean formula with the set of its satisfying assignments.

Notation: For two (or more) binary strings $u, v \in \{0,1\}^*$, we write (u,v) to denote the concatenation of the u with v (where v comes to the right of u, obviously).

Let $N = n + s + t$ be fixed from now on. Let $A_1,\ldots,A_k \subseteq \{0,1\}^{n+s}$ and let $B_1,\ldots,B_\ell \subseteq \{0,1\}^{n+t}$. Define the following two sets of assignments of length n (formally, 0, 1 strings of length n) that can be extended to satisfying assignments of $A_1,\ldots,A_k$ and $B_1,\ldots,B_\ell$, respectively (formally, those 0, 1 string of length $n+s$ and $n+t$, that are contained in all $A_1,\ldots,A_k$ and

$B_1, \ldots, B_\ell$, respectively):

$$\mathcal{U}_A := \left\{ u \in \{0,1\}^n \;\middle|\; \exists q \in \{0,1\}^s \,,\, (u,q) \in \bigcap_{i=1}^{k} A_i \right\} ,$$

$$\mathcal{V}_B := \left\{ v \in \{0,1\}^n \;\middle|\; \exists r \in \{0,1\}^t \,,\, (v,r) \in \bigcap_{i=1}^{\ell} B_i \right\} .$$

Definition 7.3 (Polynomial upper bounds on interpolants)**.** Let $\mathcal{P}$ be a propositional refutation system. Assume that $\vec{p}, \vec{q}, \vec{r}$ are pairwise disjoint sets of distinct variables, where $\vec{p}$ has n variables, $\vec{q}$ has s variables and $\vec{r}$ has t variables. Let $A_1(\vec{p},\vec{q}), \ldots, A_k(\vec{p},\vec{q})$ and $B_1(\vec{p},\vec{r}), \ldots, B_\ell(\vec{p},\vec{r})$ be two collections of formulas with the displayed variables only. Assume that for any such $A_1(\vec{p},\vec{q}), \ldots, A_k(\vec{p},\vec{q})$ and $B_1(\vec{p},\vec{r}), \ldots, B_\ell(\vec{p},\vec{r})$, if there exists a $\mathcal{P}$-refutation of size S for $A_1(\vec{p},\vec{q}) \wedge \cdots \wedge A_k(\vec{p},\vec{q}) \wedge B_1(\vec{p},\vec{r}) \wedge \ldots \wedge B_\ell(\vec{p},\vec{r})$ then there exists a Boolean circuit separating $\mathcal{U}_A$ from $\mathcal{V}_B$ of size polynomial in S.[10] In this case we say that $\mathcal{P}$ has a *polynomial upper bound on interpolant circuits*.

7.1.1. *The Communication Game Technique.* The *feasible interpolation via communication game technique* is based on transforming proofs into Boolean circuits, where the size of the resulting circuit depends on the communication complexity of each proof-line. This technique goes back to [IPU94] and [Razb95] and was subsequently applied and extended in [BPR97] and [Kra97] ([IPU94] and [BPR97] did not use explicitly the notion of interpolation of tautologies or contradictions). We shall employ the interpolation theorem of Krajíček in [Kra97], that demonstrates how to transform a small semantic refutation with each proof-line having low communication complexity into a small Boolean circuit separating the corresponding sets.

The underlying idea of the interpolation via communication game technique is that a (semantic) refutation, where each proof-line is of small (that is, logarithmic) communication complexity, can be transformed into an efficient communication protocol for the *Karchmer-Wigderson game* (following [KW88]) for two players. In the Karchmer-Wigderson game the first player knows some binary string $u \in U$ and the second player knows some different binary string $v \in V$, where U and V are disjoint sets of strings. The two players communicate by sending information bits to one another (following a protocol previously agreed on). The goal of the game is for the two players to decide on an index i such that the ith bit of u is different from the ith bit of v. An efficient Karchmer-Wigderson protocol (by which we mean a protocol that requires the players to exchange at most a logarithmic number of bits in the worst-case) can then be transformed into a small circuit separating U from V (see Definition 7.2). This efficient transformation from protocols for Karchmer-Wigderson games (described in a certain way) into circuits, was demonstrated by Razborov in [Razb95]. So overall, given a semantic refutation with proof-lines of low communication complexity, one can obtain a small circuit for separating the corresponding sets.

First, we need to define the concept of *communication complexity* in a suitable way for the interpolation theorem.

Definition 7.4 (Communication complexity)**.** Let $N = n + s + t$ and $A \subseteq \{0,1\}^N$. Let $u, v \in \{0,1\}^n$, $q^u \in \{0,1\}^s$, $r^v \in \{0,1\}^t$. Denote by u_i, v_i the ith bit of u, v, respectively, and let (u, q^u, r^v) and (v, q^u, r^v) denote the concatenation of strings u, q^u, r^v and v, q^u, r^v, respectively. Consider the following three tasks:

(1) Decide whether $(u, q^u, r^v) \in A$;

[10] Here $\mathcal{U}_A$ and $\mathcal{V}_B$ are defined as above, by identifying the $A_i(\vec{p},\vec{q})$'s and the $B_i(\vec{p},\vec{r})$'s with the sets of assignments that satisfy them.

(2) Decide whether $(v, q^u, r^v) \in A$;
(3) If one of the following holds:
  (i) $(u, q^u, r^v) \in A$ and $(v, q^u, r^v) \notin A$; or
  (ii) $(u, q^u, r^v) \notin A$ and $(v, q^u, r^v) \in A$,
  then find an $i \in [n]$, such that $u_i \neq v_i$;

Consider a game between two players, Player I and Player II, where Player I knows $u \in \{0,1\}^n, q^u \in \{0,1\}^s$ and Player II knows $v \in \{0,1\}^n, r^v \in \{0,1\}^t$. The two players communicate by exchanging bits of information between them (following a protocol previously agreed on). The *communication complexity of* A, denoted $CC(A)$, is the minimal (over all protocols) number of bits that players I and II need to exchange in the worst-case in solving each of Tasks 1, 2 and 3 above.[11]

For $A \subseteq \{0,1\}^{n+s}$ define

$$\dot{A} := \left\{(a,b,c) \;\middle|\; (a,b) \in A \text{ and } c \in \{0,1\}^t\right\},$$

where a and b range over $\{0,1\}^n$ and $\{0,1\}^s$, respectively. Similarly, for $B \subseteq \{0,1\}^{n+t}$ define

$$\dot{B} := \left\{(a,b,c) \;\middle|\; (a,c) \in B \text{ and } b \in \{0,1\}^t\right\},$$

where a and c range over $\{0,1\}^n$ and $\{0,1\}^t$, respectively.

Theorem 23 ([Kra97])**.** *Let $A_1, \ldots, A_k \subseteq \{0,1\}^{n+s}$ and $B_1, \ldots, B_\ell \subseteq \{0,1\}^{n+t}$. Let $D_1, \ldots, D_m$ be a semantic refutation from $\dot{A}_1, \ldots, \dot{A}_k$ and $\dot{B}_1, \ldots, \dot{B}_\ell$. Assume that $CC(D_i) \leq \zeta$, for all $i \in [m]$. Then, the sets $\mathcal{U}_A$ and $\mathcal{V}_B$ (as defined above) can be separated by a Boolean circuit of size $(m+n)2^{O(\zeta)}$.*

In light of Theorem 23, to demonstrate that a certain propositional refutation system $\mathcal{P}$ possesses a polynomial upper bound on interpolant circuits (see Definition 7.3) it suffices to show that any proof-line of $\mathcal{P}$ induces a set of assignments with at most a logarithmic (in the number of variables) communication complexity (Definition 7.4).

7.2. **Polynomial Upper Bounds on Interpolants for $\mathbf{R^0(lin)}$.** Here we apply Theorem 23 to show that $\mathrm{R}^0(\mathrm{lin})$ has polynomial upper bounds on its interpolant circuits. Again, in what follows we sometime identify a disjunction of linear equations with the set of its satisfying assignments.

Theorem 24. $\mathrm{R}^0(\mathrm{lin})$ *has a polynomial upper bounds on interpolant circuits (Definition 7.3).*

According to the paragraph after Theorem 23, all we need in order to establish Theorem 24 is the following lemma:

Lemma 25. *Let D be an $\mathrm{R}^0(\mathrm{lin})$-line with N variables and let $\widetilde{D}$ be the set of assignments that satisfy D.*[12] *Then, $CC(\widetilde{D}) \leq O(\log N)$.*

Proof: Let $N = n+s+t$ (and so $\widetilde{D} \in \{0,1\}^{n+s+t}$). For the sake of convenience we shall assume that the N variables in D are partitioned into (pairwise disjoint) three groups $\vec{p} := (p_1 \ldots, p_n)$, $\vec{q} := (q_1, \ldots, q_s)$ and $\vec{r} := (r_1, \ldots, r_t)$. Let $u, v \in \{0,1\}^n$, $q^u \in \{0,1\}^s$, $r^v \in \{0,1\}^t$. Assume that Player I knows u, q^u and Player II knows v, r^v.

