Propositional Proof Systems: Efficiency and Automatizability

by

Mikhail Alekhnovitch

B.S., Mathematics, Moscow State University (2000)
M.S., Mathematics, Moscow State University (2000)

Submitted to the Department of Mathematics
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

June 2003

Author .....
Department of Mathematics
April 14, 2003

Certified by .....
Madhu Sudan
Professor of Computer Science
Department of Electrical Engineering and Computer Science
Thesis Supervisor

Accepted by .....
Rodolfo Ruben Rosales
Chairman, Applied Mathematics Committee

Accepted by ..... Pavel Etingof
Chairman, Department Committee on Graduate Students

Propositional Proof Systems: Efficiency and Automatizability

by

Mikhail Alekhnovitch

Submitted to the Department of Mathematics
on April 14, 2003, in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy

Abstract

The thesis considers two fundamental questions in propositional proof complexity: lower bounds on the size of the shortest proof and automatizability of propositional proof systems.

With respect to the first part, we develop a new paradigm for proving lower bounds in propositional calculus. Our method is based on the purely computational concept of pseudorandom generator. Namely, we call a pseudorandom generator $G_n : \{0,1\}^n \to \{0,1\}^m$ *hard* for a propositional proof system $\mathcal{P}$ if $\mathcal{P}$ cannot efficiently prove the (properly encoded) statement $G_n(x_1, \ldots, x_n) \neq b$ for *any* string $b \in \{0,1\}^m$. We consider a variety of "combinatorial" pseudorandom generators inspired by the Nisan-Wigderson generator on the one hand, and by the construction of Tseitin tautologies on the other. We prove that under certain circumstances these generators are hard for such proof systems as Resolution, Polynomial Calculus and Polynomial Calculus with Resolution (PCR).

As to the second part, we prove that the problem of approximating the size of the shortest proof within factor $2^{\log^{1-o(1)} n}$ is NP-hard. This result is very robust in that it holds for almost all natural proof systems, including: Frege systems, extended Frege systems, resolution, Horn resolution, the sequent calculus, the cut-free sequent calculus, as well as the polynomial calculus. We introduce the Monotone Minimum (Circuit) Satisfying Assignment problem and reduce it to the problem of approximating the length of proofs.

Finally, we show that neither Resolution nor tree-like Resolution is automatizable unless the class **W[P]** from the hierarchy of parameterized problems is fixed-parameter tractable by randomized algorithms with one-sided error.

Thesis Supervisor: Madhu Sudan
Title: Professor of Computer Science,
Department of Electrical Engineering and Computer Science

Acknowledgments

I am very grateful to Madhu Sudan who is my supervisor for many helpful discussions. I am indebted to Alexander Razborov as well as to the whole list of collaborators which also includes Eli Ben-Sasson, Sam Buss, Shlomo Moran, Toniann Pitassi and Avi Wigderson for many interesting ideas and insights that lead to the presented results.

I would like to thank E. Ben-Sasson for his useful remarks on this text.

Contents

List of Figures

Chapter 1

Introduction

A propositional proof system as formalized by Cook and Reckhow [CR79], is an algorithm $\mathcal{A}$ that can test in polynomial time whether an input string is a valid "proof" of a universally true DNF τ (also called *tautology*). The proof here is any kind of certificate that allows $\mathcal{A}$ to conclude that τ is indeed a tautology. It is required that the propositional system be *sound* and *complete*, which means that only tautological formulas can have proofs and there always exists at least one proof for every tautology.

One can imagine a proof system as a non-deterministic heuristic for a **coNP**-complete language. A proof in this model is a non-deterministic witness that helps to test the membership in this language. It was shown in [CR79] that $\mathbf{NP} = \mathbf{coNP}$ if and only if there exists a propositional proof system in which every tautology has a proof of polynomial size.

The most fundamental question in this framework is "How large can be the shortest proof of certain interesting tautologies for various proof systems?". Since it is believed that $\mathbf{NP} \neq \mathbf{coNP}$ for any proof system there should exist a class of hard tautologies, that require superpolynomial size proofs. This naturally suggests the following goal: prove lower bounds on the proof size for as strong a propositional proof system as possible. These lower bounds are important from several aspects. Besides the natural curiosity, it is a step toward the separation of **NP** and **coNP**. Proving that $\mathbf{NP} \neq \mathbf{coNP}$ may be even harder than proving $\mathbf{P} \neq \mathbf{NP}$, however one

may assume the latter hypothesis and still try to prove the former. This approach leads to so-called *conditional* lower bounds in propositional complexity: assuming a certain problem computationally hard construct a family of tautologies, which require superpolynomial proofs in certain proof systems. There are several known examples of such results (see, for example, [Kra97] for the lower bounds based on the efficient interpolation property).

The second reason for lower bounds in propositional comlexity is that they also limit the performance of a wide class of algorithmic heuristics (i.e. Automated Theorem Provers) that are implicitly or explicitly based on the corresponding propositional proof system. Thus, a lower bound on the shortest proof size in the appropriate system would yield a lower bound on the running time of the whole class of such heuristics. Finally, lower bounds for strong systems imply independence results in certain fragments of bounded arithmetic, an interesting problem in its own right.

While there has been a great progress during recent years in proving lower bounds for weak proof systems, still no (even conditional) lower bounds are known for strong systems such as Frege or Extended Frege. The situation is somewhat similar to computational complexity, where no lower bounds are known for unrestricted circuits. In fact some researchers believe that this is not a coincidence and propositional systems are inherently dependent on the corresponding computational model, thus requiring circuit lower bounds for this model to be proved first. Also, only few known tautologies are candidates for being hard for Frege or Extended Frege.

In the first part of the thesis we propose a new paradigm for proving lower bounds for propositional proof systems, for this we use a concept from the computational complexity. Namely, we say that a function $G : \{0,1\}^n \to \{0,1\}^m$ is a good *pseudo-random* generator w.r.t. proof system $\mathcal{P}$ iff for every $y \in \{0,1\}^m$ any $\mathcal{P}$-proof of the fact $y \notin \mathrm{im}(G)$ requires superpolynomial size. In otherwords, a generator G is hard for $\mathcal{P}$ if and only if it cannot efficiently prove that any explicit element does not lie in the image of G.

Inspired by the natural proofs approach ([RR97]) on one hand and by the efficient

interpolation property on the other this definition naturally suggests one more link between computational and propositional complexity. Our conjecture is that in some cases classical pseudorandom generators from computational complexity may be also hard for propositional proof systems as strong as Frege or Extended Frege. This gives a new class of potentially hard tautologies, the plausible hardness of which is based upon the following observation: it is usually tricky (in the ordinary mathematical world!) to find an explicit vector that does not belong to the image of pseudo-random generator.

Our concrete results toward this direction are however by far more modest. The main object of our consideration is Nissan-Wigderson generators, which are one of the main tools in derandomization techniques. We show that if the underlying function satisfies the certain (wide) hardness condition then Nissan-Wigderson generators based on this function are hard for Resolution and Polynomial Calculus proof systems. In case of Polynomial Calculus, we develop a new technique for proving lower bounds in non-binomial case (e.g. when the input axioms are not binomials) based on the pseudorandom method, the latter techniques can be interesting by itself.

In the second part we address the following fundamental question about propositional proof systems, which is in a sense "orthogonal" to the efficiency of the system. Assume that τ is a tautology that requires a proof of size S in the propositional system $\mathcal{P}$. In what time a $\mathcal{P}$-proof of τ can be constructed by some algorithm $\mathcal{A}$? In otherwords, can we find a short proof efficiently if we know that such a proof exists?

Our first result in this direction states that it is **NP**-hard to produce a proof which is within a factor of $2^{\log^{1-o(1)} n}$ close to the optimal. The weakness of this bound is compensated by that it is applicable to every (sufficiently natural) proof system whatsoever.

In general, the complexity of the proof search was captured by the notion of *automatizability* suggested in [BPR00]. The proof system $\mathcal{P}$ is said to be automatizable, if there exists an algorithm that for any tautology τ finds a $\mathcal{P}$-proof of τ in time polynomial in the size of the shortest proof of τ. This notion plays an important role

in Automated Theorem Proving as it allows to reduce (at least on the theoretical level) the problem of the algorithmic proof search to the question of pure existence of the short proof.

It was shown in a sequence of works [KP95], [BPR00], [BDG+99] that sufficiently strong systems (which includes Frege, Extended Frege, $\mathbf{TC}^0$-Frege and $\mathbf{AC}^0$-Frege) are not automatizable modulo some cryptographic assumptions. However their techniques are not applicable to weak systems, of which Resolution and Tree-like Resolution are the simplest. Since Tree-like Resolution is known to be quasi-automatizable (which means that the proof can be found in quasi-polynomial time) it was a plausible candidate for being automatizable. Our second result refutes this conjecture by showing that neither Resolution nor Tree-like Resolution is automatizable unless the parameterized complexity hierarchy is tractable. As a by product we also establish a hardness of approximation of a **NP**-complete problem in the framework of parameterized complexity without using the PCP theorem, which might be of independent interest.

1.1 Structure of the Thesis

Chapter 2 contains some basic notation and the definitions of propositional proof systems used throughout the thesis. Chapters 3 through 6 are formally independent and can be read in any order, however we strongly advise to read Chapter 4 after Chapter 3 and Chapter 6 after Chapter 5. Below we give a brief summary of these chapters along with the references on the published papers that our results are based on.

- **Chapter 3** ([ABSRW00], joint with E. Ben-Sasson, A. Razborov and A. Wigderson) contains the definition of pseudorandom generators for propositional proofs systems. It proves several lower bounds on the hardness of generator tautologies for Resolution and Polynomial calculus (the latter proved only in the case when the underlying function is the XOR function).

- **Chapter 4** ([AR01b], joint with A. Razborov)
  develops the algebraic machinery to show the hardness of generator tautologies for Polynomial Calculus for a wide class of underlying functions. It constructs many new examples of tautologies hard for Polynomial Calculus and a simplified lower bound on the certain version of pigeon-hole principle.

  We are grateful to Jan Krajíček for his suggestion to consider Pigeonhole principle in the framework of this chapter.

- **Chapter 5** ([ABMP01], joint with S. Buss, S. Moran and T. Pitassi)
  proves that it is **NP**-hard to find a proof of size within $2^{\log^{1-o(1)} n}$ close to the optimal for several propositional proof systems.

  We would like to thank S. Arora for pointing out that Minimum Label Cover can be reduced to Monotone Minimum Satisfying Assignment introduced in this chapter.

- **Chapter 6** ([AR01a], joint with A. Razborov)
  shows that neither Resolution nor Tree-like Resolution are automatizable unless the parameterized complexity hierarchy is tractable by randomized algorithms.

We conclude the thesis by a brief survey on the most recent developments in Chapter 7.

Chapter 2

Preliminaries

2.1 Basic Notation

Let x be a Boolean variable, i.e. a variable that ranges over the set $\{0,1\}$. A *literal* of x is either x (denoted sometimes as x^1) or $\bar{x}$ (denoted sometimes as x^0). A *clause* is a disjunction of literals.

For any Boolean function $f : \{0,1\}^n \to \{0,1\}$, $Vars(f)$ will denote the set of its essential variables. An *assignment to* f is a mapping $\alpha : Vars(f) \to \{0,1\}$. A *restriction of* f is a mapping $\rho : Vars(f) \to \{0,1,\star\}$. We denote by $|\rho|$ the number of assigned variables, $|\rho| \stackrel{\text{def}}{=} |\rho^{-1}(\{0,1\})|$.

The *restriction of* f *by* ρ, denoted $f|_\rho$, is the Boolean function obtained from f by setting the value of each $x \in \rho^{-1}(\{0,1\})$ to $\rho(x)$, and leaving each $x \in \rho^{-1}(\star)$ as a variable.

We say that an assignment α *satisfies* f if $f(\alpha) = 1$. For Boolean functions $f_1, \ldots, f_k, g$ we say that $f_1, \ldots, f_k$ *semantically imply* g (denoted $f_1, \ldots, f_k \models g$), if every assignment to $V \stackrel{\text{def}}{=} Vars(f_1) \cup \ldots \cup Vars(f_k) \cup Vars(g)$ satisfying $f_1, \ldots, f_k$, satisfies g as well (i.e. $\forall \alpha \in \{0,1\}^V (f_1(\alpha) = \cdots = f_k(\alpha) = 1 \Rightarrow g(\alpha) = 1)$).

For n, a non-negative integer let $[n] \stackrel{\text{def}}{=} \{1, 2, \ldots, n\}$.

2.2 Propositional proof systems

By a formal definition of [CR79], a propositional proof system $\mathcal{P}$ is a polynomially computable function

$$\mathcal{P} : \{0,1\}^n \stackrel{onto}{\to} \text{TAUT}$$

from the set of binary strings onto the set of all tautologies (e.g. universally true formulas) written in some fixed encoding. A string w is called the *proof* of a formula τ in $\mathcal{P}$ if and only if $\mathcal{P}(w) = \tau$. The most important function associated with $\mathcal{P}$ is the *minimal proof size* defined as

$$S_{\mathcal{P}}(\tau) = \min_{\mathcal{P}(w)=\tau} |w|.$$

In some propositional systems instead of proofs it is common to consider *refutations* (e.g. proofs of contradiction) of unsatisfiable CNF's τ. To the abuse of notation, such formulas are also called tautologies. Below we give the defintions of propositional proof systems used throughout the thesis.

2.2.1 Resolution

Resolution is the simplest and probably the most widely studied proof system. It operates with clauses and has one rule of inference called *resolution rule* that allows to infer a new clause from two other clauses:

$$\frac{A \vee x \qquad B \vee \bar{x}}{A \vee B}.$$

A *resolution refutation* of a CNF formula τ is a resolution proof of the empty clause from the clauses appearing in τ.

The *size* of a resolution proof is the number of different clauses in it. The *width* $w(C)$ *of a clause* C is the number of literals in C. The *width* $w(\tau)$ *of a set of clauses* τ (in particular, the width of a resolution proof) is the maximal width of a clause appearing in this set.

The story of propositional proof complexity began some 35 years ago when in the seminal paper [Tse68] Tseitin proved super-polynomial lower bounds on the size of any resolution refutation of (what was afterwards called) Tseitin tautologies under one extra regularity assumption on the structure of refutation. Haken [Hak85] was the first to remove this restriction and prove exponential lower bounds for general resolution (for the pigeonhole principle). Urquhart [Urq87] proved exponential lower bounds on the size of general resolution refutations for Tseitin tautologies.

Ben-Sasson and Wigderson [BW99], strengthening a result from [CEI96] (cf. Section 2.2.2 below) proved the following width-size relation:

Proposition 2.2.1 *Let τ be an unsatisfiable CNF in n variables that has a resolution refutation of size S. Then τ has a resolution refutation of width at most $w(\tau) + O(\sqrt{n \log S})$.*

[BW99] also established a linear lower bound on the width of resolution refutation for Tseitin tautologies. In combination with Proposition 2.2.1 this gave an alternate (and much simpler) proof of the size lower bound from [Urq87].

2.2.2 Polynomial Calculus and Polynomial Calculus with Resolution

Polynomial Calculus, introduced by Clegg, Edmonds and Impagliazzo in [CEI96] is a proof system that models common algebraic reasoning. Despite its algebraic nature, Polynomial Calculus (PC) turned out extremely useful for studying "pure" propositional proof systems.

PC operates with polynomials $P \in F[x_1, \ldots, x_n]$ for some fixed field F; a polynomial P is interpreted as, and often identified with, the polynomial equation $P = 0$. Polynomial Calculus has polynomials $x_i^2 - x_i$ ($i \in [n]$) as *default axioms* and has two inference rules:

$$\frac{P_1 \qquad P_2}{\alpha P_1 + \beta P_2};\ \alpha, \beta \in F \quad \text{(Scalar Addition)}$$

and

$$\frac{P}{x \cdot P} \quad \text{(Variable Multiplication)}.$$

A *polynomial calculus refutation* of a set of polynomials Γ is a polynomial calculus proof of 1 from Γ. The *degree of a PC proof* is the maximal degree of a polynomial appearing in it. The *size of a PC proof* is the total number of monomials in the proof.

First non-trivial lower bounds on the degree of PC refutations were proved by Razborov [Raz98] (for the pigeonhole principle). Grigoriev [Gri98] proved linear lower bounds on the degree of Nullstellensatz refutations (which is a subsystem of Polynomial Calculus) for Tseitin tautologies. Finally, Buss, Grigoriev, Impagliazzo and Pitassi [BGIP01] extended the latter bound to arbitrary polynomial calculus proofs. Following [BGIP01] and the research whose outcome is presented in this thesis, Ben-Sasson and Impagliazzo [BI99] further simplified this argument, and derived linear degree lower bounds for random CNFs.

[CEI96] proved that small size resolution proofs can be simulated by low degree PC proofs (Proposition 2.2.1 is a later improvement of this result). [IPS99] observed that the same simulation works also for small size *polynomial calculus* proofs.

Motivated in part by this similarity, [ABRW02] proposed to consider the following natural system PCR extending both Polynomial Calculus and Resolution. PCR operates with polynomials $P \in F[x_1, \ldots, x_n, \bar{x}_1, \ldots, \bar{x}_n]$, where $\bar{x}_1, \ldots, \bar{x}_n$ are treated as new formal variables. PCR has all default axioms and inference rules of PC (including, of course, those that involve new variables $\bar{x}_i$), plus additional default axioms $x_i + \bar{x}_i = 1$ $(i \in [n])$. The *size* and *degree of a PCR proof* are defined in the same way as for Polynomial Calculus. It should be noted that there is not much sense in giving a separate definition for the degree of PCR proofs since the linear transformation $\bar{x}_i \mapsto 1 - x_i$ takes a PCR-proof to (essentially) PC-proof while preserving degree. This system, however, becomes extremely convenient when it is the number of clauses which matters (see [ABRW02]).

PCR is an extension of PC by definition. Also, PCR extends Resolution via the following translation. For a clause C, let C_+ $[(C_-)]$ be the set of positive [respectively,

negative] literals appearing in it. Then a CNF formula τ gets translated into the set of polynomials Γ_τ defined by $\Gamma_\tau \stackrel{\text{def}}{=} \left\{ \left(\prod_{\bar{x} \in C_-} x \cdot \prod_{x \in C_+} \bar{x} \right) \middle| \; C \in \tau \right\}$. Clearly, τ is satisfiable if and only if Γ_τ has a common root in F satisfying all default axioms

$$x_i^2 = x_i; \;\; \bar{x}_i^2 = \bar{x}_i; \;\; x_i + \bar{x}_i = 1. \tag{2.1}$$

Moreover, it is easy to see that every width w size S resolution refutation of τ can be transformed into a degree $(w+1)$ size $O(nS)$ PCR refutations of the associated set of polynomials Γ_τ (cf. [BG99, Section 5]). For ease of notation, we will omit the translation and define a *PCR refutation of a CNF* τ as a PCR refutation of Γ_τ. A *PC refutation* of τ is a PC refutation of the set of polynomials

$$\Gamma'_\tau \stackrel{\text{def}}{=} \left\{ \left(\prod_{\bar{x} \in C_-} x \cdot \prod_{x \in C_+} (1 - x) \right) \middle| \; C \in \tau \right\} \tag{2.2}$$

obtained from Γ_τ by the linear transformation $\bar{x}_i \mapsto 1 - x_i$.

In fact all our lower bounds for PC in Chapter 3 hold also for PCR so we will usually use the translation to PCR and prove PCR lower bounds which imply the hardness for PC.

[ABRW02] observed that the two simulations from [CEI96, IPS99] can be merged into one as follows:

Proposition 2.2.2 *Let Γ be a system of polynomials in the variables $x_1, \ldots, x_n$, $\bar{x}_1, \ldots, \bar{x}_n$ that have no common roots in F satisfying all default axioms* (2.1)*, and let $d(\Gamma) \stackrel{\text{def}}{=} \max\{\deg(P) \mid P \in \Gamma\}$. Then every size S PCR refutation of Γ can be transformed into another PCR refutation of Γ that has degree at most $d(\Gamma) + O(\sqrt{n \log S})$.*

2.2.3 Frege and Extended Frege

Frege proof systems are proof systems for propositional logic. A Frege proof system $\mathbb{F}$ is specified by its language L and a finite set of inference and axiom schemes. The language L is a finite set of Boolean connectives, which is complete in the sense that

any Boolean function can be represented by an L-formula. The permissible inferences are specified schematically as inferences

$$\frac{A_1 \quad A_2 \quad \cdots \quad A_k}{B}$$

which indicates that for any substitution σ of formulas for variables, $B\sigma$ may be inferred from the formulas $A_1\sigma, \ldots, A_k\sigma$. We allow $k = 0$ in the above scheme, which corresponds to axioms. Finally, the Frege proof system must be implicationally complete, i.e., if $A_1, \ldots, A_k \models B$ then there is a derivation of B from the assumptions $A_1, \ldots, A_k$ using the inferences of $\mathbb{F}$.

We define the size or length, $|C|$, of a formula C to equal the number of symbols in C, where each occurence of a variable or a connective is counted as a symbol. Likewise, if P is a Frege proof, then the symbol size of P, $|P|$, equals the number of symbols in P. If $\mathbb{F}$ is a Frege system, then $\mathbb{F} \vdash \varphi$ means that there is an $\mathbb{F}$-proof of φ. The *symbol size* of a Frege proof is the total number of symbols in the proof. The *step-length* (or length) of a Frege proof is the number of lines in the proof. $\mathbb{F} \stackrel{n}{\vdash} \varphi$ means that there exists an $\mathbb{F}$-proof P such that $|P| \leq n$.

Typical examples of Frege system include the 'textbook systems' which use the language $\{\wedge, \vee, \neg, \rightarrow\}$ and have a finite set of axiom schemes and have modus ponens as their only other rule of inference. Of course there are many possible such textbook systems since there are many choices for the axiom schemes; however, they are all essentially equivalent in terms of proof length. Indeed the following holds:

Theorem 2.2.3 ([CR79, Rec76, Sta77]) *If $\mathbb{F}_1$ and $\mathbb{F}_2$ are Frege systems with the same language, then they linearly simulate each other; i.e., for all φ, if $\mathbb{F}_1 \stackrel{n}{\vdash} \varphi$ then $\mathbb{F}_2 \stackrel{O(n)}{\vdash} \varphi$, and vice-versa.*

For Frege proof systems in differing languages, it is known that any two Frege systems $\mathbb{F}_1$ and $\mathbb{F}_2$ p-simulate each other, i.e., that any $\mathbb{F}_1$-proof can be translated into an $\mathbb{F}_2$-proof in polynomial time and vice-versa; see [CR79, Rec76] for precise definitions and proofs of this.

Extended Frege proof systems are propositional proof systems which allow the introduction of abbreviations of formulas on the fly. It is conjectured that the minimal symbol size for extended Frege proofs can sometimes be exponentially smaller than the corresponding minimal Frege proof; however, this is still open. It is known that the length of shortest extended Frege proofs is very closely linked (essentially linearly related to) the number of steps in Frege proofs [CR79, Rec76, Sta77].

Part I

Pseudorandom Generators in Propositional Proof Complexity

Chapter 3

Pseudorandom Generators for Resolution and PCR

3.1 Introduction

The notion of a pseudorandom generator originally introduced by Yao [Yao82] has become by now one of the most important concepts in theoretical computer science penetrating virtually all its subareas. In its simplest form it says the following: a mapping $G_n : \{0,1\}^n \to \{0,1\}^m$ is (computationally) secure w.r.t. some circuit class $\mathcal{C}$ if no "small" circuit $C(y_1, \ldots, y_m) \in \mathcal{C}$ can distinguish between the two probabilistic distributions $G_n(\boldsymbol{x})$ and $\boldsymbol{y}$ in the sense that $|\mathbf{P}[C(G_n(\boldsymbol{x})) = 1] - \mathbf{P}[C(\boldsymbol{y}) = 1]|$ is small ($\boldsymbol{x}$ is picked at random from $\{0,1\}^n$, and $\boldsymbol{y}$ is picked at random from $\{0,1\}^m$).

Given the importance of pseudorandom generators for computational complexity, it is natural to wonder which mappings $G_n : \{0,1\}^n \to \{0,1\}^m$ should be considered hard from the perspective of proof complexity? In this chapter we propose the following paradigm: a generator $G_n : \{0,1\}^n \to \{0,1\}^m$ is hard for some propositional proof system P if and only if for *every* string $b \in \{0,1\}^m$ there is no efficient P-proof of the (properly encoded) statement $G(x_1, \ldots, x_n) \neq b$ ($x_1, \ldots, x_n$ are treated as propositional variables). A similar suggestion is independently made in the recent preprint of Krajíček [Kra01a].

This definition is very natural: it simply says (to the extent allowed by our framework) that P can not efficiently prove even the most basic thing about the behavior of G_n, namely that it is not an onto mapping. In fact, one a priori reasonable concern might be exactly if this exceedingly natural requirement is not too strong, namely whether non-trivial generators (say, with $m \geq n+1$) can exist at all. This concern is best addressed by exhibiting how several known lower bound results fit into our framework; these examples also explain some of our motivations for introducing this concept.

Example 1 (Tseitin tautologies) Let $G = (V, E)$ be a connected undirected graph. Consider the ($\mathbb{F}_2$-linear) mapping $T_G : \{0,1\}^E \to \{0,1\}^V$ given by $T_G(\vec{x})_v \stackrel{\text{def}}{=} \oplus_{e \ni v} x_e$, where $\vec{x} \in \{0,1\}^E$ is a $\{0,1\}$-valued function on edges. Then $b \in \{0,1\}^V$ is not in $\text{im}(G)$ if and only if $\oplus_{v \in V} b_v = 1$, and if we properly encode this statement in propositional logic, we arrive exactly at the tautologies introduced by Tseitin in his seminal paper [Tse68]. These tautologies turned out to be extremely useful in propositional proof complexity, and the many strong lower bounds proved for them [Tse68, Urq87, BW99, Gri98, BGIP01, Gri01, ABRW02] never depend on the particular choice of $b \in \{0,1\}^V$. This means that all of them can be viewed as showing that the generators T_G are hard for the corresponding proof system, as long as the graph G itself has good expansion properties.

Tseitin generators $T_G : \{0,1\}^E \to \{0,1\}^V$ make little sense from the computational point of view since the size of the seed $|E|$ is larger than the size of the output $|V|$. Our next two examples are more satisfactory in this respect.

Example 2 (Natural Proofs) Let $G_n : \{0,1\}^{n^k} \to \{0,1\}^{2^n}$ be any pseudorandom *function generator* that stretches n^k random bits to a Boolean function in n variables viewed as a string of length 2^n in its truth-table representation. Assume that G_n is hard w.r.t. $2^{O(n)}$-sized circuits. Razborov and Rudich [RR97] proved that there is no "natural" (in the strict sense also defined in that paper) proof of superpolynomial lower bounds for any complexity class C that can efficiently compute G_n. Their argument shows in fact that any natural circuit lower bound techniques fail to prove

that a given function f_n does not belong to the image of G_n. Stating it equivalently, for *any* function f_n there is no natural proof of the fact $f_n \notin \text{im}(G_n)$. Although in this result we are primarily interested in the case when f_n is the restriction of SAT (or any other **NP**-complete predicate) onto strings of length n, the argument, like in Example 1 absolutely does not depend on the particular choice of f_n.

One might argue that Natural Proofs do not correspond to a propositional proof system at all, and that their definition rather explicitly includes the transition "the proof works for a single $f_n \Rightarrow$ it works for many f_n", which provides the link to the ordinary (randomized) definition of a pseudorandom generator. Our next example illustrates that this drawback sometimes can be circumvented.

Example 3 (NP-natural proofs) Razborov [Raz95a] has proposed studying the set of tautologies $\neg Circuit_t(f_n)$, expressing the fact that the function f_n cannot be computed by a circuit of size t. Alekhnovich noted that this tautology is actually a generator tautology: the generator G simply sends (an encoding of) a circuit of size t, to the truth table of the function computed by it.

Now assume there is *some* proof system P-proving that *some* Boolean function f_n is not in the image of G. This constitutes an **NP**-cerftificate[1] of hardness of f_n. Using the derandomization machinary of the NW-generator [NW94, BFNW93, IW97, IKW01] it follows that for e.g size $t = 2^{\epsilon n}$ (with arbitrary $\epsilon > 0$), such a certificate implies that $\mathbf{MA} = \mathbf{NP}$ (and in particular also $\mathbf{BPP} \subseteq \mathbf{NP}$[2]).

Put differently, assuming $\mathbf{MA} \neq \mathbf{NP}$ we conclude that for the generator G above with this choice of t, there are no efficient proofs that $f_n \notin \text{im}(G)$ for any sequence of functions f_n, in any propositional proof system whatsoever! It should be stresed though, that some of the authors believe the conclusion much more than the assumption. Nevertheless, the connection is illuminating another relationship between computational and proof complexity, and the importance of generators in both.

Our final example provides us with *conditional* lower bounds for tautologies based

[1]This fits the natural proof framework above and may be called **NP**-natural proof, only it does not use the so called "largeness condition" of Razborov and Rudich.

[2]Follows from $\mathbf{BPP} \subseteq \mathbf{MA}$ of Goldreich and Zuckerman [GZ97]

on computationally secure generators. Unfortunately it holds only for the systems that posses efficient interpolation property.

Example 4 (Hardness in presence of Feasible Interpolation) Let $G_n : \{0,1\}^n \to \{0,1\}^m$ be an arbitrary pseudorandom generator that is hard w.r.t. poly-size (in $m+n$) circuits, and let $n < m/2$. Following Razborov [Raz95b], let us take bitwise XOR of two independent copies of this generator $G'_n : \{0,1\}^{2n} \to \{0,1\}^m$; $G'_n(x_1,\ldots,x_n,x'_1,\ldots,x'_n) \stackrel{\text{def}}{=} G_n(x_1,\ldots,x_n) \oplus G_n(x'_1,\ldots,x'_n)$. Then G'_n is hard for any propositional proof system P which has the property of feasible interpolation. System P is said to have feasible interpolation iff whenever there exists a polynomial size P-proof of the formula $\varphi(x,y) \vee \psi(x,z)$ there also exists a polynomial circuit $C(x)$ that given x decides whether $\varphi(x,y)$ is true for all y or $\psi(x,z)$ is true for all z. (for more details see e.g. [Kra97] or [BP98]).

Indeed, assume for the sake of contradiction that G'_n is easy for a proof system that possesses feasible interpolation. This means that in this system there exists a polynomial size proof of $b \notin \mathrm{im}(G'_n)$, for some string $b \in \{0,1\}^m$. Let $\boldsymbol{r}$ be picked uniformly and at random from $\{0,1\}^m$, and consider the propositional formula encoding the statement $\boldsymbol{r} \notin \mathrm{im}(G_n) \vee \boldsymbol{r} \notin \mathrm{im}(G_n \oplus b)$. The fact $b \notin \mathrm{im}(G'_n)$ implies that this is a tautology and thus, by feasible interpolation, there exists a polynomial size circuit C that given $\boldsymbol{r}$ correctly tells us whether $\boldsymbol{r} \notin \mathrm{im}(G_n)$ or $\boldsymbol{r} \notin \mathrm{im}(G_n \oplus b)$. One of these answers occurs with probability at least $1/2$; thus, C can be used to break the generator G.

The study of such a keystone concept in computational complexity as pseudorandom generators, but in the new framework of proof complexity, should be interesting in its own right. As suggested by the examples above, we also keep one quite pragmatic goal in mind: we believe that pseudorandomness is methodologically the right way to think of lower bounds in the proof-theoretic setting for really strong proof systems. Whenever we have a generator $G_n : \{0,1\}^n \to \{0,1\}^{n+1}$ which is hard for a propositional proof system P, we have lower bounds for P. If we manage to increase significantly the number of output bits and construct a poly-time computable

function generator $G_n : \{0,1\}^{n^k} \to \{0,1\}^{2^n}$ that is hard for P, we, similarly to [RR97, Raz95b], can conclude that in the framework proposed in [Raz95a] there are no efficient P-proofs of $\mathbf{NP} \not\subseteq \mathbf{P}/poly$. [3]

In this chapter we begin looking at a class of generators inspired by Nisan-Wigderson generator [NW94] on the one hand, and by Example 1 on the other. Let A be an $(m \times n)$ 0-1 matrix, $g_1(x_1, \ldots, x_n), \ldots, g_m(x_1, \ldots, x_n)$ be Boolean functions such that g_i essentially depends only on the variables $X_i(A) \stackrel{\text{def}}{=} \{x_j \mid a_{ij} = 1\}$, and $G_n : \{0,1\}^n \to \{0,1\}^m$ be given by $G_n(x_1, \ldots, x_n) \stackrel{\text{def}}{=} (g_1(x_1, \ldots, x_n), \ldots, g_m(x_1, \ldots, x_n))$. Nisan and Wigderson [NW94] proved that if A satisfies certain combinatorial conditions (namely, if it is a (k, s)-design for suitable choice of parameters), and the functions g_i are computationally hard, then G_n is a good pseudorandom generator in the computational sense. In this chapter we study which combinatorial properties of the matrix A and which hardness assumptions imposed on g_i guarantee that the resulting generator G_n is hard for such proof systems as Resolution or Polynomial Calculus.

The framework of proof complexity, however, adds also the third specific dimension that determines hardness properties of G_n. Namely, in our examples the base functions g_i are at least supposed to be hard for the circuit class underlying the propositional proof system P. Thus, P can not even express the base functions, and we should encode them using certain extension variables. Using these extension variables, our tautologies can be written as 3-CNFs, and thus can be expressed in any proof system. The choice of encoding makes an important part of the framework. We propose three different encodings - functional, circuit, and linear encodings, all natural from both computational and proof complexity viewpoints.

Our results are strong lower bounds for each of these encodings (and appropriate choices of base functions and combinatorial properties of the matrix A) in such standard proof systems like Resolution, Polynomial Calculus, and PCR (which com-

[3] The general idea of this reduction is similar to the reduction in Example 2: the propositional system can not prove efficiently that an explicit **NP**-complete function does not belong to the image of G. However, for every particular system the details of implementation are a little bit different, and one has to be extra careful for weak proof systems.

bines the power of both). Naturally, the results get weaker as the encoding strength increases.

We strongly believe that this set of tautologies can serve as hard examples for much stronger systems, and specifically that the hardness of the base functions in the generators should be a key ingredient in the proof. This factor is evident in our modest results above, and if extended to stronger systems, it may be viewed as a generalization of the feasible interpolation results, reducing in a sense proof complexity to computational complexity.

The chapter is organized as follows. In Section 3.2 we give necessary definitions and describe precisely combinatorial properties of the matrix A, hardness conditions imposed on the base functions g_i and types of their encodings needed for our purposes.

The next section 3.3 contains our hardness results for resolution width and polynomial calculus degree that hold for the most general functional encoding similar in spirit to the Functional Calculus from [ABRW02]. These can be considered as far-reaching generalizations of lower bounds for Tseitin tautologies from [BW99, BGIP01]. We also state here size lower bounds directly implied by our results via the known width/size and degree/size relations.

Section 3.4 contains a stronger lower bound for the weaker linear encoding. In Section 3.5 we consider the question of maximizing the number of output bits $m = m(n)$ in the generators constructed in the previous sections. For that purpose we show that with high probability a random matrix A has very good expansion properties. The chapter is concluded by several open questions in Section 3.6.

3.2 Preliminaries

Let A be an $(m \times n)$ 0-1 matrix,

$$J_i(A) \stackrel{\text{def}}{=} \{ j \in [n] \mid a_{ij} = 1 \}, \tag{3.1}$$

$X_i(A) \stackrel{\text{def}}{=} \{x_j \mid j \in J_i(A)\}$ and $g_1(x_1,\ldots,x_n),\ldots,g_m(x_1,\ldots,x_n)$ be Boolean functions such that $Vars(g_i) \subseteq X_i(A)$. We will be interested in systems of Boolean equations

$$\begin{cases} g_1(x_1,\ldots,x_n) = 1 \\ \qquad\cdots \\ g_m(x_1,\ldots,x_n) = 1. \end{cases} \tag{3.2}$$

We want to state combinatorial properties of the matrix A and hardness conditions of the base functions g_i such that if we properly encode the system (3.2) as a CNF $\tau(A,\vec{g})$, then every refutation of this CNF in a propositional proof system P must be long. This sentence has four ingredients, and the necessary definitions for each of them are provided fairly independently.

3.2.1 Combinatorial properties of the matrix A

All hardness results proved in this chapter will be based on the following combinatorial property generalizing the classical "edge-expansion" property for ordinary graphs.

Definition 3.2.1 For a set of rows $I \subseteq [m]$ in the matrix A, we define its *boundary* $\partial_A(I)$ as the set of all $j \in [n]$ (called *boundary elements*) such that $\{a_{ij} \mid i \in I\}$ contains exactly one 1. We say that A is an (r,s,c)-*expander* if $|J_i(A)| \leq s$ for all $i \in [m]$ and $\forall I \subseteq [m](|I| \leq r \Rightarrow |\partial_A(I)| \geq c \cdot |I|)$.

Let us relate (r,s,c)-expanders to several other combinatorial properties already known from the literature.

Example 5 For an ordinary graph $G = (V,E)$, its *edge-expansion coefficient* $c_E(G)$ is defined by

$$c_E(G) \stackrel{\text{def}}{=} \min_{|U| \leq |V|/2} \frac{e(U, V-U)}{|U|},$$

where $e(U,W)$ is the number of edges between U and W (see e.g., [Alo98] and the literature cited therein). Let A_G be the incidence matrix of a graph G with m vertices and n edges (i.e., $a_{ve} \stackrel{\text{def}}{=} 1$ if and only if $v \in e$), and let d be the maximal degree of a vertex in G. Then A_G is an $(m/2,d,c)$-expander if and only if $c_E(G) \geq c$.

Example 6 Let us turn to the combinatorial property originally used in [N91, NW94]. A matrix A is called (k,s)-*design* if $|J_i(A)| = s$ for all $i \in [m]$ and

$$|J_{i_1}(A) \cap J_{i_2}(A)| \leq k \tag{3.3}$$

for all $1 \leq i_1 < i_2 \leq m$. We have the following:

Fact 1 *Every (k,s)-design is also an $(r, s, s-kr)$-expander for any parameter r.*

Proof. Let $I \subseteq [m]$ and $|I| \leq r$. Then, due to the property (3.3), every $J_i(A)$ with $i \in I$ has at most $k \cdot (r-1)$ elements which are *not* in $\partial_A(I)$. Hence it contains at least $s - k \cdot (r-1)$ elements which *are* in $\partial_A(I)$.■

3.2.2 Hardness conditions on the base functions

As explained in the Introduction, we are interested in the methods which, given a mapping $G_n : \{0,1\}^n \to \{0,1\}^m$, allow us to show that the fact $b \notin \mathrm{im}(G_n)$ is hard to prove *for every* $b \in \{0,1\}^m$. This means that we want our lower bounds on the refutation complexity to work uniformly not only for the system (3.2) but also for *all* 2^m shifted systems

$$\begin{cases} g_1(x_1, \ldots, x_n) = b_1 \\ \qquad \ldots \\ g_m(x_1, \ldots, x_n) = b_m, \end{cases}$$

$b \in \{0,1\}^m$. We will enforce this simply by requiring that the conditions placed on the base functions $g_1, \ldots, g_m$ are symmetric, i.e., they are satisfied by some f if and only if they are satisfied by $(\neg f)$.

Definition 3.2.2 A Boolean function f is *ℓ-robust* if every restriction ρ such that $f|_\rho = const$, satisfies $|\rho| \geq \ell$.

Clearly, this property is symmetric. The most important example of robust functions are the PARITY functions $x_1 \oplus \cdots \oplus x_n \oplus b$, $b \in \{0,1\}$, which are n-robust.

Our strongest hardness results for the polynomial calculus work only for this specific function.

In fact, ℓ-robust functions are already well familiar from the computational complexity literature. [FSS84, Ajt83, Yao85, Hås86] proved *computational* lower bounds for ℓ-robust functions (when ℓ is close to $n = |Vars(f)|$) w.r.t. bounded-depth circuits. $(1-\theta)n$-robust functions (where θ is meant to be a small positive constant) were recently used in [BST98] for obtaining strong lower bounds for branching programs (property "$\mathcal{P}(\theta)$"). In this chapter we will use ℓ-robust functions for constructing generators that are hard for propositional *proof* systems. It is easy to see that most functions on n-bits are (say) $0.9n$-robust.

3.2.3 Encodings

Having constructed the system (3.2), we still should decide how to represent it in propositional logic. This step is non-trivial since we are deliberately interested in the case when the propositional system P can not directly speak of the functions $g_1, \ldots, g_m$. We consider three major possibilities: *functional*, *circuit* and *linear* encodings: all of them lead to CNFs that in fact w.l.o.g. can be further restricted to 3-CNFs (see the proof of Corollary 3.3.5 below).

Functional encoding

This is the strongest possible encoding which is also universal in the sense that it obviously simulates any other conceivable encoding (in fact, it is a "localized" variant of the Functional Calculus system considered in [ABRW02]).

Definition 3.2.3 Let A be an $(m \times n)$ 0-1 matrix. For every Boolean function f with the property $\exists i \in [m](Vars(f) \subseteq X_i(A))$ we introduce a new *extension variable* y_f. Let $Vars(A)$ be the set of all these variables. For the sake of convenience, single variables sometimes will be denoted as x_i instead of y_{x_i}. For a clause $C = y_{f_1}^{\epsilon_1} \vee \ldots \vee y_{f_w}^{\epsilon_w}$ in the variables $Vars(A)$, denote by $||C||$ the Boolean function in the variables $x_1, \ldots, x_n$ given by $|C| \stackrel{\text{def}}{=} f_1^{\epsilon_1} \vee \ldots \vee f_w^{\epsilon_w}$.

Given Boolean functions $\vec{g} = (g_1, \ldots, g_m)$ such that $Vars(g_i) \subseteq X_i(A)$, we denote by $\tau(A, \vec{g})$ the CNF in the variables $Vars(A)$ that consists of all the clauses $C = y_{f_1}^{\epsilon_1} \vee \ldots \vee y_{f_w}^{\epsilon_w}$ for which there exists $i \in [m]$ such that

$$Vars(f_1) \cup \ldots \cup Vars(f_w) \subseteq X_i(A) \tag{3.4}$$

and

$$g_i \models ||C||. \tag{3.5}$$

Fact 2 *$\tau(A, \vec{g})$ is satisfiable if and only if the system (3.2) is consistent.*

Proof. If $(a_1, \ldots, a_n)$ is a solution to (3.2), then the assignment which assigns every y_f to $f(a_1, \ldots, a_n)$ is satisfying for $\tau(A, \vec{g})$. For the other direction, let $\vec{b} = (b_f | y_f \in Vars(A))$ be a satisfying assignment for $\tau(A, \vec{g})$. Let $a_j \stackrel{\text{def}}{=} b_{x_j}$; then, using those axioms $y_{f_1}^{\epsilon_1} \vee \ldots \vee y_{f_w}^{\epsilon_w}$ from $\tau(A, \vec{g})$ for which $f_1^{\epsilon_1} \vee \ldots \vee f_w^{\epsilon_w} \equiv 1$, we can show by induction on the circuit size of f that $b_f = f(a_1, \ldots, a_n)$ for every $y_f \in Vars(A)$. In particular, $g_i(a_1, \ldots, a_n) = b_{g_i} = 1$ (since $\tau(A, \vec{g})$ contains the axiom y_{g_i}). Thus, the vector $(a_1, \ldots, a_n)$ is a solution to the system (3.2).■

Circuit encoding

This encoding is much more economical in terms of the number of variables than the functional encoding. Also, it looks more natural and better conforming to the underlying idea of the Extended Frege proof system. The tautologies under this encoding will be polynomial-size as long as all g_i's have poly-size circuits, and thus are potentially hard for Frege (assuming $\mathbf{P}/poly$ contains functions computationally hard for $\mathbf{NC^1}/poly$).

Definition 3.2.4 Let A be an $(m \times n)$ 0-1 matrix, and $C_1, \ldots, C_m$ be single-output Boolean circuits over an arbitrary fixed finite basis, C_i being a circuit in the variables $X_i(A)$. For every $i \in [m]$ and every gate v of the circuit C_i we introduce a special *extension variable* y_v, and we identify extension variables corresponding to input

gates labeled by the same variable x_j. Let $Vars_{\vec{C}}(A)$ be the set of all these extension variables.