[11] In other words, $CC(A)$ is the minimal number ζ, for which there exists a protocol, such that for every input (u, q^u to Player I and v, r^v to Player II) and every task (from Tasks 1, 2 and 3), the players need to exchange at most ζ bits in order to solve the task.

[12] The notation $\widetilde{D}$ has nothing to do with the same notation used in Section 3.

By the definition of an $\mathrm{R}^0(\mathrm{lin})$-line (see Definition 3.2) we can partition the disjunction D into a *constant number* of disjuncts, where one disjunct is a (possibly empty, translation of a) clause in the $\vec{p}, \vec{q}, \vec{r}$ variables (see Section 3.1), and all other disjuncts have the following form:

$$\bigvee_{i \in I} \left(\vec{a} \cdot \vec{p} + \vec{b} \cdot \vec{q} + \vec{c} \cdot \vec{r} = \ell_i \right) , \tag{40}$$

where I is (an unbounded) set of indices, ℓ_i are integer numbers, for all $i \in I$, and $\vec{a}, \vec{b}, \vec{c}$ denote vectors of n, s and t constant coefficients, respectively.

Let us denote the (translation of the) clause from D in the $\vec{p}, \vec{q}, \vec{r}$ variables by

$$P \vee Q \vee R \, ,$$

where P, Q and R denote the (translated) sub-clauses consisting of the $\vec{p}$, $\vec{q}$ and $\vec{r}$ variables, respectively.

We need to show that by exchanging $O(\log N)$ bits, the players can solve each of Tasks 1, 2 and 3 from Definition 7.4, correctly.

Task 1: The players need to decide whether $(u, q^u, r^v) \in \widetilde{D}$. Player II, who knows r^v, computes the numbers $\vec{c} \cdot\ r^v$, for every $\vec{c}$ pertaining to every disjunct of the form shown in Equation (40) above. Then, Player II sends the (binary representation of) these numbers to Player I. Since there are only a constantly many such numbers and the coefficients in every $\vec{c}$ are also constants, this amounts to $O(\log t) \leq O(\log N)$ bits that Player II sends to Player I. Player II also computes the truth value of the sub-clause R, and sends this (single-bit) value to Player I.

Now, it is easy to see that Player I has sufficient data to compute by herself/himself whether $(u, q^u, r^v) \in \widetilde{D}$ (Player I can then send a single bit informing Player II whether $(u, q^u, r^v) \in \widetilde{D}$).

Task 2: This is analogous to Task 1.

Task 3: Assume that $(u, q^u, r^v) \in \widetilde{D}$ and $(v, q^u, r^v) \notin \widetilde{D}$ (the case $(u, q^u, r^v) \notin \widetilde{D}$ and $(v, q^u, r^v) \in \widetilde{D}$ is analogous).

The first rounds of the protocol are completely similar to that described in Task 1 above: Player II, who knows r^v, computes the numbers $\vec{c} \cdot\ r^v$, for every $\vec{c}$ pertaining to every disjunct of the form shown in Equation (40) above. Then, Player II sends the (binary representation of) these numbers to Player I. Player II also computes the truth value of the sub-clause R, and sends this (single-bit) value to Player I. Again, this amounts to $O(\log N)$ bits that Player II sends to Player I.

By assumption (that $(u, q^u, r^v) \in \widetilde{D}$ and $(v, q^u, r^v) \notin \widetilde{D}$) the players need to deal only with the following two cases:

Case 1: The assignment (u, q^u, r^v) satisfies the clause $P \vee Q \vee R$ while (v, q^u, r^v) falsifies $P \vee Q \vee R$. Thus, it must be that $\vec{u}$ satisfies the sub-clause P while $\vec{v}$ falsifies P. This means that for any $i \in [n]$ such that u_i sets to 1 a literal in P (there ought to exist at least one such i), it must be that $u_i \neq v_i$. Therefore, all that Player I needs to do is to send the (binary representation of) index i to Player II. (This amounts to $O(\log N)$ bits that Player I sends to Player II.)

Case 2: There is some linear equation

$$\vec{a} \cdot \vec{p} + \vec{b} \cdot \vec{q} + \vec{c} \cdot \vec{r} = \ell \tag{41}$$

in D, such that $\vec{a} \cdot u + \vec{b} \cdot q^u + \vec{c} \cdot r^v = \ell$. Note that (by assumption that $(v, q^u, r^v) \notin \widetilde{D}$) it must also hold that: $\vec{a} \cdot v + \vec{b} \cdot q^u + \vec{c} \cdot r^v \neq \ell$ (and so there is an $i \in [n]$, such that $u_i \neq v_i$). Player I can find linear equation (41), as he/she already received from Player II all the possible values of $\vec{c} \cdot \vec{r}$ (for all possible $\vec{c}$'s in D).

Recall that the left hand side of a linear equation $\vec{d}\cdot\vec{x}=\ell$ is called the *linear form* of the equation. By the definition of an $\mathrm{R}^0(\mathrm{lin})$-line there are only constant many distinct linear forms in D. Since both players know these linear forms, we can assume that each linear form has some index associated to it by both players. Player I sends to Player II the index of the linear form $\vec{a}\cdot\vec{p}+\vec{b}\cdot\vec{q}+\vec{c}\cdot\vec{r}$ from (41) in D. Since there are only *constantly* many such linear forms in D, it takes only constant number of bits to send this index.

Now both players need to apply a protocol for finding an $i\in[n]$ such that $u_i\neq v_i$, where $\vec{a}\cdot\vec{u}+\vec{b}\cdot q^u+\vec{c}\cdot r^v=\ell$ and $\vec{a}\cdot\vec{v}+\vec{b}\cdot q^u+\vec{c}\cdot r^v\neq\ell$. Thus, it remains only to prove the following claim:

Claim 4. There is a communication protocol in which Player I and Player II need at most $O(\log N)$ bits of communication in order to find an $i\in[n]$ such that $u_i\neq v_i$ (under the above conditions).

Proof of claim: We invoke the well-known connection between Boolean circuit-depth and communication complexity. Let $f:\{0,1\}^N\to\{0,1\}$ be a Boolean function. Denote by $\mathrm{dp}(f)$ the minimal depth of a Boolean circuit computing f. Consider a game between two players: Player I knows some $\vec{x}\in\{0,1\}^N$ and Player II knows some other $\vec{y}\in\{0,1\}^N$, such that $f(\vec{x})=1$ while $f(\vec{y})=0$. The goal of the game is to find an $i\in[N]$ such that $x_i\neq y_i$. Denote by $\mathrm{CC}'(f)$ the minimal number of bits needed for the two players to communicate (in the worst case[13]) in order to solve this game.[14] Then, for any function f it is known that $\mathrm{dp}(f)=\mathrm{CC}'(f)$ (see [KW88]).

Therefore, to conclude the proof of the claim it is enough to establish that the function $f:\{0,1\}^N\to\{0,1\}$ that receives the input variables $\vec{p},\vec{q},\vec{r}$ and computes the truth value of $\vec{a}\cdot\vec{p}+\vec{b}\cdot\vec{q}+\vec{c}\cdot\vec{r}=\ell$ has Boolean circuit of depth $O(\log N)$. In case all the coefficients in $\vec{a},\vec{b},\vec{c}$ are 1, it is easy to show[15] that there is a Boolean circuit of depth $O(\log N)$ that computes the function f. In the case that the coefficients in $\vec{a},\vec{b},\vec{c}$ are all constants, it is easy to show, by a reduction to the case where all coefficients are 1,[16] that there is a Boolean circuit of depth $O(\log N)$ that computes the function f. We omit the details. ■ ■

8. Size Lower Bounds

In this section we establish an exponential-size lower bound on $\mathrm{R}^0(\mathrm{lin})$ refutations of the clique-coloring formulas. We shall employ the theorem of Bonet, Pitassi & Raz in [BPR97] that provides exponential-size lower bounds for any semantic refutation of the clique-coloring formulas, having low communication complexity in each refutation-line.

First we recall the strong lower bound obtained by Alon & Boppana [AB87] (improving over [Razb85]; see also [And85]) for the (monotone) *clique separator* functions, defined as follows (a function $f:\{0,1\}^n\to\{0,1\}$ is called *monotone* if for all $\alpha\in\{0,1\}^n$, $\alpha'\geq\alpha$ implies $f(\alpha')\geq f(\alpha)$):

[13] Over all inputs $\vec{x},\vec{y}$ such that $f(\vec{x})=1$ and $f(\vec{y})=0$.

[14] The measure CC' is basically the same as CC defined earlier.

[15] Using the known $O(\log N)$-depth Boolean circuits for the threshold functions.

[16] For instance, consider the simple case where we have only a single variable. That is, let c be a constant and assume that we wish to construct a circuit that computes $c\cdot x=\ell$, for some integer ℓ. Then, we take a circuit that computes the function $f:\{0,1\}^c\to\{0,1\}$ that outputs the truth value of $y_1+\ldots+y_c=\ell$ (thus, in f all coefficients are 1's); and to compute $c\cdot x=\ell$ we only have to substitute each y_i in the circuit with the variable x.