By $\tau(A, \vec{C})$ we denote the CNF that consists of the following clauses:

1. $y_{v_1}^{\bar{\epsilon}_1} \vee \ldots \vee y_{v_d}^{\bar{\epsilon}_d} \vee y_v^{\pi(\epsilon_1,\ldots,\epsilon_d)}$, whenever $v := \pi(v_1, \ldots, v_d)$ is an instruction of one of the circuits $C_1, \ldots, C_m$ and $\epsilon \in \{0,1\}^d$ is an arbitrary vector;

2. y_{v_i} when v_i is the output gate of C_i, for all $i \in [m]$.

For a circuit C, let $||C||$ be the Boolean function it computes.

Fact 3 *$\tau(A, \vec{C})$ is satisfiable if and only if the system $||C_1|| = \ldots = ||C_m|| = 1$ is consistent.*

Proof. Similar to the proof of Fact 2.■

Fact 4 *There exists a substitution σ of variables from $Vars_{\vec{C}}(A)$ by variables from $Vars(A)$ such that $\sigma(\tau(A, \vec{C}))$ is a subset of the set of clauses $\tau(A, ||\vec{C}||)$. In particular, every refutation of $\tau(A, \vec{C})$ in every "reasonable" propositional proof system can be transformed (by applying σ) into a refutation of $\tau(A, ||\vec{C}||)$ in the same system which is simpler w.r.t. any "reasonable" complexity measure.*

Proof. Let $\sigma(y_v) \stackrel{\text{def}}{=} y_{||v||}$, where $||v||$ is the function computed by the gate v.■

Linear encoding

This encoding makes sense only when the functions g_i are $\mathbb{F}_2$-linear forms (for historical reasons, this special case of NW-generators is often referred to as *Nisan generators*). In some cases it is more economical than the functional encoding in terms of the number of variables. Also, it is much better structured, and we will take advantage of this in Section 3.4.

Definition 3.2.5 Let A be an $(m \times n)$ 0-1 matrix. For every $J \subseteq [n]$ such that $\exists i \in [m](J \subseteq J_i(A))$ we introduce a new *extension variable* y_J (with the intended meaning $y_J \sim \oplus_{j \in J} x_j$). Let $Vars_{\oplus}(A)$ be the set of all these variables.

Given a Boolean vector $b \in \{0,1\}^m$, we denote by $\tau_{\oplus}(A,b)$ the CNF in the variables $Vars_{\oplus}(A)$ that consists of the following clauses:

1. $y_{J_1}^{\bar{\epsilon}_1} \vee \ldots \vee y_{J_d}^{\bar{\epsilon}_d}$, whenever there exists $i \in [m]$ such that $J_1 \cup \ldots \cup J_d \subseteq J_i(A)$, the symmetric difference $J_1 \triangle \ldots \triangle J_d$ is empty and $\bar{\epsilon}_1 \oplus \ldots \oplus \bar{\epsilon}_d = 1$;

2. $y_{J_i(A)}^{b_i}$, for all $i \in [m]$.

Let us denote by $\Sigma_i(A,b_i)$ the Boolean function $\bigoplus_{j \in J_i(A)} x_j \oplus \bar{b}_i$.

Fact 5 *$\tau_{\oplus}(A,b)$ is satisfiable if and only if the system $\Sigma_1(A,b_1) = \Sigma_2(A,b_2) = \ldots = \Sigma_m(A,b_m) = 1$ of linear equations over $\mathbb{F}_2$ is consistent.*

Proof. Follows from the observation that the conjunction of clauses $y_{J_1}^{\bar{\epsilon}_1} \vee \ldots \vee y_{J_d}^{\bar{\epsilon}_d}$ for all $\bar{\epsilon}_1 \oplus \ldots \oplus \bar{\epsilon}_d = 1$ is semantically equivalent to the formula $\bigoplus_{i=1}^{d} y_{J_i} = 0$.■

Fact 6 *There exists a substitution σ of variables from $Vars_{\oplus}(A)$ by variables from $Vars(A)$ such that $\sigma(\tau_{\oplus}(A,b))$ is a subset of the set of clauses $\tau(A,\vec{\Sigma}(A,b)) \stackrel{\text{def}}{=} \tau(A,\Sigma_1(A,b_1),\Sigma_2(A,b_2),\ldots,\Sigma_m(A,b_m))$.*

Proof. $\sigma(y_J) \stackrel{\text{def}}{=} y_{\bigoplus_{j \in J} x_j}$.■

It might be instructive to look at the place occupied in our framework by original Tseitin tautologies (cf. Examples 1,5). Let A_G be the incidence matrix of an undirected graph G. Then our framework provides three different ways[4] to talk of Tseitin tautologies for graphs G of *arbitrary* degree. All these possibilities are *reasonable* in the sense that although the resulting CNF τ may have a huge size, it always possesses a sub-CNF of *polynomial* size that is still unsatisfiable. The fourth (unreasonable!) encoding is *primitive*: we allow no extension variables at all and simply represent the functions $\Sigma_i(A,b_i)$ themselves as CNFs of exponential size. For graphs of bounded degree (which is the only case researchers were interested in prior to this research), the

[4]For the circuit encoding we additionally have to fix some natural circuits computing the functions $\Sigma_i(A,b_i)$.

subtle differences between the four encodings disappear, and the whole rich spectrum of various possibilities collapses to ordinary Tseitin tautologies.

In fact, the unreasonable primitive encoding can in principle be considered in our framework as well. Namely, as we will see in Section 5, good (r, s, c)-expanders exist even for large constants s (say, $s = 10$). And for constant values of s results proved in any of our reasonable encodings can be translated to the primitive encoding with only constant time increase in the size of the tautology. The primitive encoding, however, is very counterintuitive to the main idea that the base functions g_i's should be hard for the circuit class underlying our propositional theory, and to the hope of using these tautologies for stronger proof systems. For this reason we do not discuss here neither the primitive encoding itself, nor the trade-off between the tautology size and the bounds appearing in this encoding when $s \to \infty$.

3.3 Lower bounds on width and degree in the functional encoding

In this section we establish strong lower bounds on the resolution width and PC degree in the most general functional encoding, and derive from them some size lower bounds. Our results in this section can be viewed as a far-reaching generalization of the corresponding lower bounds for Tseitin tautologies from [BW99, BGIP01].

But first a word about important and less important parameters. The parameters s, c, l of the defining tautologies will feature in most of the calculations (recall that s is the number of 1's in each row of the matrix A, which is also the number of arguments to each function g_i, c is the expansion factor of the matrix A, and ℓ will lower bound the robustness of the g_i's). We will show in Section 3.5 that almost all matrices satisfy $c > 0.9s$. Similarly, most functions satisfy $\ell > 0.9s$. Assuming this, Theorem 3.3.1 and Theorem 3.3.7 provide $\Omega(r)$ lower bounds on the width of Resolution and degree of Polynomial Calculus, respectively (recall that r is the key parameter defining what size sets expand, and can be taken to be essentially n/s;

see Section 3.5 for details). Our corollaries for the size lower bounds implied by the width and degree lower bounds will be stated (for simplicity) only for this situation.

Theorem 3.3.1 *Let A be an (r, s, c)-expander of size $(m \times n)$, and $g_1, \ldots, g_m$ be ℓ-robust functions with $Vars(g_i) \subseteq X_i(A)$, where $c + \ell \geq s + 1$. Then every resolution refutation of $\tau(A, \vec{g})$ must have width $> \frac{r(c+\ell-s)}{2\ell}$.*

Proof. The proof follows the ideology developed in [BW99]. We define a measure μ with sub-additive growth on the clauses, we show that the measure of the empty clause is large ($\mu(0) > r$), hence there must be a clause with medium size measure ($r/2 < \mu(C) \leq r$). We show that such a clause must have large width.

Fix an (r, s, c)-expander A of size $(m \times n)$ and ℓ-robust functions $g_1, \ldots, g_m$ with $Vars(g_i) \subseteq X_i(A)$, where $c + \ell \geq s + 1$.

Definition 3.3.2 *For C a clause in the variables $Vars(A)$, define $\mu(C)$ to be the minimal size of $I \subseteq [m]$ such that the following pair of conditions hold:*

$$\forall y_f^\epsilon \in C \; \exists i \in I \; (Vars(f) \subseteq X_i(A)); \tag{3.6}$$

$$\{g_i \mid i \in I\} \models |C|. \tag{3.7}$$

Claim 3.3.3 *1. For a clause C with $r/2 < \mu(C) \leq r$, $w(C) \geq \frac{r(c+\ell-s)}{2\ell}$.*

2. $\mu(0) > r$.

Proof. **Part 1)** Let I be a set of minimal size satisfying Definition 3.3.2. Since $|I| \leq r$, we get $|\partial_A(I)| \geq c \cdot |I|$. Let us partition I into I_0, any minimal subset satisfying (3.6), and $I_1 = I \setminus I_0$. Notice that by the minimality of I, removing any row from I_1 will ruin property (3.7).

We claim that for any $i_1 \in I_1$, $J_{i_1}(A)$ has small intersection with $\partial_A(I)$. Namely,

$$|J_{i_1}(A) \cap \partial_A(I)| \leq s - \ell. \tag{3.8}$$

Indeed, as we noticed above, $\{g_i \mid i \in I \setminus \{i_1\}\} \not\models |C|$. Let α be any assignment such that $g_i(\alpha) = 1$ $(i \in I \setminus \{i_1\})$ but $|C|(\alpha) = 0$. Let ρ be the restriction given by

$$\rho(x_j) \stackrel{\text{def}}{=} \begin{cases} \alpha(x_j) \text{ if } j \notin \partial_A(I) \cap J_{i_1}(A) \\ \star \text{ if } j \in \partial_A(I) \cap J_{i_1}(A). \end{cases}$$

Then, since ρ is totally defined on $Vars(g_i)$ for $i \neq i_1$, and also on $Vars(|C|)$ (by (3.6) and $i_1 \notin I_0$) we have $g_i|_\rho \equiv 1$ $(i \neq i_1)$ and $C|_\rho \equiv 0$. Hence, using (3.7), we conclude that $g_{i_1}|_\rho \equiv 0$. Since g_{i_1} is ℓ-robust and $|J_{i_1}(A)| \leq s$, this implies the desired inequality (3.8).

Now we may sum up:

$$\left.\begin{aligned} c \cdot |I| &\leq |\partial_A(I)| \\ &\leq s \cdot |I_0| + (s-\ell)|I_1| \\ &= (s-\ell)|I| + \ell \cdot |I_0| \\ &\leq (s-\ell)|I| + \ell \cdot w(C), \end{aligned}\right\} \quad (3.9)$$

which implies $w(C) \geq \frac{|I|(c+\ell-s)}{\ell}$. Recalling that $|I| > r/2$, we get our bound. Part 1) is proven.

Part 2) Suppose the contrary, that is $\mu(0) \leq r$. Then we can repeat the first part of the above argument (since that part did not use the condition $|I| > r/2$) and still get (3.9). But now $I_0 = \emptyset$, hence (3.9) alone implies a contradiction with the expansion property. This proves part 2).■

Claim 3.3.4 *Any resolution refutation of $\tau(A, \vec{g})$ must include a clause C with $r/2 < \mu(C) \leq r$.*

Proof. μ is sub-additive, i.e. if C was derived from C_0, C_1 by a single resolution step, then $\mu(C) \leq \mu(C_0) + \mu(C_1)$. Additionally, for any axiom C, $\mu(C) = 1$. The

statement now follows from Claim 3.3.3(2).■

Theorem 3.3.1 is immediately implied by Claims 3.3.4, 3.3.3(1).■

In order to see which *size* lower bounds are implied by Theorem 3.3.1 via Proposition 2.2.1, we consider only the typical (and most important) case $c+\ell-s=\Omega(s)$, for which our width lower bound is $\Omega(r)$.

Corollary 3.3.5 *Let $\epsilon>0$ be an arbitrary fixed constant, A be an $(r,s,\epsilon s)$-expander of size $(m\times n)$, and $g_1,\ldots,g_m$ be $(1-\epsilon/2)s$-robust functions. Then every resolution refutation of $\tau(A,\vec{g})$ must have size* $\exp\left(\Omega\left(\frac{r^2}{m\cdot 2^{2s}}\right)\right)/2^s$.

Proof. Fix a resolution refutation of $\tau(A,\vec{g})$ that has size S. It is easy to see that every axiom in $\tau(A,\vec{g})$ contains a sub-clause of width $\leq 2^s$ which is also an axiom of $\tau(A,\vec{g})$. Moreover, this latter clause can be easily inferred in $O(2^s)$ steps from those axioms in $\tau(A,\vec{g})$ that have width ≤ 3. This allows us to replace the original refutation by a refutation that may have a slightly bigger size $O(S\cdot 2^s)$ but uses only those axioms from $\tau(A,\vec{g})$ that have width ≤ 3. In this new refutation we infer all clauses of $\tau(A,\vec{g})$ that were used in the original refutation from width 3 clauses and then apply the original refutation itself. Hence, by Proposition 2.2.1, $\tau(A,\vec{g})$ also has a resolution refutation of width $O\left(\sqrt{|Vars(A)|\cdot\log(S\cdot 2^s)}\right)\leq O\left(\sqrt{m\cdot 2^{2s}\cdot\log(S\cdot 2^s)}\right)$. Comparing this with the lower bound of $\Omega(r)$ that comes from Theorem 3.3.1, we finish the proof of Corollary 3.3.5.■

We can obtain much better size lower bounds (i.e., get rid of the disappointing term 2^{2^s} in the denominator) for the circuit encoding. We further confine ourselves to the optimal case when the circuits $C_1,\ldots,C_m$ have size $O(s)$.

Corollary 3.3.6 *Let $\epsilon>0$ be an arbitrary fixed constant, A be an $(r,s,\epsilon s)$-expander of size $(m\times n)$, and $C_1,\ldots,C_m$ be single-output Boolean circuits over arbitrary fixed finite basis such that C_i is a circuit of size $O(s)$ in the variables $X_i(A)$, and all functions $||C_i||$ are $(1-\epsilon/2)s$-robust. Then every resolution refutation of $\tau(A,\vec{C})$ must have size* $\exp\left(\Omega\left(\frac{r^2}{ms}\right)\right)$.

Proof. By Fact 4 and Theorem 3.3.1, every resolution refutation of $\tau(A, \vec{C})$ must have width $\Omega(r)$. Since $|Vars_{\vec{C}}(A)| \leq O(ms)$, the required bound immediately follows from Proposition 2.2.1.■

Our second major result in this section generalizes the bound from [BGIP01]. Unfortunately, it also inherits all the limitations of their technique: essentially the only base functions $g_1, \ldots, g_m$ we can handle are $\mathbb{F}_2$-linear forms, and for $char(F) = 2$ our approach fails completely (cf. [Gri98]). On the positive side, note that although we do require the linearity of the base functions, the bound itself still holds for the most general *functional* framework.

Theorem 3.3.7 *Let A be an (r, s, c)-expander of size $(m \times n)$, and $b_1, \ldots, b_m \in \{0, 1\}$. Then every PCR refutation of $\tau(A, \vec{\Sigma}(A, b))$ over an arbitrary field F with $char(F) \neq 2$ must have degree $\geq \frac{rc}{4s}$.*

Proof. As the first step toward proving Theorem 3.3.7, we show one simple reduction to a lower bound problem about PC refutations in the *original* variables $x_1, \ldots, x_n$. This step is very general and does not depend on the linearity of the base functions g_i.

Definition 3.3.8 For a Boolean function $f(x_1, \ldots, x_n)$, $P_f(x_1, \ldots, x_n)$ is the (unique) multi-linear polynomial such that

$$P_f(\alpha) = \begin{cases} 0 \text{ if } f(\alpha) = 1 \\ 1 \text{ if } f(\alpha) = 0 \end{cases}$$

for all $\alpha \in \{0, 1\}^n$.

Lemma 3.3.9 *For any $(m \times n)$ 0-1 matrix A and any functions $g_1, \ldots, g_m$ with $Vars(g_i) \subseteq X_i(A)$, every degree d PCR refutation of $\tau(A, \vec{g})$ can be transformed into a PC refutation of the system*

$$P_{g_1} = \ldots = P_{g_m} = 0 \tag{3.10}$$

(in the original *variables* $x_1, \ldots, x_n$*) that has degree* $\leq s \cdot d$.

Proof of Lemma 3.3.9. Let us consider some PCR refutation π of $\tau(A, \vec{g})$. Substitute in π the polynomial $P_{f^\epsilon}(x_1, \ldots, x_n)$ for every variable y_f^ϵ. Since $\deg(P_{f^\epsilon}) \leq s$ for any $f(x_1, \ldots, x_n)$ such that $Vars(f) \subseteq X_i(A)$ for some $i \in [m]$, the degrees of all lines resulting from this substitution are at most $s \cdot d$. Moreover, any axiom from $\tau(A, \vec{g})$, as well as default axioms, gets transformed into a polynomial P such that for some $i \in [m]$ P contains only variables from $X_i(A)$, and is a semantical corollary of P_{g_i} on $\{0,1\}^{X_i(A)}$. Hence, it can be inferred from P_{g_i} in degree $\leq s$, using only variables from $X_i(A)$. Appending these auxiliary inferences to the beginning of the transformed refutation π, we obtain the required PC refutation of the system (3.10). Lemma 3.3.9 is proved.■

Thus, in order to complete the proof of Theorem 3.3.7, we should establish the $\frac{rc}{4}$ lower bound on the degree of any PC refutation π of the system (3.10) for $g_i = \Sigma_i(A, b_i)$.

The proof is based on the elegant connection between PC-degree and Gaussian width found in [BI99]. With this connection in hand, we may quote here, word by word, Theorem 3.3 from [BI99], plugging in our current parameters.

Theorem 3.3.10 *For* A *an* (r, s, c) *expander,* $\{g_i\}$ *linear equations mod* 2*, and* F *a field of characteristic* $\neq 2$*, any PCR refutation of* $P_{g_1} = \ldots = P_{g_m} = 0$ *has degree* $\geq \frac{rc}{4}$.

Theorem 3.3.7 follows.■

Corollary 3.3.11 *Let* $\epsilon > 0$ *be an arbitrary fixed constant,* A *be an* $(r, s, \epsilon s)$*-expander of size* $(m \times n)$ *and* $b_1, \ldots, b_m \in \{0,1\}$*. Then every PCR refutation of* $\tau(A, \vec{\Sigma}(A, b))$ *over an arbitrary field* F *with* $char(F) \neq 2$ *must have size* $\exp\left(\Omega\left(\frac{r^2}{m \cdot 2^{2s}}\right)\right)/2^s$.

Proof. Identical to the proof of Corollary 3.3.5, using Proposition 2.2.2.■

Corollary 3.3.12 *Let $\epsilon > 0$ be an arbitrary fixed constant, A be an $(r, s, \epsilon s)$-expander of size $(m \times n)$, $b_1, \ldots, b_m \in \{0,1\}$, and $C_1, \ldots, C_m$ be single-output Boolean circuits over arbitrary fixed finite basis such that C_i is a circuit of size $O(s)$ in the variables $X_i(A)$ that computes the function $\Sigma_i(A, b_i)$. Then every PCR refutation of $\tau(A, \vec{C})$ over an arbitrary field F with $char(F) \neq 2$ must have size* $\exp\left(\Omega\left(\frac{r^2}{ms}\right)\right)$.

Proof. Identical to the proof of Corollary 3.3.6, using Proposition 2.2.2.■

3.4 Size lower bounds for linear encoding

In this section we show better lower bounds (although our requirement on the expansion rate is somewhat stronger) on the size of PCR refutation for the more structured linear encoding than those provided by Corollaries 3.3.11, 3.3.12. We will apply the random restriction method for killing large clauses rather than directly refer to the general degree/size relation from Proposition 2.2.2. In this sense our approach is similar in spirit to that of [BP96].

Theorem 3.4.1 *Let A be an $(r, s, \frac{3}{4}s)$-expander of size $(m \times n)$, and let $b_1, \ldots, b_m \in \{0,1\}$. Then every PCR refutation of $\tau_{\oplus}(A, \vec{b})$ over an arbitrary field F with $char(F) \neq 2$ must have size* $\exp\left(\Omega\left(\frac{r^2}{m}\right)\right)$.

Proof. As the first step toward proving Theorem 3.4.1, we show how to get rid of the variables y_J for large (= of size $> s/2$) sets J. For technical reasons, we also switch during this step from the linear encoding to the functional one.

Definition 3.4.2 For an $(m \times n)$-matrix A, the set of variables $Vars_{\oplus}(A) \subseteq Vars(A)$ consists of those $y_f \in Vars(A)$ for which f has the form $\bigoplus_{j \in J} x_j$. Let also $\widetilde{Vars}_{\oplus}(A) \stackrel{\text{def}}{=} \left\{ y_{(\bigoplus_{j \in J} x_j)} \in Vars_{\oplus}(A) \,|\, |J| \leq s/2 \right\}$.

$\tau_{\oplus}(A, b)$ [$\tilde{\tau}_{\oplus}(A, b)$] is the set of those axioms in $\tau(A, \vec{\Sigma}(A, b))$ that contain variables only from $Vars_{\oplus}(A)$ [from $\widetilde{Vars}_{\oplus}(A)$, respectively].

It is worth noting that $\tau_{\oplus}(A, b)$ possesses the following clean algebraic description: if $g_i = \Sigma_i(a, b_i)$, and $f_1, \ldots, f_w$ are $\mathbb{F}_2$-linear forms then (3.5) holds if *either* the system

of linear equations $f_1 = \bar{\epsilon}_1, \ldots, f_w = \bar{\epsilon}_w$ is inconsistent *or* the vector space spanned by these equations contains $\bar{g}_i$.

Lemma 3.4.3 *Suppose that A is an $(2, s, \frac{3}{4}s)$-expander. Then every PCR refutation of $\tau_{\oplus}(A, b)$ can be transformed into a PCR refutation of $\tilde{\tau}_{\oplus}(A, b)$ that has the same size.*

Proof of Lemma 3.4.3. For every two distinct rows i_1 and i_2 we have $|\partial_A(\{i_1, i_2\})| \geq \frac{3}{2}s$ which implies $|J_{i_1}(A) \cap J_{i_2}(A)| \leq s/2$. Hence, for every $J \subseteq [n]$ with $|J| > s/2$ there can exist at most one row $i \in [m]$ such that $J \subseteq J_i(A)$. Therefore, the following mapping:

$$y_J \mapsto \begin{cases} y_{\bigoplus\{x_j \mid j \in J\}} \text{ if } |J| \leq s/2 \\ y_{\bigoplus\{x_j \mid j \in J_i(A) \setminus J\}} \oplus b_i \text{ if } |J| > s/2 \text{ and } J \subseteq J_i(A) \end{cases}$$

is well-defined. It is easy to see that it takes every axiom from $\tau_{\oplus}(A, b)$ to an axiom from $\tilde{\tau}_{\oplus}(A, b)$ which proves Lemma 3.4.3.■

Now, for a monomial $m = y_{f_1}^{\epsilon_1} \ldots y_{f_d}^{\epsilon_d}$ in the variables $\widetilde{Vars}_{\oplus}(A)$, we define its *A-degree* $\deg_A(m)$ as the minimal cardinality of a set of rows I with the property $Vars(f_1) \cup \ldots \cup Vars(f_d) \subseteq \bigcup_{i \in I} X_i(A)$. The *A-degree of a polynomial* is the maximal A-degree of a monomial in it, and similarly the *A-degree of a PCR* proof is the maximal A-degree of a polynomial in it. The following lemma rephrases Theorem 3.3.7 for $\deg_A$:

Lemma 3.4.4 *Let A be an (r, s, c)-expander of size $(m \times n)$, and $b_1, \ldots, b_m \in \{0, 1\}$. Then every PCR refutation of $\tau(A, \vec{\Sigma}(A, b))$ over an arbitrary field F with $char(F) \neq 2$ must have A-degree $\geq \frac{rc}{4s}$.*

Proof of Lemma 3.4.4.

The only difference from Theorem 3.3.7 is that we consider here A-degree instead of ordinary one. It is easy to see by inspection that this change does not affect the reduction in Lemma 3.3.9, and the same proof applies here as well.■

Lemmas 3.4.3 and 3.4.4 determine the strategy of the rest of the proof (cf. [BP96]). We want to hit the prospective refutation of $\tilde{\tau}_{\oplus}(A,b)$ by a random restriction $\boldsymbol{\rho}$ in such a way that $\boldsymbol{\rho}$ preserves the structure of $\tau(A,\vec{\Sigma}(A,b))$, and, if the size of the original refutation is small, with a high probability also kills all monomials in the variables $\widetilde{Vars}_{\oplus}(A)$ that have high A-degree.

Definition 3.4.5 For a set of rows I, let us denote by M_I the set of all restrictions ρ such that $\rho^{-1}(\{0,1\}) = \bigcup_{i\in I} X_i(A)$ and ρ satisfies all equations $\Sigma_i(A,b_i) = 1$ for all $i \in I$.

Note that if $|I| \leq r$, then, since A is an $(r,s,\frac{3}{4}s)$-expander, the linear forms $\bigoplus\{x_j \mid x_j \in X_i(A)\} = \Sigma_i(A,b_i) \oplus \bar{b}_i$ $(i \in I)$ are linearly independent (because every its subset has a form that contains a boundary variable) and thus M_I is a non-empty linear subspace.

Let $A|_I$ be the result of removing from the matrix A all rows $i \in I$ and all columns $j \in \bigcup_{i\in I} J_i(A)$. Any restriction $\rho \in M_I$ can be naturally extended to the variables from $Vars(A)$ by letting $\rho(y_f) \stackrel{\text{def}}{=} y_{f|_\rho}$. ρ takes variables from $Vars(A)$ to variables from $Vars(A|_I)$. Moreover, those y_f for which $\exists i \in I$ $(Vars(f) \subseteq X_i(A))$ are set to a constant. Finally, ρ always takes axioms from $\tau(A,\vec{g})$ to axioms from $\tau(A|_I,\vec{g}|_\rho)$. The only remaining problem is that $A|_I$ may not inherit good expansion properties: it is easy to get an example showing that it may even contain an empty row! We circumvent this difficulty by further removing all rows that have large intersection with $\bigcup_{i\in I} J_i(A)$, and show in the following lemma that this can always be done in an efficient manner.

Lemma 3.4.6 *Let A be an (r,s,c)-expander. Then every set of rows I with $|I| \leq r/2$ can be extended to a larger set of rows $\hat{I} \supseteq I$ such that $|\hat{I}| \leq 2\cdot|I|$ and $A|_{\hat{I}}$ is an $(r,s,3c-2s)$-expander.*

Proof of Lemma 3.4.6. Let us recursively add to I new rows (one row i_0 at a time) with the property $\left|J_{i_0}(A) \cap \left(\bigcup_{i\in I'} J_i(A)\right)\right| > 2(s-c)$, where I' is the current

value of I. We claim that this process will terminate (i.e., no new row can be added) in less than $|I|$ steps.

Suppose the contrary, and let $\hat{I}$ be the set of cardinality $2 \cdot |I|$ reached after $|I|$ steps. Then every row $i_0 \in \hat{I} \setminus I$ contains less than $|J_{i_0}(A) - 2(s-c)| \leq (2c-s)$ boundary elements from $\partial_A(\hat{I})$. Hence, $|\partial_A(\hat{I})| < s \cdot |I| + (2c-s) \cdot |I| = 2c \cdot |I|$, a contradiction.

We choose as our $\hat{I}$ the result of termination of this process. Let I_0 be a set of rows in $A|_{\hat{I}}$ (i.e., $I_0 \cap \hat{I} = \emptyset$) of cardinality at most r. Then $\partial_{A|_{\hat{I}}}(I_0) = \partial_A(I_0) \setminus \bigcup_{i \in \hat{I}} J_i(A)$. Since for every $i \in I_0$, $\left|J_{i_0}(A) \cap \left(\bigcup_{i \in \hat{I}} J_i(A)\right)\right| \leq 2(s-c)$, we have the bound $\left|\partial_{A|_{\hat{I}}}(I_0)\right| \geq |\partial_A(I_0)| - 2(s-c) \cdot |I_0| \geq c \cdot |I_0| - 2(s-c)|I_0| = (3c-2s) \cdot |I_0|$. Lemma 3.4.6 is proved.■

Now we are ready to finish the proof of Theorem 3.4.1. Fix a PCR refutation π of $\tilde{\tau}_{\oplus}(A,b)$. Assume w.l.o.g. that 18 divides r, and pick at random a set of rows $\boldsymbol{I}$ of cardinality $r/3$ (we are using boldface to stress that it is a random variable). Choose arbitrarily $\hat{\boldsymbol{I}} \supseteq \boldsymbol{I}$ according to Lemma 3.4.6, i.e., such that $|\hat{\boldsymbol{I}}| \leq \frac{2r}{3}$ and $A|_{\hat{\boldsymbol{I}}}$ is an $(r, s, s/4)$-expander. Pick $\boldsymbol{\rho} \in M_{\hat{\boldsymbol{I}}}$ at random, and apply this restriction to our PCR-refutation π. This will produce a PCR-refutation $\boldsymbol{\rho}(\pi)$ of $\tilde{\tau}_{\oplus}(A|_{\hat{\boldsymbol{I}}}, \boldsymbol{\rho}(\vec{\Sigma}(A,b)))$. By Lemma 3.4.4 (with $c = s/4$), $\boldsymbol{\rho}(\pi)$ must contain a non-zero monomial $\boldsymbol{\rho}(m)$ of $A|_{\hat{\boldsymbol{I}}}$-degree $> r/18$. Thus, π contains a monomial m that has A-degree $> r/18$ and *is not killed by* $\boldsymbol{\rho}$. In order to finish the proof, we only have to estimate from above the probability $\mathbf{P}[\boldsymbol{\rho}(m) \neq 0]$ for every individual monomial m with $\deg_A(m) > r/18$.

Fix any such $m = y_{f_1}^{\epsilon_1} \ldots y_{f_d}^{\epsilon_d}$, and recall that $f_1, \ldots, f_d$ are $\mathbb{F}_2$-linear forms of weight $\leq s/2$. W.l.o.g. assume that $f_1, \ldots, f_t$ form a linear basis of the space $Span(f_1, \ldots, f_d)$. Then $\bigcup_{\nu=1}^{t} Vars(f_\nu) = \bigcup_{\nu=1}^{d} Vars(f_\nu)$ and, therefore, $\deg_A(y_{f_1}^{\epsilon_1} \ldots y_{f_t}^{\epsilon_t}) = \deg_A(m) > r/18$. Hence, w.l.o.g. we can assume from the very beginning that $f_1, \ldots, f_d$ are linearly independent.

Let us now introduce one variation of the notion of A-degree. Namely, for $m = y_{f_1}^{\epsilon_1} \ldots y_{f_d}^{\epsilon_d}$, let $\deg'_A(m)$ be the minimal cardinality of a set of rows I such that these rows "cover" m in the stronger sense $\forall \nu \in [d] \exists i \in I (Vars(f_\nu) \subseteq X_i(A))$. Clearly,

$\deg_A(m) \leq \deg'_A(m)$ (and $\leq \deg(m)$). Also, $\deg'_A$ is "continuous" in the sense that for every monomial m, and every variable y_f^ϵ, $\deg'_A(m) \leq \deg'_A(m \cdot y_f^\epsilon) \leq \deg'_A(m)+1$. Therefore, we can gradually remove variables from the monomial m, one variable at a time, until we find in it a sub-monomial m' such that $\deg'_A(m')$ is *exactly* equal to $r/18$. For ease of notation, assume w.l.o.g. that $\deg'_A(m) = r/18$ for the original monomial m.

Fix now any set of rows I_0 with $|I_0| = r/18$ and such that

$$\forall \nu \in [d] \exists i \in I_0 (Vars(f_\nu) \subseteq X_i(A)). \tag{3.11}$$

We estimate the probability $\mathbf{P}[\boldsymbol{\rho}(m) \neq 0]$ as follows:

$$\mathbf{P}[\boldsymbol{\rho}(m) \neq 0] \leq \mathbf{P}\left[|I_0 \cap \boldsymbol{I}| < \frac{r^2}{100m}\right] + \max_{\substack{|I|=r/3 \\ |I_0 \cap I| \geq \frac{r^2}{100m}}} \mathbf{P}[\boldsymbol{\rho}(m) \neq 0 \,|\, |\boldsymbol{I} = I|].$$

Since $|I_0| = r/18$ and $|\boldsymbol{I}| = r/3$, we can estimate the first term by Chernoff inequality as

$$\mathbf{P}\left[|I_0 \cap \boldsymbol{I}| < \frac{r^2}{100m}\right] \leq \exp\left(-\Omega\left(r^2/m\right)\right). \tag{3.12}$$

For estimating the second term, fix any individual I such that $|I| = r/3$ and $|I_0 \cap I| \geq \frac{r^2}{100m}$, and let $\hat{I} \supseteq I$ be a corresponding set of rows satisfying the conclusion of Lemma 3.4.6. We want to estimate $\mathbf{P}[\boldsymbol{\rho}_{\hat{I}}(m) \neq 0]$, where $\boldsymbol{\rho}_{\hat{I}}$ is picked at random from $M_{\hat{I}}$ (thus, $\boldsymbol{\rho}_{\hat{I}}$ is a random variable that results from $\boldsymbol{\rho}$ after revealing $\hat{\boldsymbol{I}} = \hat{I}$).

Let $I'_0 = I_0 \cap \hat{I}$, $I'_0 = \{i_1, \ldots, i_\ell\}$; $\ell \geq r^2/100m$. Since I_0 is minimal with the property (3.11), for every $\nu \in [\ell]$ we can choose $f \in \{f_1, \ldots, f_d\}$ such that $Vars(f) \subseteq X_{i_\nu}(A)$ but $Vars(f) \not\subseteq X_i(A)$ for any other $i \in I_0$. Hence, we can assume w.l.o.g. that $Vars(f_\nu) \subseteq X_{i_\nu}(A)$ for $\nu = 1, \ldots, \ell$.

Now, let $V_0 \stackrel{\text{def}}{=} Span(f_1, \ldots, f_\ell)$ be the $\mathbb{F}_2$-linear space generated by the linear functions $f_1, \ldots, f_\ell$, and let $\hat{V} \stackrel{\text{def}}{=} Span\left(\left\{\bigoplus_{j \in J_i(A)} x_j \,\middle|\, i \in \hat{I}\right\}\right)$. $\mathbf{P}[\boldsymbol{\rho}_{\hat{I}}(m) \neq 0] \leq \mathbf{P}\left[\boldsymbol{\rho}_{\hat{I}}(y_{f_1}^{\epsilon_1} \ldots y_{f_\ell}^{\epsilon_\ell}) \neq 0\right]$, and the latter probability is less or equal than $2^{-(V_0 : V_0 \cap \hat{V})}$ (here $(V_0 : V_0 \cap \hat{V}) \stackrel{\text{def}}{=} \dim(V_0) - \dim(V_0 \cap \hat{V})$ is the standard co-dimension of linear spaces).

To see this, note that $\boldsymbol{\rho}_{\hat{I}}$ gives $\{0,1\}$-values to all variables of $f_1, ..., f_\ell$. Let $k = (V_0 : V_0 \cap \hat{V})$. We can choose k linear forms $f_{i_1}, f_{i_2}, ..., f_{i_k} \in \{f_1, ..., f_\ell\}$ such that the family $f_{i_1}, ..., f_{i_k}$ is linearly independent modulo $\hat{V}$. Then the values $\boldsymbol{\rho}_{\hat{I}}(f_{i_1}), ..., \boldsymbol{\rho}_{\hat{I}}(f_{i_k})$ are independent, each equal 0 with probability 1/2. Thus, the probability that no y_{f_i} is killed is less or equal 2^{-k}.

Clearly, $2^{-(V_0:V_0\cap\hat{V})} = 2^{\dim(V_0\cap\hat{V})-\ell}$. Hence, we only have to upper bound $\dim(V_0 \cap \hat{V})$. Let us denote by $\hat{I}^+$ the set of all rows $i \in \hat{I}$ which appear with coefficient $\alpha_i \neq 0$ in *at least one* sum of the form

$$\bigoplus_{i\in\hat{I}} \alpha_i \cdot \left(\bigoplus_{j\in J_i(A)} x_j \right) \tag{3.13}$$

that happens to belong to V_0, and let $\hat{V}^+ \stackrel{\text{def}}{=} Span\left(\left\{ \bigoplus_{j\in J_i(A)} x_j \,\middle|\, i \in \hat{I}^+ \right\}\right)$. Then $V_0 \cap \hat{V} \subseteq V_0 \cap \hat{V}^+$ by our choice of $\hat{I}^+$, and $\dim(V_0 \cap \hat{V}) \leq \dim(\hat{V}^+) \leq |\hat{I}^+|$.

In order to bound from above $|\hat{I}^+|$, we apply the expansion property to $I'_0 \cup \hat{I}^+$ (its cardinality does not exceed $r/18+2r/3 < r$). We get $|\partial_A(I'_0 \cup \hat{I}^+)| \geq \frac{3}{4}s \cdot |I'_0 \cup \hat{I}^+|$. Note that rows from $\hat{I}^+ \setminus I'_0$ may not contain elements from $\partial_A(I'_0 \cup \hat{I}^+)$ at all; otherwise, the corresponding variable would not cancel out in the sum (3.13), and this would prevent the latter from being in V_0 (note that for any form $f \in V_0$, $Vars(f) \subseteq \bigcup_{i\in I'_0} X_i(A)$).

The key observation is that every row i_ν from $\hat{I}^+ \cap I'_0$ may also contain only a relatively small number of boundary elements, namely, at most $(s/2)$. Indeed, $|Vars(f_\nu)| \leq s/2$ (see Definition 3.4.2). Therefore, if J_{i_ν} would have contained $> s/2$ boundary elements, then at least one boundary variable $x_j \in X_{i_\nu}(A)$ would not belong to $Vars(f_\nu)$, and would once more prevent the sum (3.13) from lying in V_0 (since j belongs to the boundary, x_j may not occur in other forms appearing in this sum).

Summing up the above remarks, we have the upper bound $|\partial_A(I'_0 \cup \hat{I}^+)| \leq s \cdot |I'_0 \setminus \hat{I}^+| + \frac{s}{2} \cdot |I'_0 \cap \hat{I}^+|$. Comparing it with the lower bound given by expansion, we get

$$\frac{3}{4}s \cdot |I'_0 \cup \hat{I}^+| \leq |\partial_A(I'_0 \cup \hat{I}^+)| \leq s \cdot |I'_0 \setminus \hat{I}^+| + \frac{s}{2} \cdot |I'_0 \cap \hat{I}^+|,$$

$$\frac{3}{4}s\left(|\hat{I}^+| + |I'_0 \setminus \hat{I}^+|\right) \leq s \cdot |I'_0 \setminus \hat{I}^+| + \frac{s}{2} \cdot |I'_0 \cap \hat{I}^+|$$

and

$$\frac{3}{4}s|\hat{I}^+| \leq \frac{1}{4}s \cdot |I'_0 \setminus \hat{I}^+| + \frac{s}{2} \cdot |I'_0 \cap \hat{I}^+|,$$

which implies $|\hat{I}^+| \leq \frac{2}{3}|I'_0| = \frac{2\ell}{3}$.

Therefore, $\dim(V_0 \cap \hat{V}) \leq \frac{2\ell}{3}$ and $\mathbf{P}[\boldsymbol{\rho}_{\hat{I}}(m) \neq 0] \leq 2^{-\ell/3} \leq \exp\left(-\Omega\left(r^2/m\right)\right)$. Together with (3.12) this implies $\mathbf{P}[\boldsymbol{\rho}(m) \neq 0] \leq \exp\left(-\Omega\left(r^2/m\right)\right)$. Hence, π must contain at least $\exp\left(\Omega\left(r^2/m\right)\right)$ monomials (of A-degree $\geq r/18$) since otherwise we could find a restriction ρ that kills all of them, contrary to Lemma 3.4.4. The proof of Theorem 3.4.1 is complete.■

3.5 Existence of strong expanders and hard generators

All our hardness results in the previous two sections are based upon the notion of an (r, s, c)-expander. As we noticed in Introduction, one of our eventual goals is to be able to stretch n seed bits to as many output bits m as possible so that the resulting generator is hard for as strong propositional proof systems P as possible. In this section we will see what I/O ratio can we achieve with the results from the two previous sections.

All explicit constructions of (r, s, c)-expanders we know of are based upon Examples 5, 6 from Section 3.2.1. Unfortunately, the resulting expanders turn out to be virtually useless for our purposes since they can not even break the barrier $m = n$. Let us turn instead to a simple probabilistic argument. We note that in the context of proof complexity, there is not that much advantage to having explicit constructions of hard tautologies over existence proofs.

Theorem 3.5.1 *For any parameters s, n there exists an $\left(\Omega(n/s) \cdot n^{-O(1/s)}, s, \frac{3}{4}s\right)$-expander of size $(n^2 \times n)$.*

Proof. Let us construct a random $(n^2 \times n)$ matrix $\boldsymbol{A}$ as follows. For every $i \in [n^2]$, let $J_i(\boldsymbol{A}) \stackrel{\text{def}}{=} \{j_{i1}, \ldots, j_{is}\}$, where all $j_{i\nu}$ $(i \in [n^2], \nu \in [s])$ are picked from $[n]$ independently and at random (in fact, we would also obtain the same result by letting $J_i(\boldsymbol{A})$ be uniformly and independently distributed over all s-subsets of $[n]$, but with our choice of $J_i(\boldsymbol{A})$ calculations become simpler). We wish to show that

$$\mathbf{P}[\boldsymbol{A} \text{ is not an } (r, s, 3s/4)\text{-expander}] < 1,$$

for some $r \geq \Omega(n/s) \cdot n^{-O(1/s)}$. Let p_ℓ be the probability that any given ℓ rows of the matrix $\boldsymbol{A}$ violate the expansion property. Then, clearly,

$$\mathbf{P}[\boldsymbol{A} \text{ is not an } (r, s, 3s/4)\text{-expander}] \leq \sum_{\ell=1}^{r} n^{2\ell} p_\ell. \tag{3.14}$$

Fix an arbitrary I of cardinality $\ell \leq r$. Since every column $j \in \bigcup_{i \in I} J_i(\boldsymbol{A}) \setminus \partial_{\boldsymbol{A}}(I)$ belongs to at least two sets $J_i(\boldsymbol{A})$, we have the bound $\left|\bigcup_{i \in I} J_i(\boldsymbol{A})\right| \leq |\partial_{\boldsymbol{A}}(I)| + \frac{1}{2} \cdot \left(\sum_{i \in I} |J_i(\boldsymbol{A})| - |\partial_{\boldsymbol{A}}(I)|\right) \leq \frac{1}{2}(s\ell + |\partial_{\boldsymbol{A}}(I)|)$. Hence $\partial_{\boldsymbol{A}}(I) < \frac{3}{4}s\ell$ implies also $\left|\bigcup_{i \in I} J_i(\boldsymbol{A})\right| < \frac{7}{8}s\ell$, and p_ℓ can be estimated by the union bound as

$$p_\ell \leq \binom{n}{\frac{7}{8}s\ell} \cdot \left(\frac{7s\ell}{8n}\right)^{s\ell} \leq (O(s\ell/n))^{s\ell/8} \leq (O(sr/n))^{s\ell/8}.$$

Substituting this bound into (3.14), we obtain

$$\mathbf{P}[\boldsymbol{A} \text{ is not an } (r, s, 3s/4)\text{-expander}] \leq \sum_{\ell=1}^{r} n^{2\ell} \cdot \left(O\left(\frac{sr}{n}\right)\right)^{s\ell/8}. \tag{3.15}$$

The sum in the right-hand side is the geometric progression with the base $n^2 \cdot (O(sr/n))^{\Omega(s)}$. Hence, if $r = (\epsilon n/s) \cdot n^{-1/s\epsilon}$ for a sufficiently small $\epsilon > 0$, the right-hand side of (3.15) is less than (1/2) which completes the proof of Theorem 3.5.1.■

Corollary 3.5.2 *There exists a family of $(m \times n)$ matrices $A^{(m,n)}$ such that for every $b = (b_1, \ldots, b_m) \in \{0,1\}^m$, any PCR-refutation of $\tau(A^{(m,n)}, \vec{\Sigma}(A^{(m,n)}, b))$ over an arbitrary field with $char(F) \neq 2$ must have size $\exp\left(\frac{n^{2-O(1/\log\log n)}}{m}\right)$.*

Proof. Since for $m \geq n^2$ the bound becomes trivial, we can assume that $m \leq n^2$. Apply Theorem 3.5.1 with $s = \frac{1}{2}\log_2 \log_2 n$, and cross out in the resulting matrix all rows but (arbitrarily chosen) m. This will result in an $(r, s, \frac{3}{4}s)$-expander $A^{(m,n)}$ of size $(m \times n)$, where $r \geq n^{1-O(1/\log\log n)}$. Now we only have to apply Corollary 3.3.11 and notice that $2^{2^s} = 2^{\sqrt{\log n}} \leq n^{1/\log\log n}$.■

Corollary 3.5.2 shows that in the functional encoding we can stretch n random bits to $n^{2-O(1/\log\log n)}$ bits so that this generator will be hard for (polynomial size) PCR-proofs over an arbitrary field F with $char(F) \neq 2$. In particular, it is hard for Resolution.