Definition 8.1 (Clique separator)**.** A monotone boolean function $Q^n_{k,k'}$ is called a *clique separator* if it interprets its inputs as the edges of a graph on n vertices, and outputs 1 on every input representing a k-clique, and 0 on every input representing a complete k'-partite graph (see Section 6.3).

Recall that a *monotone Boolean circuit* is a circuit that uses only monotone Boolean gates (for instance, only the fan-in two gates $\wedge, \vee$).

Theorem 26 ([AB87])**.** *Let k, k' be integers such that $3 \leq k' < k$ and $k\sqrt{k'} \leq n/(8 \log n)$, then every monotone Boolean circuit that computes a clique separator function $Q^n_{k,k'}$ requires size at least*

$$\frac{1}{8}\left(\frac{n}{4k\sqrt{k'}\log n}\right)^{(\sqrt{k'}+1)/2}.$$

For the next theorem, we need a slightly different (and weaker) version of communication complexity, than that in Definition 7.4.

Definition 8.2 (Communication complexity (second definition))**.** Let X denote n Boolean variables $x_1, \ldots, x_n$, and let S_1, S_2 be a partition of X into two disjoint sets of variables. The communication complexity of a Boolean function $f : \{0,1\}^n \to \{0,1\}$ is the number of bits needed to be exchanged by two players, one knowing the values given to the S_1 variables and the other knowing the values given to S_2 variables, in the worst-case, over all possible partitions S_1 and S_2.

Theorem 27 ([BPR97])**.** *Every semantic refutation of $\neg\text{CLIQUE}^n_{k,k'}$ (for $k' < k$) with m refutation-lines and where each refutation-line (considered as a the characteristic function of the line) has communication complexity (as in Definition 8.2) ζ, can be transformed into a monotone circuit of size $m \cdot 2^{3\zeta+1}$ that computes a separating function $Q^n_{k,k'}$.*

In light of Theorem 26, in order to be able to apply Theorem 27 to $\mathrm{R}^0(\mathrm{lin})$, and arrive at an exponential-size lower bound for $\mathrm{R}^0(\mathrm{lin})$ refutations of the clique-coloring formulas, it suffices to show that $\mathrm{R}^0(\mathrm{lin})$ proof-lines have logarithmic communication complexity:

Lemma 28. *Let D be an $\mathrm{R}^0(\mathrm{lin})$-line with N variables. Then, the communication complexity (as in Definition 8.2) of D is at most $O(\log N)$ (where D is identified here with the characteristic function of D).*

Proof: The proof is similar to the proof of Lemma 25 for solving Task 1 (and the analogous Task 2) in Definition 7.4. ∎

By direct calculations we obtain the following lower bound from Theorems 26, 27 and Lemma 28:

Corollary 29. *Let k be an integer such that $3 \leq k' = k-1$ and assume that $\frac{1}{2} \cdot n/(8\log n) \leq k\sqrt{k} \leq n/(8\log n)$. Then, for all $\varepsilon < 1/3$, every $\mathrm{R}^0(\mathrm{lin})$ refutation of $\neg\text{CLIQUE}^n_{k,k'}$ is of size at least $2^{\Omega(n^\varepsilon)}$.*

When considering the parameters of Theorem 20, we obtain a super-polynomial separation between $\mathrm{R}^0(\mathrm{lin})$ refutations and $\mathrm{R}(\mathrm{lin})$ refutations, as described below.

From Theorems 26,27 and Lemma 28 we have (by direct calculations):

Corollary 30. *Let $k = \sqrt{n}$ and $k' = (\log n)^2/8\log\log n$. Then, every $\mathrm{R}^0(\mathrm{lin})$ refutation of $\neg\text{CLIQUE}^n_{k,k'}$ has size at least $n^{\Omega\left(\frac{\log n}{\sqrt{\log\log n}}\right)}$.*

By Corollary 21, R(lin) admits polynomial-size in n refutations of $\neg\text{CLIQUE}^n_{k,k'}$ under the parameters in Corollary 30. Thus we obtain the following separation result:

Corollary 31. R(lin) *is super-polynomially stronger than* R^0(lin).

Comment 1. Note that we do not need to assume that the coefficients in R^0(lin)-lines are constants for the lower bound argument. If the coefficients in R^0(lin)-lines are only polynomially bounded (in the number of variables) then the same lower bound as in Corollary 30 also applies. This is because R^0(lin)-lines in which coefficients are polynomially bounded integers, still have low (that is, logarithmic) communication complexity (as in Definition 8.2).

9. Applications to Multilinear Proofs

In this section we arrive at one of the main benefits of the work we have done so far; Namely, applying results on resolution over linear equations in order to obtain new results for multilinear proof systems. Subsection 9.1 that follows, contains definitions, sufficient for the current paper, concerning the notion of multilinear proofs introduced in [RT06].

9.1. Background on Algebraic and Multilinear Proofs.

9.1.1. *Arithmetic and Multilinear Formulas.*

Definition 9.1 (Arithmetic formula)**.** Fix a field $\mathbb{F}$. An *arithmetic formula* is a tree, with edges directed from the leaves to the root, and with unbounded (finite) fan-in. Every leaf of the tree (namely, 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 tree. Every other node of the tree 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 formula F is the total number of nodes in its graph and is denoted by $|F|$. An arithmetic formula 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 (namely, 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 formula is the polynomial computed by the root. The *depth* of a formula F is the maximal number of edges in a path from a leaf to the root of F.

We say that an arithmetic formula has a *plus (resp., product) gate at the root* if the root of the formula is labeled with a plus (resp., product) gate.

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

Definition 9.2 (Multilinear formula)**.** An arithmetic formula is *a multilinear formula* (or equivalently, *multilinear arithmetic formula*) if the polynomial computed by *each* gate of the formula is multilinear (as a formal polynomial, that is, as an element of $\mathbb{F}[x_1, \ldots, x_n]$).

An additional definition we shall need is the following linear operator, called the *multilinearization operator*:

Definition 9.3 (Multilinearization operator)**.** 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 $q = x_1^2 x_2 + a x_4^3$ (for some $a \in \mathbb{F}$) then $\mathbf{M}[q] = x_1 x_2 + a x_4$.

The simulation of $\mathrm{R}^0(\mathrm{lin})$ by multilinear proofs will rely heavily on the fact that multilinear symmetric polynomials have small depth-3 multilinear formulas over fields of characteristic 0 (see [SW01] for a proof of this fact). To this end we define precisely the concept of symmetric polynomials.

A *renaming* of the variables $x_1, \ldots, x_n$ is a permutation $\sigma \in S_n$ (the symmetric group on $[n]$) such that x_i is mapped to $x_{\sigma(i)}$ for every $1 \leq i \leq n$.

Definition 9.4 (Symmetric polynomial)**.** Given a set of variables $X = \{x_1, \ldots, x_n\}$, a *symmetric polynomial* f over X is a polynomial in (all the variables of) X such that renaming of variables does not change the polynomial (as a formal polynomial).

9.1.2. *Polynomial Calculus with Resolution.* Here we define the PCR proof system, introduced by Alekhnovich *et al.* in [ABSRW02].

Definition 9.5 (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 the ring of polynomials $\mathbb{F}[x_1, \ldots, x_n, \bar{x}_1, \ldots, \bar{x}_n]$. The variables $\bar{x}_1, \ldots, \bar{x}_n$ are treated as new formal variables. Call the set of polynomials $x^2 - x$, for $x \in \{x_1, \ldots, x_n, \bar{x}_1, \ldots, \bar{x}_n\}$, plus the polynomials $x_i + \bar{x}_i - 1$, for all $1 \leq i \leq n$, the set of *Boolean axioms of PCR*. A *PCR proof* from Q of a polynomial g is a finite sequence $\pi = (p_1, ..., p_\ell)$ of multivariate polynomials from $\mathbb{F}[x_1, \ldots, x_n, \bar{x}_1, \ldots, \bar{x}_n]$ (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;
From p deduce $\bar{x}_i \cdot p$, for some variable $\bar{x}_i$;
Addition: From p and q deduce $\alpha \cdot p + \beta \cdot q$, for some $\alpha, \beta \in \mathbb{F}$.

A *PCR refutation of* Q is a proof of 1 (which is interpreted as $1 = 0$) from Q. The *number of steps* in a PCR proof is the number of proof-lines in it (that is, ℓ in the case of π above).

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 $x_i = 0$.

9.1.3. *Multilinear Proof Systems.* In [RT06] the authors introduced a natural (semantic) algebraic proof system that operates with multilinear arithmetic formulas denoted fMC (which stands for *formula multilinear calculus*), defined as follows:

Definition 9.6 (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 $1 \leq i \leq n$.

Definition 9.7 (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.