Corollary 3.5.3 *There exists a family of $(m \times n)$ matrices $A^{(m,n)}$ such that $\left|J_i\left(A^{(m,n)}\right)\right| \leq \log_2 n$ for all $i \in [m]$ and for every $b = (b_1, \ldots, b_m) \in \{0,1\}^m$ we have the following bounds.*

1. *Let $C_1, \ldots, C_m$ be single-output Boolean circuits over an arbitrary fixed finite basis, where C_i is a circuit of size $O(\log n)$ in the variables $X_i(A^{(m,n)})$ that computes the function $\Sigma_i(A^{(m,n)}, b_i)$. Then every PCR-refutation of $\tau(A^{(m,n)}, \vec{C})$ over an arbitrary field with $char(F) \neq 2$ must have size $\exp\left(\Omega\left(\frac{n^2}{m(\log n)^3}\right)\right)$.*
2. *Every PCR-refutation of $\tau_{\oplus}(A^{(m,n)}, b)$ over an arbitrary field with $char(F) \neq 2$ must have size $\exp\left(\Omega\left(\frac{n^2}{m(\log n)^2}\right)\right)$.*

Proof. Same as the proof of Corollary 3.5.2, only this time we let $s = \log_2 n$. Namely, Theorem 3.5.1 provides us with an $(r, s, \frac{3}{4}s)$-expander for $r \geq \Omega(n/\log n)$. The proof now follows by Corollary 3.3.6 and Theorem 3.4.1.■

Corollary 3.5.3 allows us to construct generators stretching n bits to $m = o(n^2/(\log n)^4)$ bits in the circuit encoding, and to $m = o(n^2/(\log n)^3)$ bits in the linear encoding which are hard for poly-size PCR-proofs in odd characteristic.

3.6 Open problems

Can we reduce the devastating 2^{2^s} factor in our size lower bounds for the functional framework (Corollaries 3.3.5 and 3.3.11)? One way to approach this would be to look for generalizations of the basic Proposition 2.2.1 that would take into account the structure of the variables y_f (which can be originally divided into m large groups).

Find explicit constructions of (r, s, c)-expanders with parameters that would be sufficient for (at least, some of) the applications in the current chapter and in [Raz02b] (as we remarked above, one step in this direction was made in [CRVW02]).

More open problems representing the next generation of tasks faced by this theory can be found in [Raz02b].

Chapter 4

General Hardness Criterium for Polynomial Calculus

4.1 Introduction

The propositional proof systems which recently received much attention are so-called algebraic proof systems simulating the most basic algebraic facts and constructions. The idea to use algebraic machinery in the proof complexity originally appeared in [BIK+94] who defined the Nullstellensatz refutation system motivated by Hilbert's Nullstellensatz. [CEI96] introduced an even more natural algebraic proof system that directly simulates the process of generating an ideal from a finite set of generators, called Polynomial Calculus (PC for short). This system is a potential candidate for automatic theorem provers [CEI96]; thus it seems interesting and important to prove lower bounds for Polynomial Calculus.

Known approaches to the lower bounds on the degree of Polynomial Calculus use the idea of locality (discussed in Section 4.2.2), with one notable exception [Kra01b]. First papers [Raz98, IPS99] devoted to Pigeonhole principle involved also rather technical and specific calculations (pigeon dance).

Recently [Gri98] came up with a simple idea how to avoid completely such calculations and prove $\Omega(n)$ degree lower bounds by using Tseitin tautologies. The original

proof in [Gri98] embraced only Nullstellensatz proof system, then it was generalized to PC in [BGIP01] and further developed in [BI99, Gri01, ABSRW00]. One drawback of this idea is that it essentially uses the representation with binomial ideals and can be applied only to binomial functions as the base functions. But there are very few binomials; if we insist on Boolean relations $x^2 - x = 0$, we have only PARITY functions, and in characteristic 2 there are no binomials at all (this restriction can be sometimes circumvented by using low degree reductions [BGIP01] but not always, see for example the case of random k-CNF in characteristic 2 in Section 4.4.3 below).

This situation is in a sharp contrast with the situation for Resolution where in Chapter 3 we gave a hardness criterion (robustness) satisfied by a random function and showed that the induced tautologies are hard provided the underlying structure has sufficient expansion. The ideology there is similar to that of Natural Proofs in [RR97]: every lower bound proof which works for a single function must also work for a large class of functions specified by a constructive combinatorial property.

In this chapter we fill this gap by giving a hardness criterion (immunity) and proving linear lower bounds for PC refutations of wide class of tautologies based on immune functions. It is worth noting that over fields of positive characteristic p immunity coincides, up to negation, with the notion of weak MOD_p-degree introduced (for not necessarily prime p) in [Gre00] as an integral part of attempt to understand the *computational* power of multi-linear polynomials.

As some applications of our results, we consider mod_p Tseitin tautologies from [BGIP01] in the Boolean framework (i.e., when our ideal contains the axioms $x_i^2 = x_i$) and prove their hardness over fields of characteristic different from p. Next we introduce the analog of Tseitin tautologies in characteristic 0 (called Flow tautologies) and show that they are hard over any field. The most important impact of our approach, however, is that we can work directly with the field $\mathbb{F}_2$ which is the most interesting case. In particular, we can do random k-CNF over this field, thus we prove the conjecture from [BI99].

Also, we consider the Pigeonhole principle and prove a hardness result for its

version $EPHP_n^m$ introduced in [BW99]. This result follows from [Raz98] but our proof is conceptually simpler since it does not use the technique of pigeon dance at all. At the end we show a weak relation between robust functions from Chapter 3 and immune polynomials in characteristic zero which allows us to get some lower bounds for the *polynomial calculus* (also in characteristic 0) based on the *robustness* of underlying functions.

The chapter is organized as follows. Section 4.2 contains the necessary definitions and the intuition of the lower bound technic based on locality. We prove our main hardness results in Section 4.3 and show the implied lower bounds in Section 4.4. Finally we present some open questions in Section 4.5.

4.2 Preliminaries

Fix an arbitrary field $\mathbb{F}$. We will be working in the $\mathbb{F}$-algebra $S_n(\mathbb{F})$ which results from factoring the polynomial ring $\mathbb{F}[x_1, \ldots, x_n]$ by the ideal generated by the relations $x_i^2 - x_i$ $(1 \leq i \leq n)$. Every element $f \in S_n(\mathbb{F})$ has a unique representation as a multi-linear polynomial (which determines its *degree* $\deg(f)$), and a unique representation as a $\mathbb{F}$-valued function on $\{0,1\}^n$. We will be alternately exploiting both representations. All polynomials considered in this chapter (unless the opposite is stated explicitly) will be multi-linear so we sometimes omit this word.

For a polynomial f, $Vars(f)$ will denote the set of its essential variables. An *assignment to* f is a mapping $\alpha : Vars(f) \to \{0,1\}$.

For historical reasons, when studying a system of algebraic equations one is interested in the set of its *roots*. Thus an assignment α *satisfies* a polynomial f iff α is the root of f. Accordingly, every Boolean function g uniquely defines the multi-linear polynomial p_g which is equal to 0 on the assignment α if $g(\alpha) = 1$ and to 1 if $g(\alpha) = 0$.

If $\vec{v} = \{v_1, \ldots, v_k\}$ is a tuple of variables, and $\vec{\epsilon} \in \{0,1\}^k$ then by $\chi_{\vec{\epsilon}}(\vec{v})$ we denote the multi-linear polynomial which is equal to 1 if $\vec{v} = \vec{\epsilon}$ and to 0 otherwise (formally, $\chi_{\vec{\epsilon}}(\vec{v}) \stackrel{\text{def}}{=} \prod_{i \in [k]}(1 - v_i - \epsilon_i + 2v_i\epsilon_i)$). The following identity is obvious but extremely

useful:

$$f = \sum_{\vec{\epsilon} \in \{0,1\}^k} \chi_{\vec{\epsilon}}(\vec{v}) \cdot (f|_{\vec{v}=\vec{\epsilon}}) \tag{4.1}$$

($\vec{v} = \vec{\epsilon}$ is the restriction which assigns all v_i to ϵ_i and leaves all other variables unassigned).

Let $Span(f_1, \ldots, f_k)$ be the ideal generated by $f_1, \ldots, f_k$. We say that $f_1, \ldots, f_k$ *semantically imply* g (denoted $f_1, \ldots, f_k \models g$), if g equals zero on the set of roots of ideal $Span(f_1, ..., f_k)$ (i.e. $\forall \alpha \in \{0,1\}^V (f_1(\alpha) = \cdots = f_k(\alpha) = 0 \Rightarrow g(\alpha) = 0)$, where $V = Vars(f_1) \cup \ldots \cup Vars(f_k) \cup Vars(g)$).

Denote by T_n the set of all *multi-linear terms*, i.e., products of the form $x_{i_1} x_{i_2} \ldots x_{i_d}$ with $1 \leq i_1 < i_2 < \ldots < i_d \leq n$. The *degree* $\deg(t)$ *of a term* t is the number of variables occurring in it. Let $T_{n,d} \stackrel{\text{def}}{=} \{t \in T_n \mid \deg(t) \leq d\}$, and $S_{n,d}(\mathbb{F}) \stackrel{\text{def}}{=} \mathbb{F}T_{n,d}$ be the linear space of all multi-linear polynomials of degree at most d. We write $t \in f$ if a term t is contained in polynomial f with non-zero coefficient.

4.2.1 Tautologies induced by Nisan-Wigderson generator

In this section we define the general structure of our tautologies. We use the notion of expander matrix introduced in Section 3.2.1.

Let A be an $(m \times n)$ 0-1 matrix, which is (r, s, c)-expander and $J_i(A) \stackrel{\text{def}}{=} \{j \in [n] \mid a_{ij} = 1\}$.

Let $X_i(A) \stackrel{\text{def}}{=} \{x_j \mid j \in J_i(A)\}$ and $f_1(x_1, \ldots, x_n), \ldots, f_m(x_1, \ldots, x_n)$ be polynomials such that $Vars(f_i) \subseteq X_i(A)$. We will be interested in the system of equations

$$\begin{cases} f_1(x_1, \ldots, x_n) = 0 \\ \qquad \ldots \\ f_m(x_1, \ldots, x_n) = 0. \end{cases} \tag{4.2}$$

It was proved in Section 3 that if (the characteristic functions of the set of roots of) f_i's posses certain hardness property called robustness and A is a sufficiently good expander, then the system (4.2) is hard for Resolution. Based upon the machinery developed in [Gri98, BGIP01], we also showed that the system (4.2) is hard for Poly-

nomial Calculus when f_i's correspond to the PARITY functions and char $\mathbb{F} \neq 2$. In this chapter we introduce a *general* hardness condition on f_i which will imply that this system is hard for Polynomial Calculus.

Definition 4.2.1 A polynomial f is called *ℓ-immune* iff for any non-zero polynomial g, $f \models g$ implies $\deg(g) \geq \ell$. A Boolean function g is *ℓ-immune* over a field if its associated polynomial p_g over this field is ℓ-immune.

This definition says that a polynomial is hard iff it has no non-trivial semantic corollaries of small degree. Clearly, a polynomial is ℓ-immune if and only if the characteristic function of the set of its roots is so.

Let us now relate immunity over fields of positive characteristic p with the following notion:

Definition 4.2.2 ([Gre00]) The *weak MOD_p-degree* of a Boolean function $g(x_1, \ldots, x_n)$ is the smallest degree of a non-zero multi-linear polynomial $f(x_1, \ldots, x_n)$ with integer coefficients such that $g(\alpha) = 0 \Rightarrow f(\alpha) = 0 \bmod p$, for all $\alpha \in \{0,1\}^n$.

We need one elementary lemma.

Lemma 4.2.3 *A polynomial $f(x_1, \ldots, x_n)$ over a field $\mathbb{F}$ is ℓ-immune if and only if for any non-zero multi-linear polynomial $g(x_1, \ldots, x_n)$* **with integer coefficients** *$f \models g$ implies $\deg(g) \geq \ell$.*

Proof. Part "only if" is obvious. For another direction, suppose for the sake of contradiction that $g_0 \in S_n(\mathbb{F})$, $f \models g_0$ and $\deg(g_0) < \ell$. We want to find an *integer* polynomial g with the same properties.

Introduce $\mathbb{F}$-valued variables g_t, where $t \in T_n$, corresponding to the (unknown) coefficients of g. The condition $f \models g$ is equivalent to the system of linear equations $\{\sum_{t \in T_n} g_t t(\alpha) = 0 \mid f(\alpha) = 0\}$ over g_t with integer coefficients. Let us add the conditions $g_t = 0$ for all terms t with $\deg(t) \geq \ell$ and $g_{t_0} \neq 0$ for an arbitrary term t_0 appearing in g.

We defined the system of uniform linear equations and inequalities with integer coefficients which has a solution in $\mathbb{F}$. It follows from the basics of algebra that it also has a solution over integers, and this solution defines the desired g.■

Corollary 4.2.4 *1. A Boolean function is ℓ-immune over a field $\mathbb{F}$ with* $\mathrm{char}\,\mathbb{F} = p > 0$ *if and only if the weak MOD_p-degree of its negation is at least ℓ.*

2. A Boolean function $g(x_1, \ldots, x_n)$ is ℓ-immune over a field $\mathbb{F}$ with $\mathrm{char}\,\mathbb{F} = 0$ *if and only if the smallest degree of a non-zero multi-linear polynomial $f(x_1, \ldots, x_n)$ with integer coefficients such that $\forall \alpha \in \{0,1\}^n (g(\alpha) = 1 \Rightarrow f(\alpha) = 0)$ is at least ℓ.*

Thus, in positive characteristic immunity is the same (up to negation) as weak degree, and the main reason why we introduced this new terminology is because the usage of the term "weak degree" in characteristic 0 (see [ABFR94]) is totally incompatible with our purposes.

The following simple lemma shows that the notion of immunity behaves well with respect to restrictions.

Lemma 4.2.5 *1. If f is ℓ-immune then, for any restriction ρ, $f|_\rho$ is $(\ell - |\rho|)$-immune.*

2. Let $V \subseteq Vars(f)$, and assume that for every restriction ρ with $\rho^{-1}(\star) = V$, $f|_\rho$ is ℓ-immune. Then f is ℓ-immune.

Proof. **Part 1)** Suppose that ρ assigns $v_1, \ldots, v_k$ to $\epsilon_1, \ldots, \epsilon_k$, and assume the contrary, that is $f|_{\vec{v}=\vec{\epsilon}} \models g$, where $g \neq 0$ and $\deg(g) < \ell - k$. We can assume w.l.o.g. that $Vars(g) \cap \{v_1, \ldots, v_k\} = \emptyset$. Then, clearly, $f \models \chi_{\vec{\epsilon}}(\vec{v}) \cdot g$ and $\chi_{\vec{\epsilon}}(\vec{v}) \cdot g \neq 0$, a contradiction.

Part 2) Assume the contrary, that is $f \models g$, where $g \neq 0$ and $\deg(g) < \ell$. Pick up an arbitrary restriction ρ with $\rho^{-1}(\star) = V$ and such that $g|_\rho \neq 0$. Then $f|_\rho \models g|_\rho$, contrary to the assumption that $f|_\rho$ is ℓ-immune.■

4.2.2 Local strategy for PC lower bounds

Now we briefly describe how the idea of locality can help in proving lower bounds on the degree of PC refutation. First, let us recall some standard notions from commutative algebra (adapted to the special case of the ring $S_n(\mathbb{F})$).

Definition 4.2.6 An ordering $\preceq$ of T_n is *admissible* if:

1. $\forall t_1, t_2 \in T_n(\deg(t_1) < \deg(t_2) \Rightarrow t_1 \prec t_2)$.

2. If $t_1 \preceq t_2$ and $t \in T_n$ does not contain any variables from t_1, t_2, then $tt_1 \preceq tt_2$.

Fix an admissible ordering $\preceq$ on T_n. For $f \in S_n(\mathbb{F})$, $LT(f) \in T_n$ is the *leading term* of f w.r.t. $\preceq$. Part 1 of Definition 4.2.6 implies that $\deg(LT(f)) = \deg(f)$.

Definition 4.2.7 For an ideal V the term t is called *reducible* mod V (and with respect to some admissible ordering $\preceq$) if V contains some polynomial f such that $LT(f) = t$. The set of irreducible terms Δ is linearly independent modulo V and the algebra $S_n(\mathbb{F})$ can be represented as the direct sum

$$S_n(\mathbb{F}) = \mathbb{F}\Delta \oplus V.$$

The operator of projection onto the first coordinate, called *reduction operator* (and denoted R_V) maps each term t to the unique polynomial $R_V(t) \in \mathbb{F}\Delta$ such that $t - R_V(t) \in V$.

[CEI96] were the first to consider these classical notions in the case when V is a pseudo-ideal, i.e. not necessarily closed under the multiplication rule. This leads to the following chain of definitions.

For $f_1, \ldots, f_m \in S_{n,d}(\mathbb{F})$, denote by $V_{n,d}(f_1, \ldots, f_m)$ the set of all $g \in S_{n,d}(\mathbb{F})$ that are provable from $f_1, \ldots, f_m$ by a polynomial calculus proof of degree at most d. Due to the presence of the addition rule, $V_{n,d}(f_1, \ldots, f_m)$ is a linear subspace in $S_{n,d}(\mathbb{F})$. A term $t \in T_{n,d}$ is called *reducible* if $t = LT(f)$ for some $f \in V_{n,d}(f_1, \ldots, f_m)$, and *irreducible* otherwise. Denote by $\Delta_{n,d}(f_1, \ldots, f_m)$ the set of all irreducible terms

in $T_{n,d}$. Terms from $\Delta_{n,d}(f_1, \ldots, f_m)$ are linearly independent modulo $V_{n,d}(f_1, \ldots, f_m)$, and analogously with the classical case we have the representation

$$S_{n,d}(\mathbb{F}) = \mathbb{F}\Delta_{n,d}(f_1, \ldots, f_m) \oplus V_{n,d}(f_1, \ldots, f_m) \tag{4.3}$$

of $S_{n,d}(\mathbb{F})$ as the direct sum. Denote the projection onto the first coordinate (also called *reduction operator*) by $R_{n,d,f_1,\ldots,f_m}$.

In order to prove that $1 \notin V_{n,d}(f_1, ..., f_m)$ one has to show that

$$R_{n,d,f_1,\ldots,f_m}(1) \neq 0.$$

For doing that it suffices to produce a non-trivial linear operator R on $S_{n,d}(\mathbb{F})$ which is stronger than $R_{n,d,f_1,\ldots,f_m}$ in the sense that

$$\mathrm{Ker}(R_{n,d,f_1,\ldots,f_m}) \subseteq \mathrm{Ker}(R)$$

and show that $\mathrm{Ker}(R) \neq S_{n,d}(\mathbb{F})$. The following lemma states what need be checked for that.

Lemma 4.2.8 ([Raz98]) *Suppose that $f_1, ..., f_m$ are axioms, and $d < n$. If there exist a linear operator $R \not\equiv 0$ on $S_{n,d}(\mathbb{F})$ such that:*

1) $\forall i\ R(f_i) = 0$;

2) $\forall t, x_j\ (\deg(t) < d \rightarrow R(x_j \cdot t) = R(x_j \cdot R(t)))$

then there is no PC refutation of $\{f_1, ..., f_m\}$ with degree less or equal than d.

[Raz98] proposed to construct the operator R from the previous lemma locally:

$$R(t) \stackrel{\text{def}}{=} R_{V(t)}(t), \tag{4.4}$$

where $R_{V(t)}$ is the classical reduction operator of the ordinary ideal $V(t)$ generated by some "small" subset of axioms dependent on t. The advantage we gain in this way

is that the structure of classical reduction operators is much better understood, and, unlike their counterparts for pseudo-ideals, they can be also studied by semantical means.

If R is any operator satisfying assumptions of Lemma 4.2.8, then every polynomial $f = \sum_i a_i t_i$ derivable from $\{f_1, \ldots, f_m\}$ in degree $\leq d$ can be alternatively represented as the sum

$$f = \sum_i a_i(t_i - R(t_i)).$$

If, moreover, $R(t)$ is given by (4.4), then t_i's are leading terms in $(t_i - R(t_i))$, there is no cancellation between them, and each polynomial $t_i - R(t_i)$ is a corollary of a "small" number of axioms. Thus the idea of locality says that any polynomial in $V_{n,d}(f_1, ..., f_m)$ can be represented by the sum of corollaries of small number of axioms without leading terms cancellation or, informally, *everything we can infer in small degree we can also infer locally.*

4.3 Main results

In this section we prove that for any (r, s, c)-expander A and ℓ-immune f_i's any PC refutation of the system (4.2) has degree greater than $r(\ell/4-(s-c))$ (Theorem 4.3.8). This bound presumes that $\ell > 4(s - c)$ and thus it is not applicable to expanders with small constant s and c. We managed to strengthen the degree lower bound to $rc/2$ in the case when f_i's have maximal immunity s (Theorem 4.3.13). This bound will allow us to estimate the hardness of refutation of the random k-CNF for small k in the fields of characteristic 2 (Section 4.4.3).

The heart of our proof is the following theorem.

Theorem 4.3.1 *Suppose that $\vec{y}, \vec{v}, \vec{z}$ is a partition of $\{x_1, ..., x_n\}$; $\vec{P} = \vec{P}(\vec{y}, \vec{v})$, $Q = Q(\vec{v}, \vec{z})$ are polynomials over $\vec{y} \cup \vec{v}$, $\vec{v} \cup \vec{z}$ respectively, where Q is $(2|\vec{v}| + 1)$-immune. Suppose that a term $t(\vec{y}, \vec{v})$ free of z-variables is reducible mod $Span(\vec{P}, Q)$ w.r.t some fixed admissible ordering. Then t is reducible mod $Span(\vec{P})$ w.r.t. the same ordering.*

Proof. The naive idea would be to apply an arbitrary homomorphism of the form ρ :

$\vec{z} \to \vec{f}(\vec{v})$ which kills Q to 0 and therefore has the property $t = \rho(R_{Span(\vec{P},Q)}(t)) \bmod Span(\vec{P})$ (such a homomorphism always exists since Q is $(|\vec{v}|+1)$-immune and hence none of $\chi_{\vec{\epsilon}}(\vec{v})$ in $\sum \chi_{\vec{\epsilon}}(\vec{v})Q|_{\vec{v}=\vec{\epsilon}} = Q$ is its corollary). This does not work in general because the degree of some terms in $R_{Span(\vec{P},Q)}(t)$ may increase under ρ and become more than $\deg(t)$. Still as we show below this idea suffices in the partial case when Q has the maximal immunity (Lemma 4.3.14). But in the general case we have to use more complicated methods.

In Definition 4.3.2 and the following chain of lemmas we assume that $\vec{y}, \vec{v}, \vec{z}, \vec{P}, Q$ satisfy assumptions of Theorem 4.3.1, and all reduction operators are taken w.r.t. some fixed admissible ordering $\preceq$. Let $k \stackrel{\text{def}}{=} |\vec{v}|$.

Definition 4.3.2 (Operator R^Q) The linear operator R^Q is defined on T_n in the following way:

$$R^Q(t) \stackrel{\text{def}}{=} \sum_{\vec{\epsilon} \in \{0,1\}^k} \chi_{\vec{\epsilon}}(\vec{v}) \cdot R_{Span(Q|_{\vec{v}=\vec{\epsilon}})}(t|_{\vec{v}=\vec{\epsilon}}),$$

and extended to $S_n(\mathbb{F})$ by linearity.

The intuitive meaning of the operator R^Q is that it tries to reduce z-degree of the term t locally and independently within the subcubes specified by all possible assignments to $\vec{v}$. It ignores y-variables (since they do not appear in Q), but it can in general increase the number of v-variables.

Lemma 4.3.3 *For any polynomial f, $f = R^Q(f) \bmod Span(Q)$.*

Proof. Due to linearity, we only need to prove that $Q \models t - R^Q(t)$ for every term t. This semantic inference holds if and only if for any tuple $\vec{\epsilon}$ the polynomial $Q|_{\vec{v}=\vec{\epsilon}}$ implies

$$(t - R^Q(t))|_{\vec{v}=\vec{\epsilon}} = (t|_{\vec{v}=\vec{\epsilon}} - R_{Span(Q|_{\vec{v}=\vec{\epsilon}})}(t|_{\vec{v}=\vec{\epsilon}})),$$

and the latter clearly takes place.■

Lemma 4.3.4 *If $|Vars(t) \cap \vec{z}| \leq k$ then $R^Q(t) = t$.*

Proof.

We need to check that two polynomials t and $R^Q(t)$ are equal. For this it is sufficient to check the equality of $t|_{\vec{v}=\vec{\epsilon}}$ and $R^Q(t)|_{\vec{v}=\vec{\epsilon}} = R_{Span(Q|_{\vec{v}=\vec{\epsilon}})}(t|_{\vec{v}=\vec{\epsilon}})$ for each tuple $\vec{\epsilon} \in \{0,1\}^k$. But since Q is $(2k+1)$-immune, $Q|_{\vec{v}=\vec{\epsilon}}$ is $(k+1)$-immune by Lemma 4.2.5(1); therefore the term $t|_{\vec{v}=\vec{\epsilon}}$ of degree $\leq k$ is irreducible mod $Span(Q|_{\vec{v}=\vec{\epsilon}})$.■

The following lemma is the heart of the whole argument.

Lemma 4.3.5 *Suppose that f is a polynomial and $\vec{P}, Q \models f$. Then $\vec{P} \models R^Q(f)$.*

Proof. As usual, we only have to prove that for any tuple $\vec{\epsilon}$, $\vec{P}|_{\vec{v}=\vec{\epsilon}} \models R^Q(f)|_{\vec{v}=\vec{\epsilon}}$. Since $\vec{P}, Q \models f$, by Lemma 4.3.3 we have $\vec{P}, Q \models R^Q(f)$, hence $\vec{P}|_{\vec{v}=\vec{\epsilon}}, Q|_{\vec{v}=\vec{\epsilon}} \models R^Q(f)|_{\vec{v}=\vec{\epsilon}}$.

Fix an arbitrary tuple $\vec{\delta}$ for $\vec{y}$ such that $\vec{P}|_{\vec{v}=\vec{\epsilon}}(\vec{\delta}) = 0$. We have

$$Q|_{\vec{v}=\vec{\epsilon}} \models R^Q(f)|_{\vec{v}=\vec{\epsilon},\vec{y}=\vec{\delta}}, \tag{4.5}$$

and we have to show that

$$R^Q(f)|_{\vec{v}=\vec{\epsilon},\vec{y}=\vec{\delta}}$$

is actually equal to 0. But $R^Q(f)|_{\vec{v}=\vec{\epsilon},\vec{y}=\vec{\delta}} = R_{Span(Q|_{\vec{v}=\vec{\epsilon}})}(f|_{\vec{v}=\vec{\epsilon}})|_{\vec{y}=\vec{\delta}}$. Since all terms in $R_{Span(Q|_{\vec{v}=\vec{\epsilon}})}(f|_{\vec{v}=\vec{\epsilon}})$ are irreducible mod $Span(Q|_{\vec{v}=\vec{\epsilon}})$, and the set of irreducible terms is closed downward, the same is true for

$$R_{Span(Q|_{\vec{v}=\vec{\epsilon}})}(f|_{\vec{v}=\vec{\epsilon}})|_{\vec{y}=\vec{\delta}}.$$

Along with (4.5) this implies $R^Q(f)|_{\vec{v}=\vec{\epsilon},\vec{y}=\vec{\delta}} = 0$ and completes the proof of Lemma 4.3.5.■

Lemma 4.3.6 *Suppose that $\vec{P} \models f_0 + \sum_{z_i \in \vec{z}} z_i f_i$, where f_0 is free of z-variables. Then $\vec{P} \models f_0$.*

Proof. Apply the restriction which maps all z_i to 0.■

Let us now finish the proof of Theorem 4.3.1. Let $f \stackrel{\text{def}}{=} R_{Span(\vec{P},Q)}(t)$ and $f' \stackrel{\text{def}}{=} R^Q(f)$.

By Lemma 4.3.5, $\vec{P} \models R^Q(t-f)$, and by Lemma 4.3.4, $R^Q(t) = t$. Thus, $\vec{P} \models (t-f')$. Our goal is to show that all terms in f' are either less than t or contain some z-variables, after that we can apply Lemma 4.3.6 and show that t can be reduced mod $Span(\vec{P})$.

Let us divide the terms in f into two groups:

$$\begin{aligned} G_1 &= \{t_1 \,|\, |Vars(t) \cap \vec{z}| \leq k\} \\ G_2 &= \{t_1 \,|\, |Vars(t) \cap \vec{z}| > k\}. \end{aligned}$$

Consider some monomial $t'_1 \in f'$ such that $|Vars(t'_1) \cap \vec{z}| = \emptyset$. Clearly, $t'_1 \in R^Q(t_1)$ for some term $t_1 \in f$. If $t_1 \in G_1$ then, since G_1 is invariant under R^Q by Lemma 4.3.4, $t'_1 = t_1 \prec t$. Assume that $t_1 \in G_2$. Then $\deg(t'_1) = |Vars(t'_1) \cap \vec{v}| + |Vars(t'_1) \cap \vec{y}| \leq k + |Vars(t'_1) \cap \vec{y}| = k + |Vars(t_1) \cap \vec{y}| < |Vars(t_1) \cap \vec{y}| + |Vars(t_1) \cap \vec{z}| \leq \deg(t_1) \leq \deg(t)$.

Thus all monomials in f' which are free of z-variables either are contained in f or have degree less than $\deg(t)$. Hence Lemma 4.3.6 implies that t is reducible by $\vec{P}$. Theorem 4.3.1 is proved.■

Corollary 4.3.7 *Under the assumptions of Theorem 4.3.1,*

$$R_{Span(\vec{P},Q)}(t) = R_{Span(\vec{P})}(t).$$

Proof. Obviously, all terms in $R_{Span(\vec{P})}(t)$ are free of z-variables. Therefore, by Theorem 4.3.1 all of them are irreducible also mod $Span(\vec{P}, Q)$.■

Theorem 4.3.8 *Assume that A is an (r, s, c)-expander, ℓ is an integer such that $s \leq r(\ell/4 - (s-c))$ and $f_1, \ldots, f_m$ are ℓ-immune polynomials over an arbitrary field such that $Vars(f_i) \subseteq X_i(A)$. Then any PC refutation of the system $f_1 = \ldots = f_m = 0$ has degree greater than $r(\ell/4 - (s-c))$.*

Proof. The idea of the proof is to construct a linear operator R which behaves locally like the reduction operator modulo the corresponding ideal, and use Lemma 4.2.8

after that. To describe R it is sufficient to define it on the set of terms t with $\deg(t) \leq r(\ell/4 - (s-c))$. First we need to give some more notation.

Definition 4.3.9 Assume that A is an $(m \times n)$-matrix and $f_1, ..., f_m$ are polynomials with $Vars(f_i) \subseteq X_i(A)$. For a term t denote by $J(t)$ the index set $\{j | t \text{ contains } x_j\}$. For a set of rows $I \subseteq [m]$ let $Span(I) \stackrel{\text{def}}{=} Span(\{f_i | i \in I\})$.

Now we are ready to define our linear operator R. Fix $A, f_1, \ldots, f_m, r, s, c, \ell$ satisfying assumptions of Theorem 4.3.8. For a term t we define a set of axioms $Sup(t) \subseteq [m]$ and then reduce t mod $Span(Sup(t))$:

Definition 4.3.10 For a term t define the following inference relation $\vdash_t$ on the set $[m]$ of rows of A:

$$I \vdash_t i \equiv \left| J_i(A) \cap \left[\bigcup_{i' \in I} J_{i'}(A) \cup J(t) \right] \right| > \frac{\ell}{4}. \tag{4.6}$$

Let the *support* $Sup(t)$ of t be the set of all rows which can be inferred via $\vdash_t$ from the empty set. Define $R(t) \stackrel{\text{def}}{=} R_{Span(Sup(t))}(t)$.

The rest of the proof is devoted to checking that R satisfies conditions 1) and 2) of Lemma 4.2.8. First we need to estimate the cardinality of $Sup(t)$.

Lemma 4.3.11 *For a term t with $\deg(t) \leq r(\ell/4 - (s-c))$, $|Sup(t)| < r$.*

Proof. Assume the contrary. Then there exists a set $I = \{i_1, \ldots, i_r\}$ of distinct rows such that $\{i_1, \ldots, i_{\nu-1}\} \vdash_t i_\nu$ $(1 \leq \nu \leq r)$. By Definition 3.2.1 it has at least cr boundary elements. By (4.6), each row $i \in I$ has strictly less than $s - \ell/4$ boundary elements not contained in $J(t)$. Thus, $J(t)$ has more than $cr - r(s - \ell/4) = r(\ell/4 - (s-c))$ elements. We got contradiction, which proves Lemma 4.3.11.■

Lemma 4.3.12 *Assume that $s - c < \ell/4$, t is a term and I is a set of rows such that $I \supseteq Sup(t)$ and $|I| \leq r$. Then*

$$R_{Span(I)}(t) = R_{Span(Sup(t))}(t).$$

Proof. Let us apply the expansion property to the set $I \setminus Sup(t)$. It will yield a row $i \in I \setminus Sup(t)$ with at least c boundary elements. In other words,

$$J_i(A) \cap [\cup_{i' \in I \setminus (Sup(t) \cup \{i\})} J_{i'}(A)] \leq s - c.$$

Also, since $Sup(t)$ is closed under $\vdash_t$, we have

$$J_i(A) \cap [\cup_{i' \in Sup(t)} J_{i'}(A) \cup J(t)] \leq \ell/4.$$

Altogether it implies that

$$J_i(A) \cap [\cup_{i' \in I \setminus \{i\}} J_{i'}(A) \cup J(t)] \leq (s - c) + \ell/4 < \ell/2.$$

Let us now set in Theorem 4.3.1

$$\begin{aligned}
\vec{y} &:= \{x_1, \ldots, x_n\} \setminus J_i(A) \\
\vec{v} &:= J_i(A) \cap [\cup_{i' \in I \setminus \{i\}} J_{i'}(A) \cup J(t)] \\
\vec{z} &:= J_i(A) \setminus [\cup_{i' \in I \setminus \{i\}} J_{i'}(A) \cup J(t)] \\
\vec{P} &:= \{ f_{i'} \mid i' \in I \setminus \{i\} \} \\
Q &:= f_i,
\end{aligned}$$

and apply in this situation its Corollary 4.3.7. We conclude that $R_{Span(I)}(t) = R_{Span(I \setminus \{i\})}(t)$. We continue this elimination process until we descend to $R_{Sup(t)}(t)$.■

Now we finish the proof of Theorem 4.3.8. We have produced an operator R on T_n, and we consider its restriction on $T_{n,r(\ell/4-(s-c))}$. Let us check that it indeed satisfies the conditions of Lemma 4.2.8.

To see that $R(f_i) = 0$ for each axiom f_i, let $t_i \stackrel{\text{def}}{=} \prod X_i(A)$. Recall that $\deg(t_i) \leq s \leq r(\ell/4 - (s - c))$. Thus $|Sup(t_i)| < r$ (by Lemma 4.3.11) and for any term t in f_i clearly $Sup(t) \subseteq Sup(t_i)$. Next, $i \in Sup(t_i)$. By Lemma 4.3.12, $R(f_i) = R_{Sup(t_i)}(f_i) = 0$. Thus 1) is satisfied.

To check the second condition, consider a term t with $\deg(t) \leq r(\ell/4 - (s - c) - 1$ and a variable x_j. By Lemma 4.3.11, $|Sup(x_j t)| < r$. By Lemma 4.3.12, $R_{Sup(t)}(t) = R_{Sup(x_j t)}(t)$. For any term $t' \in R_{Sup(t)}(t)$, $Sup(x_j t') \subseteq Sup(x_j t)$. To see this, it is sufficient to notice that $J(t') \subseteq \bigcup_{i' \in Sup(t)} J_{i'}(A) \cup J(t)$. Thus $R_{Sup(x_j t')}(x_j t') = R_{Sup(x_j t)}(x_j t')$ and

$$R(x_j R(t)) = R_{Sup(x_j t)}(x_j R_{Sup(x_j t)}(t)) = R_{Sup(x_j t)}(x_j t),$$

where the last equality follows from the fact that $x_j R_{Sup(x_j t)}(t)$ and $x_j t$ are equal modulo the ideal $Span(Sup(x_j t))$.

Finally, $|J_i(A)| \geq \ell$ for every $i \in [m]$ (since f_i is ℓ-immune), hence $Sup(1) = \emptyset$ and $R(1) = 1$.

Theorem 4.3.8 is proved.■

The bound proved in Theorem 4.3.8 is not applicable in the case when c is small (say, $c < 1$). We will see in Section 4.4.1 that for sufficiently large constant s "good expanders" (that is, with c close to s) do exist but for small s the question about the hardness of the system (4.2) remains open even for random matrices. When c is small, we succeeded in proving lower bounds only in the partial case, when all f_i have the maximal immunity $\ell = s$. It is easy to see that the class of polynomials with maximal immunity consists exactly of polynomials having the form $\alpha \cdot \chi_{\vec{\epsilon}}(\vec{v})$, $\alpha \in \mathbb{F}^*$.

Theorem 4.3.13 *Assume that A is an (r, s, c)-expander and let $\vec{\epsilon}^{(i)} \in \{0,1\}^{X_i(A)}$ $(i \in [m])$. Then any PC refutation of the system $\{\chi_{\vec{\epsilon}^{(i)}}(X_i(A)) \mid i \in [m]\}$ has degree greater than $(rc/2)$.*

Proof. Analogous to the proof of Theorem 4.3.8, but in this partial case the statement of Theorem 4.3.1 can be strengthened while the proof becomes trivial:

Lemma 4.3.14 *Suppose that $\vec{y}, \vec{v}, \vec{z}$ is a partition of $\{x_1, ..., x_n\}$; $\vec{P} = \vec{P}(\vec{y}, \vec{v})$, $Q = Q(\vec{v}, \vec{z})$ are polynomials over $\vec{y} \cup \vec{v}$, $\vec{v} \cup \vec{z}$ respectively, where Q is divisible by $(z - \epsilon)$ for*

some $z \in \vec{z}$, $\epsilon \in \{0,1\}$. Suppose that a term $t(\vec{y},\vec{v})$ free of z-variables is reducible mod $Span(\vec{P},Q)$ w.r.t some fixed admissible ordering. Then t is reducible mod $Span(\vec{P})$.

Proof. Consider any polynomial f s.t. $\vec{P},Q \models f$ and $t = LT(f)$. Applying the restriction $z = \epsilon$, we obtain $\vec{P}|_{z=\epsilon}, Q|_{z=\epsilon} \models f|_{z=\epsilon}$. Since $\vec{P}$ does not depend on z and $Q|_{z=\epsilon} = 0$, $\vec{P} \models f|_{z=\epsilon}$, and clearly $t = LT(f|_{z=\epsilon})$.■

Now we build our operator R in the same way as in Definition 4.3.10, but this time we use another inference relation (notice that this relation infers a *set* of rows at a time rather than a single row):

Definition 4.3.15 For a term t define the following inference relation $\vdash_t$ on the set $[m]$ of rows of A:

$$I \vdash_t I_1 \equiv |I_1| \leq r/2 \ \wedge \ \partial_A(I_1) \subseteq \left[\bigcup_{i \in I} J_i(A) \cup J(t)\right]. \tag{4.7}$$

Let the *support* $Sup(t)$ of t be the set of all rows which can be inferred via $\vdash_t$ from the empty set, and $R(t) \stackrel{\text{def}}{=} R_{Span(Sup(t))}(t)$.

Lemma 4.3.16 *For a term t with $\deg(t) \leq (cr/2)$, $|Sup(t)| \leq r/2$.*

Proof. Assume the contrary, and choose a chain of subsets $I_1, \ldots, I_2, \ldots$ such that $I_1 \cup \ldots \cup I_{\nu-1} \vdash_t I_\nu$ and $|\cup_\nu I_\nu| > r/2$. Let k be the *smallest* index for which $\left|\bigcup_{\nu=1}^{k} I_\nu\right| > r/2$. Then, clearly, $\left|\bigcup_{\nu=1}^{k} I_\nu\right| \leq r$ (since $|I_\nu| \leq r/2$). Therefore, $\left|\partial_A\left(\bigcup_{\nu=1}^{k} I_\nu\right)\right| > (rc/2)$. On the other hand, (4.7) implies that every new boundary element that results from appending via $\vdash_t$ some set of rows must belong to $J(t)$, therefore $\partial_A\left(\bigcup_{\nu=1}^{k} I_\nu\right) \subseteq J(t)$. This contradiction with the assumption $\deg(t) \leq (cr/2)$ proves Lemma 4.3.16.■

The following is analogous to Lemma 4.3.12.

Lemma 4.3.17 *Assume that t is a term, and I is a subset of rows such that $I \supseteq Sup(t)$ and $|I| \leq r/2$. Then*

$$R_{Span(I)}(t) = R_{Span(Sup(t))}(t).$$

Proof. Since $Sup(t) \not\vdash_t I \setminus Sup(t)$, by (4.7) some row $i \in I \setminus Sup(t)$ contains an element from $\partial_A(I) \setminus J(t)$. Thus we can remove i by Lemma 4.3.14. In such a way we consequently get rid of all the axioms in $I \setminus Sup(t)$.■

The rest of the proof is quite analogous to that of Theorem 4.3.8.■

4.4 Applications

In this section we describe some concrete lower bounds that can be proved using the results of Section 4.3. For applications in this chapter we need good expanders with parameters slightly different from those considered in Chapter 3. We start with constructions of these expanders.

4.4.1 More on expanders

[CS88] in their work introduced the notion of a sparse hypergraph which in our language (rows correspond to edges, columns correspond to vertices) looks as follows: an $(m \times n)$ 0-1 matrix A is (x, y)-*sparse* if for every $J \subseteq [n]$ with $|J| \leq xn$ we have $|\{i \in [m] \mid J_i(A) \subseteq J\}| \leq y \cdot |J|$. They also established (implicitly and for the case $c = 1/2$) the following connection between sparsity and expansion (*union bound*) which was later utilized in [BP96, BKPS98, ABSRW00]: any $(m \times n)$ $\left(\frac{r(k+c)}{2n}, \frac{2}{k+c}\right)$-sparse matrix in which every row contains exactly k ones is an (r, k, c)-expander, for arbitrary parameters r, k, c.

[CS88, Lemma 1] gave a sufficient condition for a random $(\Delta n \times n)$ matrix (in which every row has exactly k ones) to be (x, y)-sparse. They considered only the case when the parameters k, y, Δ (the latter is denoted in [CS88] by c) are constants. We need the following simple generalization of their lemma.