9.2. **From R(lin) Proofs to PCR Proofs.** We now demonstrate a general and straightforward translation from R(lin) proofs into PCR proofs over fields of characteristic 0. We use the term "translation" in order to distinguish it from a *simulation*; since here we are not interested in the size of PCR proofs. In fact we have not defined the size of PCR proofs at all. We shall be interested only in the *number of steps* in PCR proofs.

From now on, all polynomials and arithmetic formulas are considered over some fix field $\mathbb{F}$ of characteristic 0. Recall that any field of characteristic 0 contains (an isomorphic copy of) the integer numbers, and so we can use integer coefficients in the field.

Definition 9.8 (Polynomial translation of R(lin) proof-lines)**.** Let D be a disjunction of linear equations:

$$\left(a_1^{(1)}x_1 + \ldots + a_n^{(1)}x_n = a_0^{(1)}\right) \vee \cdots \vee \left(a_1^{(t)}x_1 + \ldots + a_n^{(t)}x_n = a_0^{(t)}\right). \tag{42}$$

We denote by $\widehat{D}$ its translation into the following polynomial:[17]

$$\left(a_1^{(1)}x_1 + \ldots + a_n^{(1)}x_n - a_0^{(1)}\right) \cdots \left(a_1^{(t)}x_1 + \ldots + a_n^{(t)}x_n - a_0^{(t)}\right). \tag{43}$$

If D is the *empty disjunction*, we define $\widehat{D}$ to be the polynomial 1.

It is clear that every $0,1$ assignment to the variables in D, satisfies D, if and only if $\widehat{D}$ evaluates to 0 under the assignment.

Proposition 3. *Let $\pi = (D_1, \ldots, D_\ell)$ be an R(lin) proof sequence of D_ℓ, from some collection of initial disjunctions of linear equations $Q_1, \ldots, Q_m$. Then, there exists a PCR proof of $\widehat{D}_\ell$ from $\widehat{Q}_1, \ldots, \widehat{Q}_m$ with at most a polynomial in $|\pi|$ number of steps.*

Proof: We proceed by induction on the number of lines in π.

The base case is the translation of the axioms of R(lin) via the translation scheme in Definition 9.8. An R(lin) Boolean axiom $(x_i = 0) \vee (x_i = 1)$ is translated into $x_i \cdot (x_i - 1)$ which is already a Boolean axiom of PCR.

For *the induction step*, we translate every R(lin) inference rule application into a polynomial-size PCR proof sequence as follows. We use the following simple claim:

Claim 5. Let p and q be two polynomials and let s be the minimal size of an arithmetic formula computing q. Then one can derive in PCR, with only a polynomial in s number of steps, from p the product $q \cdot p$.[18]

Proof of claim: By induction on s. ∎

[17] This notation should not be confused with the same notation in Section 6.3.

[18] Again, note that we only require that the number of steps in the proof is polynomial. We do not consider here the *size* of the PCR proof.

Assume that $D_i = D_j \vee L$ was derived from D_j using the Weakening inference rule of R(lin), where $j < i \leq \ell$ and L is some linear equation. Then, by Claim 5, $\widehat{D}_i = \widehat{D}_j \cdot \widehat{L}$ can be derived from $\widehat{D}_j$ with a derivation of at most polynomial in $|D_j \vee L|$ many steps.

Assume that D_i was derived from D_j where D_j is $D_i \vee (0 = k)$, using the Simplification inference rule of R(lin), where $j < i \leq \ell$ and k is a non-zero integer. Then, $\widehat{D}_i$ can be derived from $\widehat{D}_j = \widehat{D}_i \cdot -k$ by multiplying with $-k^{-1}$ (via the Addition rule of PCR).

Thus, it remains to simulate the *resolution rule* application of R(lin). Let A, B be two disjunctions of linear equations and assume that $A \vee B \vee ((\vec{a} + \vec{b}) \cdot \vec{x} = a_0 + b_0)$ was derived in π from $A \vee (\vec{a} \cdot \vec{x} = a_0)$ and $B \vee (\vec{b} \cdot \vec{x} = b_0)$ (the case where $A \vee B \vee ((\vec{a} - \vec{b}) \cdot \vec{x} = a_0 - b_0)$ was derived from $A \vee (\vec{a} \cdot \vec{x} = a_0)$ and $B \vee (\vec{b} \cdot \vec{x} = b_0)$, is similar).

We need to derive $\widehat{A} \cdot \widehat{B} \cdot ((\vec{a} + \vec{b}) \cdot \vec{x} - a_0 - b_0)$ from $\widehat{A} \cdot (\vec{a} \cdot \vec{x} - a_0)$ and $\widehat{B} \cdot (\vec{b} \cdot \vec{x} - b_0)$. This is done by multiplying $\widehat{A} \cdot (\vec{a} \cdot \vec{x} - a_0)$ with $\widehat{B}$ and multiplying $\widehat{B} \cdot (\vec{b} \cdot \vec{x} - b_0)$ with $\widehat{A}$ (using Claim 5), and then adding the resulted polynomials together. ∎

Remark 4. When translating R(lin) proofs into PCR proofs we actually do not make any use of the "negative" variables $\bar{x}_1, \ldots, \bar{x}_n$. Nevertheless, the multilinear proof systems make use of these variables in order to polynomially simulate PCR proofs (see Theorem 33 and its proof in [RT06]).

We shall need the following corollary in the sequel:

Corollary 32. *Let $\pi = D_1, \ldots, D_\ell$ be an* $\mathrm{R}^0(\mathrm{lin})$ *proof of D_ℓ, and let s be the maximal size of an* $\mathrm{R}^0(\mathrm{lin})$*-line in π. Then there is a PCR proof π' of $\widehat{D}_\ell$ with polynomial-size in $|\pi|$ number of steps and such that every line of π' is a translation (via Definition 9.8) of an* $\mathrm{R}^0(\mathrm{lin})$*-line (Definition 3.2), where the size of the* $\mathrm{R}^0(\mathrm{lin})$*-line is polynomial in s.*

Proof: The simulation of R(lin) by PCR shown above, can be thought of as, first, considering $\widehat{D}_1, \ldots, \widehat{D}_\ell$ as the "skeleton" of a PCR proof of $\widehat{D}_\ell$. And second, for each D_i that was deduced by one of R(lin)'s inference rules from previous lines, one inserts the corresponding PCR proof sequence that simulates the appropriate inference rule application (as described in the proof of Proposition 3). By definition, those PCR proof-lines that correspond to lines in the skeleton $\widehat{D}_1, \ldots, \widehat{D}_\ell$ are translations of $\mathrm{R}^0(\mathrm{lin})$-lines (with size at most polynomial in s). Thus, to conclude the proof of the corollary, one needs only to check that for any $\mathrm{R}^0(\mathrm{lin})$-line D_i that was deduced by one of R(lin)'s inference rules from previous $\mathrm{R}^0(\mathrm{lin})$-lines (as demonstrated in the proof of Proposition 3), the inserted corresponding PCR proof sequence uses only translations of $\mathrm{R}^0(\mathrm{lin})$-lines (with size polynomial in s). This can be verified by a straightforward inspection. ∎

9.3. **From PCR Proofs to Multilinear Proofs.** We now recall the general simulation result proved in [RT06] stating 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 (that is, the number of proof lines in the PCR proof sequence is polynomial). If the 'multilinearization' (namely, the result of applying the $\mathbf{M}[\cdot]$ operator – see Definition 9.3) 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. More formally, we have:

Theorem 33 ([RT06])**.** *Fix a field $\mathbb{F}$ (not necessarily of characteristic 0) 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, that is, $s = \Sigma_{i=1}^{m}|\Phi_i|$, 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.*

9.3.1. *Depth-3 Multilinear Proofs.* Here we show that multilinear proofs operating with depth-3 multilinear formulas (that is, depth-3 fMC) over fields of characteristic 0 polynomially simulate $\mathrm{R}^0(\mathrm{lin})$ proofs. In light of Proposition 32 and Theorem 33, to this end it suffices to show that any $\mathrm{R}^0(\mathrm{lin})$-line D translates into a corresponding polynomial p (via the translation in Definition 9.8) such that $\mathbf{M}[p]$ has a multilinear formula of size polynomial (in the number of variables) and depth at most 3 (with a plus gate at the root) over fields of characteristic 0.

We need the following proposition from [RT06]:

Proposition 4 ([RT06])**.** *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 $\Sigma_{i=1}^{c}|X_i| = n$. 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 n), with a plus gate at the root.*

The following is the key lemma of the simulation:

Lemma 34. *Let D be an $\mathrm{R}^0(\mathrm{lin})$-line with n variables and let $p = \widehat{D}$ (see Definition 9.8). Then, $\mathbf{M}[p]$ has a depth-3 multilinear formula over fields of characteristic 0, with a plus gate at the root and size at most polynomial in the size of D.*

Proof: Assume that the underlying variables of D are $\vec{x} = x_1 \ldots, x_n$. By the definition of an $\mathrm{R}^0(\mathrm{lin})$-line (see Definition 3.2) we can partition the disjunction D into a *constant number* of disjuncts, where one disjunct is a (possibly empty, translation of a) clause C,[19] and all other disjuncts have the following form:

$$\bigvee_{i=1}^{m} (\vec{a} \cdot \vec{x} = \ell_i) \ , \tag{44}$$

where the ℓ_i's are integers, m is not necessarily bounded and $\vec{a}$ denotes a vector of n *constant* integer coefficients.