Let k be an integer constant, $y = y(n)$ be any real parameter such that $(k-1)y > 1$ and $\Delta = \Delta(n)$ be an arbitrary integer parameter satisfying

$$\Delta = o\left(n^{(k-1-y^{-1})}\right). \tag{4.8}$$

Then a random $(\Delta n \times n)$ *matrix in which every row has exactly* k *ones is* $\left(\Omega\left(\Delta^{-y/((k-1)y-1)}\right), y\right)$*-sparse with probability* $1 - o(1)$.

The proof literally follows the proof of [CS88, Lemma 1]; we only need to change the values of their bounds $f(n), g(n)$ to

$$f(n) \stackrel{\text{def}}{=} e\left(\frac{e}{y}\right)^y \cdot n^{-((k-1)y-1)/2} \cdot \Delta^{y/2},$$
$$g(n) \stackrel{\text{def}}{=} \left(n \cdot \Delta^{-1/(k-1-y^{-1})}\right)^{1/2}.$$

The necessary asymptotics $f(n) \to 0$, $g(n) \to \infty$ then follow from (4.8).

Putting things together by setting $y \stackrel{\text{def}}{=} \frac{2}{k+c}$, we have:

Lemma 4.4.1 *Assume that* $k \geq 3$ *is a fixed integer constant,* $0 < c = c(n) < k - 2$ *is an arbitrary real parameter, and* $\Delta = \Delta(n)$ *is an arbitrary integer parameter satisfying* $\Delta = o\left(n^{(k-c-2)/2}\right)$. *Then a random* $(\Delta n \times n)$ *matrix* A *in which each line* $J_i(A)$ *is chosen from all* $\binom{n}{k}$ k*-subsets of* $[n]$ *independently and at random is* $\left(\Omega\left(\frac{n}{\Delta^{2/(k-c-2)}}\right), k, c\right)$*-expander with probability* $1 - o(1)$.

Now let us turn to another source of good expanders. For an ordinary graph $G = (V, E)$ and $r \geq 1$ let

$$c_E(r, G) \stackrel{\text{def}}{=} \min_{|U| \leq r} \frac{e(U, V - U)}{|U|},$$

where $e(U, W)$ is the number of edges between U and W. This is a minor generalization of the edge-expansion coefficient $c_E(G) = c_E(|V|/2, G)$ previously studied in graph theory (see e.g. [Alo98] and the literature cited therein). Clearly, the incidence matrix A_G of a graph G is an $(r, d(G), c_E(r, G))$-expander for an arbitrary r (cf. Example 5), where $d(G)$ is the maximal degree of a vertex.

Suppose now that the graph G is d-regular. Then, clearly, $c_E(r, G) = d - \max_{|U| \leq r} ad(G|_U)$, where $G|_U$ is the subgraph induced on U, and ad is the average degree. [BCS78] proved the following bound on this quantity in terms of the second

eigenvalue $\lambda_2(G)$ of the graph G:

$$ad(G|_U) \leq \frac{|U|(d-\lambda_2(G))}{|V|} + \lambda_2(G).$$

This implies

$$c_E(r,G) \geq d\left(1-\frac{r}{|V|}\right) - \lambda_2(G).$$

Recall that a *Ramanujan graph* is a d-regular graph G with $\lambda_2(G) \leq 2\sqrt{d-1}$; explicit constructions of such graphs were given in [LPS88, Mar88]. Summing up the above, we have:

Lemma 4.4.2 *The incidence matrix of any d-regular Ramanujan graph G on n vertices is an $(r, d, d(1-r/n) - 2\sqrt{d-1})$-expander for any parameter $r > 0$.*

4.4.2 Tseitin tautologies: Boolean version

A Tseitin tautology is an unsatisfiable CNF capturing the basic combinatorial principle that for every graph, the sum of degrees of all vertices is even. These tautologies were originally used by Tseitin [Tse68] to present the first super-polynomial lower bounds on refutation size for a certain restricted form of Resolution (regular resolution).

In the sequence of works [Gri98, BGIP01, BI99] their authors studied the hardness of Tseitin tautologies for Polynomial Calculus. This research was essentially dependent on the fact that the tautologies can be written in the form of binomial ideals. Then the arguments proposed in [BGIP01] (and simplified in [BI99]) show that any PC-refutation of Tseitin tautologies has degree $\Omega(n)$.

[BGIP01] generalized Tseitin tautologies to the case when each vertex in the graph contains MOD_p function and used them to lower bound the refutation degree of the counting principle $Count_p$. Their definition, stated informally in terms of a flow on a graph, says the following: each directed edge e has the corresponding variable x_e which ranges over $\{0, ..., p-1\}$; the intuitive meaning of x_e is the value that flows along the edge e. The mod_p Tseitin principle says that the sum of flows in all vertices

is equal to 0 mod p.

This definition is a natural generalization of the usual mod_2 Tseitin tautologies. [BGIP01] proved an $\Omega(n)$ lower bound on the refutation degree of these tautologies in the version of Polynomial Calculus in which the relations $x_i^2 - x_i$ (hardwired in our definition of $S_n(\mathbb{F})$) are weakened to $x_i^p - x_i$. They also observed that a Boolean version can be obtained by repeating every edge of the underlying graph p times. We would like to propose another Boolean version which is more straightforward and does not involve any encodings.

In our variant the variable x_e written on the directed edge e can have only Boolean values 0 and 1. There are two different constants F_0 and F_1 from $\{0, \ldots, p-1\}$ which define the amount of flow along e when x_e is equal to 0 and 1 respectively. One can easily see (by applying an affine transformation) that the exact choice of constants is not essential; for definiteness we set $F_0 := 0,\ F_1 := 1$.

Definition 4.4.3 (mod_p **Tseitin Formulas)** Let G be a finite oriented graph and $\sigma : V(G) \to \{0, ..., p-1\}$ be an arbitrary function. Assign a distinct Boolean variable x_e to each directed edge $e \in E(G)$. For $v \in V(G)$ denote by $MOD_p(G, \sigma, v)$ the following Boolean predicate:

$$\sum_{\{w|\, <w,v> \in E\}} x_{<w,v>} - \sum_{\{w|\, <v,w> \in E\}} x_{<v,w>} = \sigma(v) \bmod p.$$

The mod_p *Tseitin formula* of G and σ is defined as

$$T_p(G, \sigma) \stackrel{\text{def}}{=} \bigwedge_{v \in V(G)} \{MOD_p(G, \sigma, v)\}.$$

It is easy to see that $T_2(G, \sigma)$ coincides with ordinary Tseitin tautologies. We prove that for graphs G with sufficiently large edge-expansion $T_p(G, \sigma)$ requires large refutation degree in fields $\mathbb{F}$ with char $\mathbb{F} \neq p$.

Theorem 4.4.4 *The Boolean function in d variables which outputs 1 on $\alpha_1, \ldots, \alpha_d$ if and only if $\sum_{j=1}^{d} \epsilon_j \alpha_j \equiv \sigma \bmod p$, where $\epsilon_j \in \{\pm 1\}$ and $\sigma \in \{0, 1, \ldots, p-1\}$ are*

arbitrary, is $\lfloor \frac{d}{4(p-1)} \rfloor$*-immune over any field* $\mathbb{F}$ *with* char $\mathbb{F} \neq p$.

Proof. We can assume w.l.o.g. $\epsilon_1 = \epsilon_2 = \ldots = \epsilon_{d_1} = 1$, $\epsilon_{d_1+1} = \ldots = \epsilon_d = -1$ and $d_1 \geq d/2$. Given Corollary 4.2.4(1), the main result from [Gre00] (Theorem 3.4) implies that for every $\sigma \in \{0, 1, \ldots, p-1\}$, $MOD_{p,\sigma}(x_1, \ldots, x_{d_1})$ is $\lfloor \frac{d_1}{2(p-1)} \rfloor \geq \lfloor \frac{d}{4(p-1)} \rfloor$-immune over any field with char $\mathbb{F} \notin \{0, p\}$. By Corollary 4.2.4(2), this bound can be also extended to fields $\mathbb{F}$ with char $\mathbb{F} = 0$. Theorem 4.4.4 now follows from Lemma 4.2.5(2) applied to $V \stackrel{\text{def}}{=} \{x_{d_1+1}, \ldots, x_d\}$.■

Theorem 4.3.8, Lemma 4.4.2 and Theorem 4.4.4 imply

Corollary 4.4.5 *For any fixed prime* p *there exists a constant* $d_0 = d_0(p)$ *such that the following holds. If* $d \geq d_0$, G *is a* d*-regular Ramanujan graph on* n *vertices (augmented with an arbitrary orientation of its edges), and* char $\mathbb{F} \neq p$, *then for every function* σ *every PC refutation of* $T_p(G, \sigma)$ *over* $\mathbb{F}$ *has degree* $\Omega(dn)$.

4.4.3 Random k-CNF in characteristic 2

An interesting test for a propositional proof system is how effective it behaves on the random input.

Definition 4.4.6 (Random k-CNF's) Let $\mathcal{F} \sim \mathcal{F}_k^{n,\Delta}$ denote that $\mathcal{F}$ is a random k-CNF formula on n variables and $\Delta \cdot n$ clauses, chosen by picking $\Delta \cdot n$ clauses independently and at random from the set of all $\binom{n}{k} \cdot 2^k$ clauses, with repetitions. Δ is called the *clause density.*

[CS88] showed that the random 3-CNF with n variables and $\Delta \cdot n$ clauses requires exponential refutation in Resolution, for an arbitrary constant Δ. [BI99] proved that the random 3-CNF requires Polynomial Calculus refutation of degree $\Omega(n)$ over any field $\mathbb{F}$ with char $\mathbb{F} \neq 2$, provided $\Delta = \Delta(n)$ is small enough. They used binomial technics proposed in [BGIP01] and the fact that the random CNF has good expansion properties proved in [CS88]. [BI99] conjectured that the same lower bound on the degree holds for 3-CNF's over the fields $\mathbb{F}$ with char $\mathbb{F} = 2$ as well. We give a positive answer to their conjecture.

Using Theorem 4.3.13 and Lemma 4.4.1 with $c = 1/\ln(\Delta+2)$ we immediately get the following

Corollary 4.4.7 *Let $\mathcal{F} \sim \mathcal{F}_k^{n,\Delta}$, where $k \geq 3$ is a fixed integer and $\Delta = \Delta(n)$ is an arbitrary parameter satisfying $\Delta \leq o\left(n^{(k-2)/2}\right)$. Then every Polynomial Calculus refutation of $\mathcal{F}$ over an arbitrary field $\mathbb{F}$ has degree $\Omega\left(\frac{n}{\Delta^{2/(k-2)} \cdot \log \Delta}\right)$ with probability $1 - o(1)$.*

4.4.4 Collapsable functions and flow tautologies

In this section we define a wide class of *collapsable* functions and prove that they have strong immunity. Then we introduce the analog of Tseitin tautologies in characteristic zero, in which each vertex contains a linear inequality over $\mathbb{R}$. The principle says that the flow can not be positive in all vertices of the graph. We call this family *flow tautologies.*

Definition 4.4.8 A Boolean function f is *ℓ-collapsable* iff for any subset of variables $S \subseteq \{x_1, ..., x_n\}$ with $|S| < \ell$ there exists a restriction ρ which leaves variables from S unassigned and such that $f|_\rho = 1$. A polynomial is *ℓ-collapsable* iff the characteristic function of the set of its roots in $\{0,1\}^n$ is ℓ-collapsable.

In other words, a polynomial $f \in S_n(\mathbb{F})$ is ℓ-collapsable iff for any choice of $n-\ell+1$ variables we can satisfy it (make equal to 0) by some restriction of these variables to 0 and 1. It turns out that collapsable polynomials have strong immunity.

Theorem 4.4.9 *Every ℓ-collapsable polynomial f is ℓ-immune.*

Proof. Assume that $f \models g$ and $\deg(g) < \ell$. Let us choose any term t in g of maximal possible degree. Let $S \stackrel{\text{def}}{=} Vars(t)$. By the definition of ℓ-collapsable polynomial, there exists a restriction ρ which sends f (and hence g) to 0 and does not touch t. But $g|_\rho$ still contains t and hence is non-zero. This contradiction proves Theorem 4.4.9.■

One good example of collapsable functions is made by threshold functions $\sum_{i=1}^n x_i > k$. In particular, the majority function MAJ_n defined by the predicate $\sum_{i=1}^n x_i > n/2$

is $n/2$-collapsable. To see that, assume that a subset of variables S with $|S| < n/2$ is given. Assign the rest of variables to 1, it will satisfy the function. Thus, we have shown that MAJ_n is $n/2$-immune over any field (another, more direct proof of this result can be also easily extracted from the proof of [Tsa96, Theorem 4.1])

Now we are ready to define the analog of Tseitin principle in the characteristics 0. Recall that in the case of mod_p Tseitin tautologies we have an oriented graph G with Boolean variables corresponding to its edges, and the axioms in each vertex v saying that its flow is equal to $\sigma(v)$ mod p. If instead of fixing the flow mod p we demand that it is positive in each vertex, we get *flow tautologies.*

Definition 4.4.10 (Flow tautologies) Let G be a finite oriented graph. Assign a distinct Boolean variable x_e to each directed edge $e \in E(G)$. For $v \in V(G)$ denote by $PosFlow(G, v)$ the following Boolean predicate:

$$\sum_{\{w|\, <w,v> \in E\}} (1 - 2x_{<w,v>}) > \sum_{\{w|\, <v,w> \in E\}} (1 - 2x_{<v,w>}).$$

The *Flow tautology* of G is defined as (the negation of)

$$Fl(G) \stackrel{\text{def}}{=} \bigwedge_{v \in V(G)} PosFlow(G, v).$$

It is easy to see that, up to negating some variables, $PosFlow(G, v)$ coincides with the majority function in $d(v)$ variables and hence is $d(v)/2$-collapsable. Thus by Theorem 4.4.9 and Lemma 4.4.2 we have

Corollary 4.4.11 *If G is a d-regular Ramanujan graph on n vertices with $d \geq 255$ (augmented with an arbitrary orientation of its edges) then every PC refutation of $Fl(G)$ over an arbitrary field has degree $\Omega(dn)$.*

4.4.5 Extended Pigeonhole Principle

In this section we prove degree lower bounds for PC refutations of Extended Pigeonhole principle defined in [BW99].

The Pigeonhole principle with m pigeons and n holes states that there is no 1-1 map from $[m]$ to $[n]$, as long as $m > n$. This can be stated by a formula on mn variables x_{ij}, where $x_{ij} = 1$ means that i is mapped to j.

Definition 4.4.12 PHP_n^m is the conjunction of the following clauses:

- $P_i \stackrel{\text{def}}{=} \bigvee_{1 \leq j \leq n} x_{ij}$ for $1 \leq i \leq m$
- $H_{i,i'}^j \stackrel{\text{def}}{=} \bar{x}_{ij} \vee \bar{x}_{i'j}$ for $1 \leq i < i' \leq m, 1 \leq j \leq n$.

The problem with this classical definition is that the clauses P_i can not be expressed as polynomials of low degree. That is why in case of Polynomial Calculus one usually considers a stronger version of PHP_n^m in which no pigeon can simultaneously "fly" into two different holes (which in particular implies that the big disjunction can be replaced with a linear function, see e.g. [Raz98]). There is, however, still another way to state PHP_n^m in order to express it by a family of low degree polynomials which is perhaps more natural in our framework developed in Chapter 3.

Definition 4.4.13 ([BW99]) For $f(\vec{x})$ a Boolean function, a *nondeterministic extension* of f is a function $g(\vec{x}, \vec{y})$ such that $f(\vec{x}) = 1$ iff $\exists y\ g(\vec{x}, \vec{y}) = 1$. $\vec{x}$-variables are called *original* variables and $\vec{y}$-variables are called *extension* variables.

$EPHP_n^m$ is obtained from PHP_n^m by replacing every row axiom P_i with some nondeterministic extension CNF formula EP_i using distinct extension variables $\vec{y}_i$ for distinct rows.

Now, we express Pigeonhole principle as a family of polynomials by encoding every clause C of $EPHP_n^m$ with the corresponding polynomial p_C. Since EP_i can be chosen as 3-CNF, this eliminates the problem with the degree of axioms.

Our main result in this section is the new proof of the following theorem. The advantage of this new proof is that it does not use any specific calculations.

Theorem 4.4.14 *For $m = O(n)$ any Polynomial Calculus refutation of $EPHP_n^m$ must have degree $\Omega(n)$.*

Proof.

Let us consider some PC refutation $\mathcal{P}$ of $EPHP_n^m$. Choose any $m \times n$ (r, s, c)-expander A with constant s, c and $r = \Omega(n)$ (for example, we can take the random expander from Lemma 4.4.1). Let us restrict our Pigeonhole principle so that the pigeon i may "fly" only into holes $j \in J_i(A)$. Namely, let us apply to $\mathcal{P}$ the restriction ρ that sets $x_{ij} = 0$ for all $j \notin J_i(A)$.

Our next goal is to eliminate all extension variables from the proof. For that, consider the ith extension axiom $EP_i|_\rho$ in the refutation $\mathcal{P}|_\rho$. By definition, $P_i(\vec{x}_i) = 1$ iff $\exists \vec{y}_i \; g(\vec{x}_i, \vec{y}_i) = 1$. Clearly the dependence of $\vec{y}_i$ on $\vec{x}_i$ can be made deterministic in the sense that there exist functions $\vec{h}_i(\vec{x}_i)$ s.t. $P_i|_\rho(\vec{x}_i) = 1$ iff $(EP_i)|_\rho(\vec{x}_i, \vec{h}_i(x_i)) = 1$. Since we restricted all but s variables of $\vec{x}_i$ to zero, every $\vec{h}_i$ essentially depends on at most s variables. Let us replace in the proof $\mathcal{P}|_\rho$ each extension variable $y_{ik} \in \vec{y}_i$ with the polynomial $1 - p_{h_{ik}}$. Clearly the degree of $\mathcal{P}|_\rho$ will increase by at most a factor of s. Thus in order to prove the theorem it is sufficient to estimate the degree of this new refutation.

It is easy to see that initial polynomials corresponding to the axioms EP_i will be mapped into the polynomials that semantically correspond to the clauses $\bigvee_{j \in J_i(A)} x_{ij}$. Thus w.l.o.g. we can assume that they are mapped into polynomials $f_i \stackrel{\text{def}}{=} \prod_{j \in J_i(A)}(1 - x_{ij})$ so we have a PC refutation of the system

$$\{f_1, ..., f_m\} \cup \{x_{ij} \cdot x_{i'j} \mid i \neq i', \; j \in J_i(A) \cap J_{i'}(A)\}$$

that has degree at most $s \cdot \deg(\mathcal{P})$.

In order to finish the proof, we need a multi-valued version of Theorem 4.3.13 (cf. [ABRW02]). Namely, suppose that instead of Boolean variables x_j we have multi-valued variables $\hat{x}_j \in \{1, ..., d\}$ that are represented by tuples of Boolean variables $\vec{x}_j = (x_{1j}, ..., x_{dj})$ with the intended semantical meaning $x_{\ell j} \stackrel{\text{def}}{=} (\hat{x}_j = \ell)$. Like in the boolean case, for a tuple $\vec{\epsilon} \in \{1, ..., d\}^k$ let $\chi_{\vec{\epsilon}}(\vec{x}_1, \vec{x}_2, ..., \vec{x}_k) \stackrel{\text{def}}{=} \prod_{j=1}^k (1 - x_{\epsilon_j, j})$.

Suppose that at the top of equations $f_1(\vec{x}_1, \ldots, \vec{x}_n) = \ldots = f_m(\vec{x}_1, \ldots, \vec{x}_n) = 0$ with the same meaning as before our system additionally contains the equations

$x_{\ell j}x_{\ell' j} = 0$ for all $j \in [n], 1 \leq \ell < \ell' \leq d$. We adjust Definition 4.3.9 for the multi-valued case as follows:

$$J(t) \stackrel{\text{def}}{=} \{j | t \text{ has a non-empty intersection with } \vec{x}_j\}$$

and

$$Span(I) \stackrel{\text{def}}{=} Span(\{f_i | i \in I\} \cup \{x_{\ell j}x_{\ell' j} | j \in \bigcup_{i \in I} J_i(A),\ 1 \leq \ell < \ell' \leq d\}).$$

Then the analogue of Theorem 4.3.13 in this multi-valued framework looks like this:

Assume that A is an (r, s, c)-expander, and let $\bar{\epsilon}^{(i)} \in \{1, ..., d\}^{X_i(A)}$. Then any PC refutation of the system $\{\chi_{\bar{\epsilon}^{(i)}}(X_i(A)) | i \in [m]\} \cup \{x_{\ell j}x_{\ell' j} | j \in [n],\ 1 \leq \ell < \ell' \leq d\}$ has degree greater than $(rc/2)$.

In this form Theorem 4.3.13 can be directly applied to our case (the multi-valued variable $\hat{x}_j$ runs over $\{i \in [m] \mid j \in J_i(A)\}$). Theorem 4.4.14 follows.■

4.4.6 Relation between robustness and immunity

In this section we discuss the relation between the hardness condition considered in Chapter 3 and that of the current chapter. We show that every $(s - k)$-robust function (see the definition below) is also $\omega(1)$-immune in the fields of characteristic 0 ($k = const$, $s \to \infty$). This estimate is extremely week but still even sufficiently large constant immunity gives non-trivial lower bounds on the degree of PC refutations.

Recall, that a Boolean function f is *ℓ-robust* if every restriction ρ such that $f|_\rho = const$, satisfies $|\rho| \geq \ell$. This notion is clearly invariant under negations. In order to compare it with non-invariant immunity let us call the function f *ℓ-semi-robust* if $f|_\rho \equiv 0$ implies $|\rho| \geq \ell$. Thus, a Boolean function is ℓ-robust iff it is ℓ-semi-robust and so is its negation. Every ℓ-immune Boolean function is ℓ-semi-robust by Lemma 4.2.5(1), therefore the notion of immunity is stronger. As the example of the MOD_p function shows, in positive characteristic immunity can be a much stronger requirement than (semi-)robustness.

In the opposite direction the following estimate holds:

Theorem 4.4.15 *Assume that* $\mathrm{char}\ \mathbb{F} = 0$, k *is a constant. Then every* $(s-k)$*-semi-robust Boolean function* f *in* s *variables has immunity* $\omega(1)$ *when* $s \to \infty$.