Let us denote by q the polynomial representing the clause C.[20]

Consider a disjunct as shown in (44). Since the coefficients $\vec{a}$ are constants, $\vec{a} \cdot \vec{x}$ can be written as a sum of constant number of linear forms, each with the *same* constant coefficient. In other words, $\vec{a} \cdot \vec{x}$ can be written as $z_1 + \ldots + z_d$, for some constant d, where for all $i \in [d]$:

$$z_i := b \cdot \sum_{j \in J} x_j \ , \tag{45}$$

for some $J \subseteq [n]$ and some constant integer b. We shall assume without loss of generality that d is the same constant for every disjunct of the form (44) inside D (otherwise, take d to be the maximal such d).

Thus, (44) is translated (via the translation scheme in Definition 9.8) into:

$$\prod_{i=1}^{m} (z_1 + \ldots + z_d - \ell_i) \ . \tag{46}$$

[19] If there is more than one clause in D, we simply combine all the clauses into a single clause.

[20] C is a translation of a clause (that is, disjunction of literals) into a disjunction of linear equations, as defined in Section 3.1. The polynomial q is then the polynomial translation of this disjunction of linear equations, as in Definition 9.8.

By fully expanding the product in (46), we arrive at:

$$\sum_{r_1+\ldots+r_{d+1}=m} \left(\alpha_{r_{d+1}} \cdot \prod_{k=1}^{d} z_k^{r_k} \right), \tag{47}$$

where the r_i's are non-negative integers, and where the α_r's, for every $0 \leq r \leq m$ are just integer coefficients, formally defined as follows (this definition is not essential; we present it only for the sake of concreteness):

$$\alpha_r := \sum_{\substack{U \subseteq [m] \\ |U|=r}} \prod_{j \in U} (-\ell_j) \,. \tag{48}$$

Claim 6. The polynomial $\widehat{D}$ (the polynomial translation of D) is a linear combination (over $\mathbb{F}$) of polynomially (in $|D|$) many terms, such that each term can be written as

$$q \cdot \prod_{k \in K} z_k^{r_k} \,,$$

where K is a collection of a constant number of indices, r_k's are non-negative integers, and the z_k's and q are as above (that is, the z_k's are linear forms, where each z_k has a single coefficient for all variables in it, as in (45), and q is a polynomial translation of a clause).

Proof of claim: Denote the total number of disjuncts of the form (44) in D by h. By definition (of R^0(lin)-line), h is a constant. Consider the polynomial (47) above. In $\widehat{D}$, we actually need to multiply h many polynomials of the form shown in (47) and the polynomial q.

For every $j \in [h]$ we write the (single) linear form in the jth disjunct as a sum of constantly many linear forms $z_{j,1} + \ldots + z_{j,d}$, where each linear form $z_{j,k}$ has the same coefficient for every variable in it. Thus, $\widehat{D}$ can be written as:

$$q \cdot \prod_{j=1}^{h} \left(\sum_{r_1+\ldots+r_{d+1}=m_j} \underbrace{\left(\alpha^{(j)}_{r_{d+1}} \cdot \prod_{k=1}^{d} z_{j,k}^{r_k} \right)}_{(\star)} \right), \tag{49}$$

(where the m_j's are not bounded, and the coefficients $\alpha^{(j)}_{r_{d+1}}$ are as defined in (48) except that here we add the index (j) to denote that they depend on the jth disjunct in D). Denote the maximal m_j, for all $j \in [h]$, by m_0. The size of D, denoted $|D|$, is at least m_0. Note that since d is a constant, the number of summands in each (middle) sum in (49) is polynomial in m_0, which is at most polynomial in $|D|$. Thus, by expanding the outermost product in (49), we arrive at a sum of polynomially in $|D|$ many summands. Each summand in this sum is a product of h terms of the form $(\star)$ multiplied by q. ∎

It remains to apply the multilinearization operator (Definition 9.3) on $\widehat{D}$, and verify that the resulting polynomial has a depth-3 multilinear formula with a plus gate at the root and of polynomial-size (in $|D|$). Since $\mathbf{M}[\cdot]$ is a linear operator, it suffices to show that when applying $\mathbf{M}[\cdot]$ on each summand in $\widehat{D}$, as described in Claim 6, one obtains a (multilinear) polynomial that has a depth-3 multilinear formula with a plus gate at the root, and of polynomial-size in the number of variables n (note that clearly $n \leq |D|$). This is established in the following claim:

Claim 7. The polynomial $\mathbf{M}\left[q \cdot \prod_{k \in K} z_k^{r_k}\right]$ has a depth-3 multilinear formula of polynomial-size in n (the overall number of variables) and with a plus gate at the root (over fields of characteristic 0), under the same notation as in Claim 6.

Proof of claim: Recall that a power of a symmetric polynomial is a symmetric polynomial in itself. Since each z_k (for all $k \in K$) is a symmetric polynomial, then its power $z_k^{r_k}$ is also symmetric. The polynomial q is a translation of a clause, hence it is a product of two symmetric polynomials: the symmetric polynomial that is the translation of the disjunction of literals with positive signs, and the symmetric polynomial that is the translation of the disjunction of literals with negative signs. Therefore, $q \cdot \prod_{k \in K} z_k^{r_k}$ is a product of constant number of symmetric polynomials. By Proposition 4, $\mathbf{M}\left[q \cdot \prod_{k \in K} z_k^{r_k}\right]$ (where here the $\mathbf{M}[\cdot]$ operator operates on the $\vec{x}$ variables in the z_k's and q) is a polynomial for which there is a polynomial-size (in n) depth-3 multilinear formula with a plus gate at the root (over fields of characteristic 0). ■ ■

We now come to the main corollary of this section.

Corollary 35. *Multilinear proofs operating with depth-*3 *multilinear formulas (that is, depth-*3 fMC *proofs) polynomially-simulate* $\mathrm{R}^0(\mathrm{lin})$ *proofs.*

Proof: Immediate from Corollary 32, Theorem 33 and Proposition 34.

For the sake of clarity we repeat the chain of transformations needed to prove the simulation. Given an $\mathrm{R}^0(\mathrm{lin})$ proof π, we first use Corollary 32 to transform π into a PCR proof π', with number of steps that is at most polynomial in $|\pi|$, and where each line in π' is a polynomial translation of some $\mathrm{R}^0(\mathrm{lin})$-line with size at most polynomial in the maximal line in π (which is clearly at most polynomial in $|\pi|$). Thus, by Proposition 34 each polynomial in π' has a corresponding multilinear polynomial with a polynomial-size in $|\pi|$ depth-3 multilinear formula (and a plus gate at the root). Therefore, by Theorem 33, we can transform π' into a depth-3 fMC proof with only a polynomial (in $|\pi|$) increase in size. ■

9.4. **Small Depth-**3 **Multilinear Proofs.** Since $\mathrm{R}^0(\mathrm{lin})$ admits polynomial-size (in n) refutations of the m to n pigeonhole principle (for any $m > n$) (as defined in 6.1), Corollary 35 and Theorem 15 yield:

Theorem 36. *For any $m > n$ there are polynomial-size (in n) depth-*3 fMC *refutations of the m to n pigeonhole principle* PHP_n^m *(over fields of characteristic* 0*).*

This improves over the result in [RT06] that demonstrated a polynomial-size (in n) depth-3 fMC refutations of a weaker principle, namely the m to n *functional* pigeonhole principle.

Furthermore, corollary 35 and Theorem 19 yield:

Theorem 37. *Let G be an r-regular graph with n vertices, where r is a constant, and fix some modulus p. Then there are polynomial-size (in n) depth-*3 fMC *refutations of Tseitin* mod p *formulas* $\neg\textsc{Tseitin}_{G,p}$ *(over fields of characteristic* 0*).*

The polynomial-size refutations of Tseitin graph tautologies here are different than those demonstrated in [RT06]. Theorem 37 establishes polynomial-size refutations over any field of characteristic 0 of Tseitin mod p formulas, whereas [RT06] required the field to contain a primitive pth root of unity. On the other hand, the refutations in [RT06] of Tseitin mod p formulas do not make any use of the semantic nature of the fMC proof system, in the sense that they do not utilize the fact that the base field is of characteristic 0 (which in turn enables one to efficiently represent any symmetric [multilinear] polynomial by a depth-3 multilinear formula).

10. Relations with Extensions of Cutting Planes

In this section we tie some loose ends by showing that, in full generality, R(lin) polynomially simulates R(CP) with polynomially bounded coefficients, denoted R(CP*). First we define the

R(CP*) proof system – introduced in [Kra98] – which is a common extension of resolution and CP* (the latter is cutting planes with polynomially bounded coefficients). The system R(CP*), thus, is essentially resolution operating with disjunctions of linear inequalities (with polynomially bounded integral coefficients) augmented with the cutting planes inference rules.

A linear inequality is written as

$$\vec{a} \cdot \vec{x} \geq a_0 \,, \tag{50}$$

where $\vec{a}$ is a vector of integral coefficients $a_1, \ldots, a_n$, $\vec{x}$ is a vector of variables $x_1, \ldots, x_n$, and a_0 is an integer. The *size* of the linear inequality (50) is the sum of all $a_0, \ldots, a_n$ written in *unary notation* (this is similar to the size of linear equations in R(lin)). A *disjunction of linear inequalities* is just a disjunction of inequalities of the form in (50). The semantics of a disjunction of inequalities is the natural one, that is, a disjunction is true under an assignment of integral values to $\vec{x}$ if and only if at least one of the inequalities is true under the assignment. The *size of a disjunction of linear inequalities* is the total size of all linear inequalities in it. We can also add in the obvious way linear inequalities, that is, if L_1 is the linear inequality $\vec{a} \cdot \vec{x} \geq a_0$ and L_2 is the linear inequality $\vec{b} \cdot \vec{x} \geq b_0$, then $L_1 + L_2$ is the linear inequality $(\vec{a} + \vec{b}) \cdot \vec{x} \geq a_0 + b_0$.

The proof system R(CP*) operates with disjunctions of linear inequalities with integral coefficients (written in *unary* representation), and is defined as follows (our formulation is similar to that in [Koj07]):[21]

Definition 10.1 (R(CP*))**.** Let $K := \{K_1, \ldots, K_m\}$ be a collection of disjunctions of linear inequalities (whose coefficients are written in unary representation). An *R(CP*)-proof from K of a disjunction of linear inequalities* D is a finite sequence $\pi = (D_1, ..., D_\ell)$ of disjunctions of linear inequalities, such that $D_\ell = D$ and for each $i \in [\ell]$: either $D_i = K_j$ for some $j \in [m]$; or D_i is one of the following *R(CP*)-axioms*:

(1) $x_i \geq 0$, for any variable x_i;
(2) $-x_i \geq -1$, for any variable x_i;
(3) $(\vec{a} \cdot \vec{x} \geq a_0) \vee (-\vec{a} \cdot \vec{x} \geq 1 - a_0)$, where all coefficients (including a_0) are integers;

or D_i was deduced from previous lines by one of the following *R(CP*)-inference rules*:

(1) Let A, B be two disjunctions of linear inequalities and let L_1, L_2 be two linear inequalities.[22] From $A \vee L_1$ and $B \vee L_2$ derive $A \vee B \vee (L_1 + L_2)$.
(2) Let L be some linear equation.
From a disjunction of linear equations A derive $A \vee L$.
(3) Let A be a disjunction of linear equations.
From $A \vee (0 \geq 1)$ derive A.
(4) Let c be a non-negative integer.
From $(\vec{a} \cdot \vec{x} \geq a_0) \vee A$ derive $(c\vec{a} \cdot \vec{x} \geq ca_0) \vee A$.
(5) Let A be a disjunction of linear inequalities, and let $c \geq 1$ be an integer.
From $(c\vec{a} \cdot \vec{x} \geq a_0) \vee A$ derive $(a \cdot \vec{x} \geq \lceil a_0/c \rceil) \vee A$.

An *R(CP*) refutation of* a collection of disjunctions of linear inequalities K is a proof of the empty disjunction from K. The *size* of a proof π in R(CP*) is the total size of all the disjunctions of linear inequalities in π, denoted $|\pi|$.

In order for R(lin) to simulate R(CP*) proofs, we need to fix the following translation scheme. Every linear inequality L of the form $\vec{a} \cdot \vec{x} \geq a_0$ is translated into the following disjunction,

[21] When we allow coefficients to be written in *binary representation*, instead of unary representation, the resulting proof system is denoted R(CP).

[22] In all R(CP*)-inference rules, A, B are possibly the empty disjunctions.

denoted $\widehat{L}$:

$$(\vec{a}\cdot\vec{x}=a_0)\vee(\vec{a}\cdot\vec{x}=a_0+1)\vee\cdots\vee(\vec{a}\cdot\vec{x}=a_0+k)\ , \tag{51}$$

where k is such that a_0+k equals the sum of all positive coefficients in $\vec{a}$, that is, $a_0+k=\max_{\vec{x}\in\{0,1\}^n}(\vec{a}\cdot\vec{x})$ (in case the sum of all positive coefficients in $\vec{a}$ is less than a_0, then we put $k=0$). An inequality with no variables of the form $0\geq a_0$ is translated into $0=a_0$ in case it is false (that is, in case $0<a_0$), and into $0=0$ in case it is true (that is, in case $0\geq a_0$). Note that since the coefficients of linear inequalities (and linear equations) are written in *unary* representation, any linear inequality of size s translates into a disjunction of linear equations of size $O(s^2)$. Clearly, every $0,1$ assignment to the variables $\vec{x}$ satisfies L if and only if it satisfies its translation $\widehat{L}$. A disjunction of linear inequalities D is translated into the disjunction of the translations of all the linear inequalities in it, denoted $\widehat{D}$. A collection $K:=\{K_1,\ldots,K_m\}$ of disjunctions of linear inequalities, is translated into the collection $\left\{\widehat{K}_1,\ldots,\widehat{K}_m\right\}$.

Theorem 38. *R(lin) polynomially-simulates R(CP*). In other words, if π is an R(CP*) proof of a linear inequality D from a collection of disjunctions of linear inequalities $K_1,\ldots,K_t$, then there is an R(lin) proof of $\widehat{D}$ from $\widehat{K}_1,\ldots,\widehat{K}_t$ whose size is polynomial in $|\pi|$.*

Proof: By induction on the number of proof-lines in π.

Base case: Here we only need to show that the axioms of R(CP*) translates into axioms of R(lin), or can be derived with polynomial-size (in the size of the original R(CP*) axiom) R(lin) derivations (from R(lin)'s axioms).

R(CP*) axiom number (1): $x_i\geq 0$ translates into the R(lin) axiom $(x_i=0)\vee(x_i=1)$.

R(CP*) axiom number (2): $-x_i\geq -1$, translates into $(-x_i=-1)\vee(-x_i=0)$. From the Boolean axiom $(x_i=1)\vee(x_i=0)$ of R(lin), one can derive with a constant-size R(lin) proof the line $(-x_i=-1)\vee(-x_i=0)$ (for instance, by subtracting twice each equation in $(x_i=1)\vee(x_i=0)$ from itself).

R(CP*) axiom number (3): $(\vec{a}\cdot\vec{x}\geq a_0)\vee(-\vec{a}\cdot\vec{x}\geq 1-a_0)$. The inequality $(\vec{a}\cdot\vec{x}\geq a_0)$ translates into

$$\bigvee_{b=a_0}^{h}(\vec{a}\cdot\vec{x}=b)\,,$$

where h is the maximal value of $\vec{a}\cdot\vec{x}$ over $0,1$ assignments to $\vec{x}$ (that is, h is just the sum of all positive coefficients in $\vec{a}$). The inequality $(-\vec{a}\cdot\vec{x}\geq 1-a_0)$ translates into

$$\bigvee_{b=1-a_0}^{f}(-\vec{a}\cdot\vec{x}=b)\,,$$

where f is the maximal value of $-\vec{a}\cdot\vec{x}$ over $0,1$ assignments to $\vec{x}$ (that is, f is just the sum of all negative coefficients in $\vec{a}$). Note that one can always flip the sign of any equation $\vec{a}\cdot\vec{x}=b$ in R(lin). This is done, for instance, by subtracting twice $\vec{a}\cdot\vec{x}=b$ from itself. So overall R(CP*) axiom number (3) translates into

$$\bigvee_{b=a_0}^{h}(\vec{a}\cdot\vec{x}=b)\vee\bigvee_{b=1-a_0}^{f}(-\vec{a}\cdot\vec{x}=b)\,,$$

that can be converted inside R(lin) into

$$\bigvee_{b=-f}^{a_0-1}(\vec{a}\cdot\vec{x}=b)\vee\bigvee_{b=a_0}^{h}(\vec{a}\cdot\vec{x}=b)\,. \tag{52}$$

Let $\mathcal{A}' := \{-f, -f+1, \ldots, a_0 - 1, a_0, a_0+1, \ldots, h\}$ and let $\mathcal{A}$ be the set of all possible values that $\vec{a}\cdot\vec{x}$ can get over all possible Boolean assignments to $\vec{x}$. Notice that $\mathcal{A} \subseteq \mathcal{A}'$. By Lemma 8, for any $\vec{a}\cdot\vec{x}$, there is a polynomial-size (in the size of the linear form $\vec{a}\cdot\vec{x}$) derivation of $\bigvee_{\alpha\in\mathcal{A}}(\vec{a}\cdot\vec{x}=\alpha)$. By using the R(lin) Weakening rule we can then derive $\bigvee_{\alpha\in\mathcal{A}'}(\vec{a}\cdot\vec{x}=\alpha)$ which is equal to (52).

Induction step: Here we simply need to show how to polynomially simulate inside R(lin) every inference rule application of R(CP*).

Rule (1): Let A, B be two disjunctions of linear inequalities and let L_1, L_2 be two linear inequalities. Assume we already have a R(lin) proofs of $\widehat{A}\vee\widehat{L}_1$ and $\widehat{B}\vee\widehat{L}_2$. We need to derive $\widehat{A}\vee\widehat{B}\vee\widehat{L_1+L_2}$. Corollary 7 shows that there is a polynomial-size (in the size of $\widehat{L}_1$ and $\widehat{L}_2$; which is polynomial in the size of L_1 and L_2) derivation of $\widehat{L_1+L_2}$ from $\widehat{L}_1$ and $\widehat{L}_2$, from which the desired derivation immediately follows.

Rule (2): The simulation of this rule in R(lin) is done using the R(lin) Weakening rule.

Rule (3): The simulation of this rule in R(lin) is done using the R(lin) Simplification rule (remember that $0\geq 1$ translates into $0=1$ under our translation scheme).

Rule (4): Let c be a non-negative integer. We need to derive $(c\vec{a}\cdot\widehat{\vec{x}\geq ca_0})\vee\widehat{A}$ from $(\vec{a}\cdot\widehat{\vec{x}\geq a_0})\vee\widehat{A}$ in R(lin). This amounts only to "adding together" c times the disjunction $(\vec{a}\cdot\widehat{\vec{x}\geq a_0})$ in $(\vec{a}\cdot\widehat{\vec{x}\geq a_0})\vee\widehat{A}$. This can be achieved by c many applications of Corollary 7. We omit the details.

Rule (5): We need to derive $(\vec{a}\cdot\widehat{\vec{x}\geq\lceil a_0/c\rceil})\vee\widehat{A}$, from $(c\vec{a}\cdot\widehat{\vec{x}\geq a_0})\vee\widehat{A}$. Consider the disjunction of linear equations $(c\vec{a}\cdot\widehat{\vec{x}\geq a_0})$, which can be written as:

$$(c\vec{a}\cdot\vec{x}=a_0)\vee(c\vec{a}\cdot\vec{x}=a_0+1)\vee\ldots\vee(c\vec{a}\cdot\vec{x}=a_0+r)\,, \tag{53}$$

where a_0+r is the maximal value $c\vec{a}\cdot\vec{x}$ can get over $0,1$ assignments to $\vec{x}$. By Lemma 8 there is a polynomial-size (in the size of $\vec{a}\cdot\vec{x}$) R(lin) proof of

$$\bigvee_{\alpha\in\mathcal{A}}(\vec{a}\cdot\vec{x}=\alpha)\,, \tag{54}$$

where $\mathcal{A}$ is the set of all possible values of $\vec{a}\cdot\vec{x}$ over $0,1$ assignments to $\vec{x}$.

We now use (53) to cut-off from (54) all equations $(\vec{a}\cdot\vec{x}=\beta)$ for all $\beta<\lceil a_0/c\rceil$ (this will give us the desired disjunction of linear equations). Consider the equation $(\vec{a}\cdot\vec{x}=\beta)$ in (54) for some fixed $\beta<\lceil a_0/c\rceil$. Use the resolution rule of R(lin) to add this equation to itself c times inside (54). We thus obtain

$$(c\vec{a}\cdot\vec{x}=c\beta)\vee\bigvee_{\alpha\in\mathcal{A}\setminus\{\beta\}}(\vec{a}\cdot\vec{x}=\alpha)\,. \tag{55}$$

Since β is an integer and $\beta<\lceil a_0/c\rceil$, we have $c\beta<a_0$. Thus, the equation $(c\vec{a}\cdot\vec{x}=c\beta)$ does not appear in (53). We can then successively resolve $(c\vec{a}\cdot\vec{x}=c\beta)$ in (55) with each equation $(c\vec{a}\cdot\vec{x}=a_0),\ldots,(c\vec{a}\cdot\vec{x}=a_0+r)$ in (53). Hence, we arrive at $\bigvee_{\alpha\in\mathcal{A}\setminus\{\beta\}}(\vec{a}\cdot\vec{x}=\alpha)$. Overall, we can cut-off all equations $(\vec{a}\cdot\vec{x}=\beta)$, for $\beta<\lceil a_0/c\rceil$, from (54). We then get the disjunction

$$\bigvee_{\alpha\in\mathcal{A}'}(\vec{a}\cdot\vec{x}=\alpha)\,,$$

where $\mathcal{A}'$ is the set of all elements of $\mathcal{A}$ greater or equal to $\lceil a_0/c\rceil$ (in other words, all values greater or equal to $\lceil a_0/c\rceil$ that $\vec{a}\cdot\vec{x}$ can get over $0,1$ assignments to $\vec{x}$). Using the Weakening rule of R(lin) (if necessary) we can arrive finally at the desired disjunction $(\vec{a}\cdot\widehat{\vec{x}\geq\lceil a_0/c\rceil})$, which concludes the R(lin) simulation of R(CP*)'s inference Rule (5). ∎

Appendix A. Feasible Monotone Interpolation

Here we formally define the feasible monotone interpolation property. The definition is taken mainly from [Kra97]. Recall that for two binary strings of length n (or equivalently, Boolean assignments for n propositional variables) α, α', we denote by $\alpha' \geq \alpha$ that α' is *bitwise* greater than α, that is, that for all $i \in [n]$, $\alpha'_i \geq \alpha_i$ (where α'_i and α_i are the ith bits of α' and α, respectively). Let $A(\vec{p}, \vec{q}), B(\vec{p}, \vec{r})$ be two collections of formulas in the displayed variables only, where $\vec{p}, \vec{q}, \vec{r}$ are pairwise disjoint sequences of distinct variables (similar to the notation at the beginning of Section 7). Assume that there is no assignment that satisfies both $A(\vec{p}, \vec{q})$ and $B(\vec{p}, \vec{r})$. We say that $A(\vec{p}, \vec{q}), B(\vec{p}, \vec{r})$ are *monotone* if one of the following conditions hold:

(1) If $\vec{\alpha}$ is an assignment to $\vec{p}$ and $\vec{\beta}$ is an assignment to $\vec{q}$ such that $A(\vec{\alpha}, \vec{\beta}) = 1$, then for any assignment $\vec{\alpha}' \geq \vec{\alpha}$ it holds that $A(\vec{\alpha}', \vec{\beta}) = 1$.
(2) If $\vec{\alpha}$ is an assignment to $\vec{p}$ and $\vec{\beta}$ is an assignment to $\vec{r}$ such that $B(\vec{\alpha}, \vec{\beta}) = 1$, then for any assignment $\vec{\alpha}' \leq \vec{\alpha}$ it holds that $B(\vec{\alpha}', \vec{\beta}) = 1$.

Fix a certain proof system $\mathcal{P}$. Recall the definition of the interpolant function (corresponding to a given unsatisfiable $A(\vec{p}, \vec{q}) \wedge B(\vec{p}, \vec{r})$; that is, functions for which (39) in Section 7 hold). Assume that for every monotone $A(\vec{p}, \vec{q}), B(\vec{p}, \vec{r})$ there is a transformation from every $\mathcal{P}$-refutation of $A(\vec{p}, \vec{q}) \wedge B(\vec{p}, \vec{r})$ into the corresponding interpolant *monotone* Boolean circuit $C(\vec{p})$ (that is, $C(\vec{p})$ uses only monotone gates[23]) and whose size is polynomial in the size of the refutation (note that for every monotone $A(\vec{p}, \vec{q}), B(\vec{p}, \vec{r})$ the corresponding interpolant circuit must compute a monotone function;[24] the interpolant circuit itself, however, might not be monotone, namely, it may use non-monotone gates). In such a case, we say that $\mathcal{P}$ has the *feasible monotone interpolation property*. This means that, if a proof system $\mathcal{P}$ has the feasible monotone interpolation property, then an exponential lower bound on monotone circuits that compute the interpolant function corresponding to $A(\vec{p}, \vec{q}) \wedge B(\vec{p}, \vec{r})$ implies an exponential-size lower bound on $\mathcal{P}$-refutations of $A(\vec{p}, \vec{q}) \wedge B(\vec{p}, \vec{r})$.

Definition A.1 (**Feasible monotone interpolation property**)**.** Let $\mathcal{P}$ be a propositional refutation system. Let $A_1(\vec{p}, \vec{q}), \ldots, A_k(\vec{p}, \vec{q})$ and $B_1(\vec{p}, \vec{r}), \ldots, B_\ell(\vec{p}, \vec{r})$ be two collections of formulas with the displayed variables only (where $\vec{p}$ has n variables, $\vec{q}$ has s variables and $\vec{r}$ has t variables), such that *either* (the set of satisfying assignments of) $A_1(\vec{p}, \vec{q}), \ldots, A_k(\vec{p}, \vec{q})$ meet condition 1 above *or* (the set of satisfying assignments of) $B_1(\vec{p}, \vec{r}), \ldots, B_\ell(\vec{p}, \vec{r})$ meet condition 2 above. Assume that for any such $A_1(\vec{p}, \vec{q}), \ldots, A_k(\vec{p}, \vec{q})$ and $B_1(\vec{p}, \vec{r}), \ldots, B_\ell(\vec{p}, \vec{r})$, if there exists a $\mathcal{P}$-refutation for $A_1(\vec{p}, \vec{q}) \wedge \cdots \wedge A_k(\vec{p}, \vec{q}) \wedge B_1(\vec{p}, \vec{r}) \wedge \ldots \wedge B_\ell(\vec{p}, \vec{r})$ of size S then there exists a monotone Boolean circuit separating $\mathcal{U}_A$ from $\mathcal{V}_B$ (as defined in Section 7.1) of size polynomial in S. In this case we say that $\mathcal{P}$ possesses the *feasible monotone interpolation property*.

Acknowledgments

We wish to thank Arist Kojevnikov for useful correspondence on his paper. This work was carried out in partial fulfillment of the requirements for the Ph.D. degree of the second author.

[23] For instance, a *monotone Boolean circuit* is a circuit that uses only $\wedge, \vee$ gates of fan-in two (see also Section 8). In certain cases, the monotone interpolation technique is also applicable for a larger class of circuits, that is, circuits that compute with real numbers and that can use any nondecreasing real functions as gates (this was proved by Pudlák in [Pud97]).

[24] That is, if $\alpha' \geq \alpha$ then $C(\alpha') \geq C(\alpha)$.

References

[AB87] Noga Alon and Ravi B. Boppana. The monotone circuit complexity of boolean functions. *Combinatorica*, 7(1):1–22, 1987. 8, 26

[ABE02] Albert Atserias, Maria L. Bonet, and Juan L. Esteban. Lower bounds for the weak pigeonhole principle and random formulas beyond resolution. *Information and Computation*, 176:152–136, August 2002. 1.2, 3, 6.3, 6.3, 20, 3

[ABSRW02] Michael Alekhnovich, Eli Ben-Sasson, Alexander A. Razborov, and Avi Wigderson. Space complexity in propositional calculus. *SIAM J. Comput.*, 31(4):1184–1211 (electronic), 2002. 9.1.2

[And85] A. E. Andreev. On a method for obtaining lower bounds for the complexity of individual monotone functions. *Dokl. Akad. Nauk SSSR (in Russian)*, 282(5):1033–1037, 1985. [Engl. Transl. Soviet Math. Dokl., vol. 31 (1985), pp. 530-534]. 8

[BGIP01] Samuel Buss, Dima Grigoriev, Russell Impagliazzo, and Toniann Pitassi. Linear gaps between degrees for the polynomial calculus modulo distinct primes. *J. Comput. System Sci.*, 62(2):267–289, 2001. Special issue on the 14th Annual IEEE Conference on Computational Complexity (Atlanta, GA, 1999). 6.2

[BP97] Samuel Buss and Toniann Pitassi. Resolution and the weak pigeonhole principle. In *Computer science logic (Aarhus, 1997)*, volume 1414 of *Lecture Notes in Comput. Sci.*, pages 149–156. Springer, Berlin, 1997. 3

[BPR97] Maria Bonet, Toniann Pitassi, and Ran Raz. Lower bounds for cutting planes proofs with small coefficients. *The Journal of Symbolic Logic*, 62(3):708–728, 1997. 1.1, 1.2, 1.2, 6.3, 2, 7.1.1, 8, 27

[CR79] Stephen A. Cook and Robert A. Reckhow. The relative efficiency of propositional proof systems. *The Journal of Symbolic Logic*, 44(1):36–50, 1979. 2

[Hak85] Armin Haken. The intractability of resolution. *Theoret. Comput. Sci.*, 39(2-3):297–308, 1985. 1

[HIK+05] Edward Hirsch, Dmitry Itsykson, Arist Kojevnikov, Alexander Kulikov, and Sergey Nikolenko. Report on the mixed boolean-algebraic solver. Technical report, Laboratory of Mathematical Logic of St. Petersburg Department of Steklov Institute of Mathematics, November 2005. url: `http://logic.pdmi.ras.ru/~basolver/basolver-firstreport.pdf` 1.1

[HK06] Edward Hirsch and Arist Kojevnikov. Several notes on the power of Gomory-Chvátal cuts. *Annals of Pure and Applied Logic*, 141:429–436, 2006. 1.1

[IPU94] Russel Impagliazzo, Toniann Pitassi, and Alasdair Urquhart. Upper and lower bounds for tree-like cutting planes proofs. In *Ninth Annual Symposium on Logic in Computer Science*, pages 220–228. IEEE Comput. Soc. Press, 1994. 7.1.1

[Koj07] Arist Kojevnikov. Improved lower bounds for tree-like resolution over linear inequalities. In *Proceedings of the 10th International Conference on Theory and Applications of Satisfiability Testing (SAT), 2007*. Preliminary version in *Electronic Colloquium on Computational Complexity, ECCC*, January 2007. Report No. TR07-010. 1.1, 10

[Kra94] Jan Krajíček. Lower bounds to the size of constant-depth propositional proofs. *The Journal of Symbolic Logic*, 59(1):73–86, 1994. 7

[Kra97] Jan Krajíček. Interpolation theorems, lower bounds for proof systems, and independence results for bounded arithmetic. *The Journal of Symbolic Logic*, 62(2):457–486, 1997. 1.1, 1.2, 1.2, 6.3, 7.1, 7.1.1, 23, A

[Kra98] Jan Krajíček. Discretely ordered modules as a first-order extension of the cutting planes proof system. *The Journal of Symbolic Logic*, 63(4):1582–1596, 1998. 1.1, 1.2, 6.3, 10

[Kra01] Jan Krajíček. On the weak pigeonhole principle. *Fund. Math.*, 170(1-2):123–140, 2001. Dedicated to the memory of Jerzy Łoś. 6.3

[Kra07] Jan Krajíček. An exponential lower bound for a constraint propagation proof system based on ordered binary decision diagrams. To appear in *The Journal of Symbolic Logic*. Preliminary version available in *Electronic Colloquium on Computational Complexity, ECCC*, January 2007. Report No. TR07-007. 6.3

[KW88] Mauricio Karchmer and Avi Wigderson. Monotone circuits for connectivity require super-logarithmic depth. In *Proceedings of the 20th Annual ACM Symposium on Theory of Computing*, pages 539–550. ACM, 1988. 7.1.1, 7.2

[Pud97] Pavel Pudlák. Lower bounds for resolution and cutting plane proofs and monotone computations. *The Journal of Symbolic Logic*, 62(3):981–998, Sept. 1997. 6.3, 23

[Razb85] Alexander A. Razborov. Lower bounds on the monotone complexity of some Boolean functions. *Dokl. Akad. Nauk SSSR (in Russian)*, 281(4):798–801, 1985. [English translation in Sov. Math. Dokl., vol . 31 (1985), pp. 354-357.]. 8

[Razb95] Alexander A. Razborov. Unprovability of lower bounds on circuit size in certain fragments of bounded arithmetic. *Izv. Ross. Akad. Nauk Ser. Mat.*, 59(1):201–224, 1995. 7.1.1

[Razb02] Alexander A. Razborov. Proof complexity of pigeonhole principles. In *Developments in language theory (Vienna, 2001)*, volume 2295 of *Lecture Notes in Comput. Sci.*, pages 110–116. Springer, Berlin, 2002. 1

[Raz04] Ran Raz. 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*, pages 633–641, Chicago, IL, 2004. ACM. 1

[Raz06] Ran Raz. Separation of multilinear circuit and formula size. *Theory of Computing, Vol. 2, article 6*, 2006. 1

[RT06] Ran Raz and Iddo Tzameret. The strength of multilinear proofs. *Comput. Complexity (to appear)*. Preliminary version in *Electronic Colloquium on Computational Complexity, ECCC*, January 2006. Report No. TR06-001. (document), 1, 1.1, 1.2, 1.2, 1, 2, 6.2, 9, 9.1.3, 4, 9.3, 33, 9.3.1, 4, 9.4, 9.4

[SW01] Amir Shpilka and Avi Wigderson. Depth-3 arithmetic circuits over fields of characteristic zero. *Comput. Complexity*, 10:1–27, 2001. 9.1.1

[Tse68] G. C. Tseitin. *On the complexity of derivations in propositional calculus*. Studies in constructive mathematics and mathematical logic Part II. Consultants Bureau, New-York-London, 1968. 6.2

Department of Applied Mathematics and Computer Science, Weizmann Institute, Rehovot 76100, Israel

E-mail address: ranraz@wisdom.weizmann.ac.il

School of Computer Science, Tel Aviv University, Tel Aviv 69978, Israel

E-mail address: tzameret@tau.ac.il