Proof. By Corollary 4.2.4(2), we may assume w.l.o.g. that $\mathbb{F} = \mathbb{Q}$. We will need the following classical definition of Ramsey numbers.

Definition 4.4.16 The *Ramsey number* $R_k(l_1, \ldots, l_r)$ is the smallest n such that if all k-subsets of $[n]$ are coloured in r colours, then there exists a colour ν and an ℓ_ν-subset of $[n]$ all of whose k-subsets have colour ν.

Let $N_k(d)$ be the smallest s such that for every non-zero polynomial $g \in \mathbb{Q}[x_1, \ldots, x_s]$ with $\deg(g) \leq d$ there is a restriction ρ such that $|\rho| \leq s-k-1$ and $g|_\rho$ does not have $(0-1)$ roots. Thus, this is the inverse function to what we are studying: namely, if g is a semantic corollary of the polynomial p_f (where f is a $(s-k)$-semi-robust Boolean function), then $N_k(\deg(g)) > s$. N_k is monotone and we need to prove that $\forall d\ N_k(d) < \infty$. Clearly, $N_k(0) = k+1$. Our result follows from the following recursive bound:

$$N_k(d) \leq R_d(2^{k+1}d, 2^{k+1}d, N_k(d-1)).$$

In order to see this, let $s \geq R_d(2^{k+1}d, 2^{k+1}d, N_k(d-1))$ and suppose that $\deg(g) \leq d$. Colour all d-subsets of $[s]$ in three colours, $+, -, 0$, according to the sign of the coefficient in front of the corresponding monomial in g. Let us denote $m = 2^{k+1}d$. According to the definition of Ramsey numbers, we have two cases.

Case 1. *For some* m *variables, say,* $x_1, \ldots, x_m$, g *contains all monomials* x_I *with* $I \in [m]^d$, *and it contains them with the same sign.* Consider $k+1$ variables $x_{m+1}, x_{m+2}, \ldots, x_{m+k+1}$. If there exists a restriction ρ to all variables except $x_{m+1}, x_{m+2}, \ldots, x_{m+k+1}$ s.t. $g|_\rho$ does not have $(0-1)$ roots, then our recursive bound follows. Otherwise for any assignment of $Vars(g) \setminus \{x_{m+1}, x_{m+2}, \ldots, x_{m+k+1}\}$ there exist values for $x_{m+1}, x_{m+2}, \ldots, x_{m+k+1}$ which map g to 0. In other words, at least one of the functions

$$g|_{(x_{m+1}=\epsilon_1, \ldots, x_{m+k+1}=\epsilon_{k+1})}$$

is equal to 0 on the chosen assignment or, equivalently,

$$\prod_{\vec{\epsilon}\in\{0,1\}^{k+1}} g|_{(x_{m+1}=\epsilon_1,\ldots,x_{m+k+1}=\epsilon_{k+1})} = 0 \text{ in } S_s(\mathbb{Q}).$$

This, however, is impossible since all monomials x_I with $I \in [m]^d$ still appear in all $g|_{(x_{m+1}=\epsilon_1,\ldots,x_{m+k+1}=\epsilon_{k+1})}$ with the same sign, and therefore the monomial $\prod_{i=1}^{m} x_i$ has a non-zero coefficient in this product.

Case 2. *For some $N_k(d-1)$ varaiables, say, $x_1,\ldots,x_{N_k(d-1)}$, g contains no monomial x_I with $I \in [N_k(d-1)]^d$.* In this case we simply apply any restriction to the remaining variables that does not kill g completely (which reduces the degree), and then apply the inductive assumption.

Theorem 4.4.15 is proved.■

According to the convention made in Section 4.2 (cf. Definition 4.2.1), we say that a polynomial f is *ℓ-semi-robust* if the characteristic function of the set of its roots is so.

Theorem 4.4.17 *For any fixed integers k, α there exists an integer s_0 s.t. for any $s \geq s_0$, $(r, s, s-\alpha)$-expander A and $f_1,\ldots,f_m$ $(s-k)$-semi-robust polynomials over an arbitrary field of characteristic 0 with $Vars(f_i) \subseteq X_i(A)$, any PC refutation of the system $f_1 = \ldots = f_m = 0$ has degree $\Omega(r)$.*

This theorem follows from Theorems 4.3.8 and 4.4.15. Using the construction of random expanders from Lemma 4.4.1 we can build various families of hard tautologies based on $(s-k)$-semi-robust functions.

4.5 Open problems

The most interesting of remaining questions is what can be said about the system (4.2) in the case of small expansion factor $c > 0$? Can one show its hardness provided that f_i are sufficiently immune, otherwords is it possible to combine our Theorems 4.3.8, 4.3.13 into one general statement?

Does there exist a relation between robustness and immunity stronger than that of Theorem 4.4.15?

Part II

Lower Bounds on Automatizability

Chapter 5

Hardness to Aprroximate Minimum Propositional Proof Length

5.1 Introduction

This paper proves lower bounds on the hardness of finding short propositional proofs of a given tautology and on the hardness of finding short resolution refutations. When considering Frege proof systems, which are textbook-style proof systems for propositional logic, the problem can be stated precisely as the following optimization problem:

Minimum Length Frege Proof:
Instance: A propositional formula φ which is a tautology.
Solution: A Frege proof P of φ.
Objective function: The number of symbols in the proof P.

For a fixed Frege system $\mathbb{F}$, let $\min_{\mathbb{F}}(\varphi)$ denote the minimum number of symbols in an $\mathbb{F}$-proof of φ. An algorithm M is said to approximate the Minimum Length Frege Proof problem within factor α, if for all tautologies φ, $M(\varphi)$ produces a Frege proof of φ of length $\leq \alpha \cdot \min_{\mathbb{F}}(\varphi)$. (Here, α may be a constant or may be a function

of the length of φ.)

We are interested only in *polynomial time* algorithms for solving this problem. However, there is a potential pitfall here since the shortest proof of a propositional formula could be substantially longer than the formula itself,[1] and in this situation, an algorithm with runtime bounded by a polynomial of the length of the input could not possibly produce a proof of the formula. In addition, it seems reasonable that a "feasible" algorithm which is searching for a proof of a given length ℓ should be allowed runtime polynomial in ℓ, even if the formula to be proved is substantially shorter than ℓ. Therefore we shall only discuss algorithms that are polynomial time in the length of the shortest proof (or refutation) of the input.

Note that an alternative approach would be to consider a similar problem, **Minimum Length Equivalent Frege Proof**, an instance of which is a Frege proof of some tautology φ, and the corresponding solutions are (preferably shorter) proofs of φ. While our results are all stated in terms of finding a short proof to a given tautology, they hold also for that latter version where the instance is a proof rather than a formula.

A yet different approach could be studying algorithms which output the *size* (i.e., number of symbols) of a short proof of the input formula, rather than the proof itself. In this case it is possible for an algorithm to have run time bounded by a polynomial of the length of the input formula, even if the size of the shortest proof is exponential in the size of the formula. In the final section of this paper, we show that strong non-approximability results can be obtained for algorithms with run time bounded by a polynomial of the length of the formula for a variety of proof systems.

A related minimization problem concerns finding the shortest Frege proof when proof length is measured in terms of the number of steps, or lines, in the proof:

Minimum Step-Length Frege Proof:

Instance: A propositional formula φ which is a tautology.

Solution: A Frege proof P of φ.

[1] Is is known that **NP** $\neq$ **coNP** implies that some tautologies require superpolynomially long Frege proofs.

Objective function: The number of steps in the proof P.

Resolution is a propositional proof system which is popular as a foundation for automated theorem provers. Since one is interested in finding resolution refutations quickly it is interesting to consider the following problem:

Minimum Length Resolution Refutation

Instance: An unsatisfiable set Γ of clauses.

Solution: A resolution refutation R of Γ.

Objective function: The number of inferences (steps) in R.

The main results of this paper state that a variety of minimum propositional proof length problems, including the Minimum Length Frege Proof, the Minimum Step-Length Frege Proof and the Minimum Length Resolution Refutation problems, cannot be approximated to within a factor $2^{\log^{1-o(1)} n}$ by any polynomial time algorithm unless $\mathbf{P} = \mathbf{NP}$ (we use here the recent result of [DS99], see section 5.2). Our results apply to all Frege systems, to all extended Frege systems, to resolution, to Horn clause resolution, to the sequent calculus, and to the cut-free sequent calculus; in addition, they apply whether proofs are measured in terms of symbols or in terms of steps (inferences), and they usually apply to either dag-like or tree-like versions of all these systems.

We let $\mathbb{F} \stackrel{k}{\vdash} \varphi$ mean that φ has an $\mathbb{F}$-proof of $\leq k$ symbols. One of the first prior results about the hardness of finding optimal length of Frege proofs was the second author's result [Bus95a] that, for a particular choice of Frege system $\mathbb{F}_1$ with the language $\wedge$, $\vee$, $\neg$ and $\rightarrow$, there is no polynomial time algorithm which, on input a tautology φ and a $k > 0$, can decide whether $\mathbb{F}_1 \stackrel{k}{\vdash} \varphi$, unless P equals $\mathbf{NP}$. This result however applies only to a particular Frege system, and not to general Frege systems. It also did not imply the hardness of approximating Minimum Length Frege Proofs.

A second related result, which follows from the results of Krajíček and Pudlák [KP95], is that if the RSA cryptographic protocol is secure, then there is no polynomial time algorithm for approximating the Minimum Step-Length Frege Proof problem to within a polynomial.

Another closely related prior result is the connection between the (non)automatizability of Frege systems and the (non)feasibility of factoring integers that was recently discovered by Bonet-Pitassi-Raz [BPR97]. A proof system T is said be *automatizable* provided there is an algorithm M and a polynomial p such that whenever $T \vdash^n \varphi$ holds, $M(\varphi)$ produces some T-proof of φ in time $p(n)$ (see [CEI96]). Obviously the automatizability of Frege systems is closely related to the solution of the Minimum Length Frege Proof problem: if a proof system S is automatizable, then the minimum length proof problem for S can be approximated to within a polynomial factor. The two concepts are not equivalent since automatization is a property of the search problem, whereas the minimum length proof problem concerns the associated decision problem. Our theorems give a super-linear lower bound on the automatizability of the Minimum Proof Length problem based on the assumption that $\mathbf{P} \neq \mathbf{NP}$. It has recently been shown by Bonet-Pitassi-Raz [BPR97] that Frege systems are not automatizable unless Integer Factorization is in P, and more recently, that bounded-depth Frege systems are also not automatizable under a similar hardness assumption [BDG+99]. These results give stronger non-approximability conclusions, but require assuming a much stronger complexity assumption.

For resolution, the first prior hardness result was Iwama-Miyano's proof in [IM95] that it is **NP**-hard to determine whether a set of clauses has a read-once refutation (which is necessarily of linear length). Subsequently, Iwama [Iwa97] proved that it is in **NP**-hard to find shortest resolution refutations; unlike us, he did not obtain an approximation ratio bounded away from 1.

In section 5.2, we introduce the MMSA and Circuit MMSA problems and discuss the relevant prior results about the hardness of approximating **NP**-optimization problems. Section 5.3 discusses the main results about the hardness of approximating minimum length of refutations on the example of Resolution. Section 5.4 contains the main results about the hardness of approximating minimum length Frege proofs. The proofs in sections 5.3 and 5.4 depend critically on the hardness of approximation of (Circuit) MMSA problem.

5.2 Monotone Minimum Satisfying Assignment

In this section we introduce the Monotone Minimum Satisfying Assignment (MMSA) problem and show a few structural results about its complexity from the point of view of the hardness of approximation. In fact for our further sections it is enough to use the recent result of [DS99] (Theorem 5.2.2 below) which shows that MMSA is hard to approximate within $2^{\log^{1-o(1)} n}$ factor unless $\mathbf{P} = \mathbf{NP}$. This result appeared after submission of our paper thus we prefer to keep the content of the section to give more global picture.

The reader can find a general introduction to and survey of the hardness of approximation and of probabilistically checkable proofs in [Bel96] and [AL96]. Recall that an A-reduction, as defined by [KST97], is a polynomial-time Karp-reduction which preserves the non-approximating ratio to within a constant factor.

Consider the following **NP**-optimization problems:

Monotone Minimum Satisfying Assignment:

Instance: A monotone formula $\varphi(x_1, \ldots, x_n)$ over the basis $\{\vee, \wedge\}$.

Solution: An assignment $\langle v_1, \ldots, v_n \rangle$ such that $\varphi(v_1, \ldots, v_n) = \top$.

Objective function: The number of v_i's which equal $\top$.

We henceforth let $\gamma(\varphi)$ denote the value of the optimal solution for the MMSA problem for φ i.e., the minimum number of variables v_i which must be set *True* to force φ to have value *True*.

We will also consider the *Circuit MMSA* problem which is to find the minimum number of variables which must be set *True* to force a given monotone circuit over the basis $\{\wedge, \vee\}$ evaluate *True*. It does not matter whether we consider circuits with bounded fanin or unbounded fanin since they can simulate each other. It is apparent that Circuit MMSA is at least as hard as MMSA.

Recall the Minimum Hitting Set problem, which is:

Minimum Hitting Set:

Instance: A finite collection $\mathcal{S}$ of nonempty subsets of a finite set U.

Solution: A subset V of U that intersects every member of $\mathcal{S}$.

Objective function: The cardinality of V.

It is easy to see that MMSA is at least as hard as Minimum Hitting Set: namely Minimum Hitting Set can be reduced (via an A-reduction) to the special case of MMSA where the propositional formula is in conjunctive normal form. Namely, given $\mathcal{S}$ and U, identify members of U with propositional variables and form a CNF formula which has, for each set in $\mathcal{S}$, a conjunct containing exactly the members of that set.

Lund and Yannakakis [LY94] noted that Minimum Hitting Set is equivalent to Minimum Set Cover (under A-reductions). Furthermore, it follows from [RS97] that the problem of approximating Minimum Set Cover to within $\Omega(\ln n)$ factor is not in polynomial time unless $\mathbf{P} = \mathbf{NP}$.

We can get stronger results than the above reduction of Minimum Set Cover to MMSA if we use a construction due to S. Arora [private communication] to reduce MMSA to the Minimum Label Cover problem.

Minimum Label Cover: (see [AL96])

Instance: The input consists of: (i) a regular bipartite graph $G = (U, V, E)$, (ii) an integer N in unary, and (iii) for each edge $e \in E$, a partial function $\Pi_e : \{1, \ldots, N\} \to \{1, \ldots, N\}$ such that 1 is in the range of Π_e.

The integers in $\{1, \ldots, N\}$ are called *labels*. A *labeling* associates a nonempty set of labels with every vertex in U and V. A labeling *covers* an edge $e = (u, v)$ (where $u \in U$, $v \in V$) iff for every label ℓ assigned to v, there is some label t assigned to u such that $\Pi_e(t) = \ell$.

Solution: A labeling which covers all edges.

Objective function: The number of all labels assigned to vertices in U and V.

A Π_4-*formula* is a propositional formula which is written as an AND of OR's of AND's of OR's.

Theorem 5.2.1 (S. Arora) *There is an A-reduction from Minimum Label Cover to MMSA such that the instances of Label Cover are mapped to* Π_4 *formulas.*

Proof. Suppose we have an instance of Label Cover $(G, N, \{\Pi_e\})$. Let $m = |U|$ and $k = |V|$. For $1 \leq i \leq m$ and $1 \leq \ell \leq N$, let u_i^ℓ be a propositional variable with the intended meaning that $u_i^\ell = \top$ iff ℓ is one of the labels assigned to vertex $i \in U$. Likewise, for $1 \leq j \leq m$ and $1 \leq \ell \leq N$ let v_j^ℓ denote the condition that ℓ is one of the labels assigned to vertex $j \in V$. We shall construct a formula φ involving the variables u_i^ℓ and v_j^ℓ so that the minimal satisfying assignments for φ are precisely those truth assignments which correspond to minimal weight labelings which cover all edges. We define φ to be

$$\bigwedge_{j=1}^{k} \bigvee_{\ell=1}^{N} \left(v_j^\ell \wedge \bigwedge_{i|(i,j)\in E} \bigvee_{t|\Pi_e(t)=\ell} u_i^t \right)$$

The formula φ clearly is a monotone Π_4-formula and has size polynomial in N, m, k. It is easy to verify that any minimum satisfying assignment for φ corresponds to a labeling which covers all edges and which has a minimum number of labels: to see this, one should note that any minimum size labeling, as well as an minimum satisfying assignment, will have exactly one label assigned to each vertex V.■

It was proved in [ABSS93] that Minimum Label Cover is not approximable within a $2^{\log^{(1-\varepsilon)} n}$ factor unless **NP** $\subseteq$ **QP**. An immediate corollary of Theorem 5.2.1 is that MMSA enjoys the same hardness of approximation, even when restricted to Π_4-formulas.

Recently [DS99] improved both the factor and the hypothesis of Label Cover. They considered the case of MMSA for Π_3 formulas and showed its hardness directly from some strong version of PCP-Theorem. They also show how to inverse the reduction of Theorem 5.2.1 and reduce MMSA for Π_3 to Label Cover.

We will use their result in further sections:

Theorem 5.2.2 (I. Dinur, S. Safra, [DS99]) *If* $\mathbf{P} \neq \mathbf{NP}$, *then there is no polynomial time algorithm which can approximate MMSA (and hence Circuit MMSA) within a factor* $2^{\log^{1-o(1)} n}$.

If one only demands the circuit compute a monotone function (instead of being monotone) then stronger bounds for non-approximability are known [Uma99]. However our results essentially require that C be monotone. In the next sections we reduce the Circuit MMSA problem first to the Minimum Length Resolution Refutation problem and then to other problems on minimum proof length. This will establish the same hardness of approximation results for these proof length optimization problems.

5.3 The Hardness of Refutations

In this section we prove the simplest hardness result which can be applied to any "reasonable" refutation system. Very informaly the only requirements from the system are its soundness and completeness and each formula in the refutation should be the corollary of a finite number of formulas. All such natural systems as Resolution, Polynomial Calculus, (Bounded Depth) Frege refutation posses these required properties.

We give the proof only for Resolution but it is quite analogous for the other refutation systems. Also, the Hardness Theorem for Resolution shows the intuition of the reduction from Circuit MMSA which will be used in the next sections when we consider the inferences of tautologies.

We shall be concerned exclusively with the propositional resolution systems. Recall that the resolution rule allows an inference of the form

$$\frac{C \cup \{p\} \qquad D \cup \{\overline{p}\}}{C \cup D}$$

It is well-known that resolution is sound and complete as a refutation system; namely, a set Γ is unsatisfiable if and only if the empty clause can be derived using only

resolution inferences from the clauses in Γ.

A *Horn* clause is a clause which contains at most one positive literal.

A resolution refutation consists of a sequence of clauses, ending with the empty clause. We can measure the length or size of a refutation in terms of either its step-length or its symbol-length. The *step-length* of the refutation is just equal to the number of clauses in the refutation. The *symbol-length* is defined to equal the sum of the cardinalities of the clauses appearing in the refutation. (Unlike the situation for Frege proofs where proof length is traditionally defined in terms of symbol-length, there seems to be no fixed convention on how to measure the length of resolution refutations: thus, we shall always explicitly include one of the modifiers 'step-' or 'symbol-'.)

A refutation can be either tree-like or dag-like: unless it is explicitly stated otherwise, refutations are considered to be dag-like. We shall obtain the best possible results in that our upper bounds on length will apply to tree-like refutations and our lower bounds on length will apply to dag-like refutations. In the introduction, we introduced the Minimum Length Resolution Refutation problem: henceforth we'll be more precise and talk about the Minimum Step-Length Resolution Refutation and the the Minimum Symbol-Length Resolution Refutation problems.

Theorem 5.3.1 *There is an A-reduction from the Circuit MMSA problem to the Minimum Length Resolution Refutation problems. This reduction works for both tree-like and dag-like refutations and for both step-length and symbol-length. Furthermore, the reduction produces only sets of Horn clauses.*

Together with Theorem 5.2.2 this yields

Corollary 5.3.2 *If* $\mathbf{P} \neq \mathbf{NP}$*, then there is no polynomial time algorithm which can approximate Minimum Step-Length (Symbol-Length) Resolution Refutation to within* $2^{\log^{1-o(1)} n}$ *factor.*

These hardness results apply to both dag-like and tree-like resolution. In addition, the hardness results apply to Horn resolution, where all the input clauses are Horn clauses.

To prove Theorem 5.3.1, we shall construct the reduction from Circuit MMSA to Minimum Length Resolution Refutation problems. Let C be an instance of Circuit MMSA; we define a set of clauses Γ_C which will be an A-reduction to the Minimum Length Resolution Refutation problems.

Enumerate the subcircuits of C as $C_1, \ldots, C_\ell$, where the input variables are first in the enumeration and where each C_i is listed only after all of its own subcircuits are enumerated, and thus C_ℓ is C. Obviously the number ℓ of subcircuits is less than the number of symbols n in C. We introduce new propositional variables $y_1, \ldots, y_\ell$, and define the set Γ_C to contain the following clauses:

a. The clause $\{\overline{y_\ell}\}$ is in Γ_C.

b. For each $i \leq \ell$, if C_i is $(C_j \wedge C_k)$, then the clause $\{\overline{y_j}, \overline{y_k}, y_i\}$ is in Γ_C.

c. For each $i \leq \ell$, if C_i is $(C_j \vee C_k)$, then the clauses $\{\overline{y_j}, y_i\}$ and $\{\overline{y_k}, y_i\}$ are in Γ_C.

The above clauses describe the evaluation of C; however, note that they say nothing about the truth of the input variables $x_1, \ldots, x_p$ of C. For each variable x_i of C, we introduce new variables $x_{i,j}$ for $j = 1, 2, \ldots, m$, and further include in Γ_C the following clauses:

d. For each $i \leq p$, the clauses $\{x_{i,1}\}$ and $\{\overline{x_{i,m}}, y_i\}$ and

$$\{\overline{x_{i,j}}, x_{i,j+1}\} \quad \text{for } j = 1, \ldots, m-1$$

are included in Γ_C. These clauses are said to be *associated with* x_i.

That completes the definition of Γ_C. Informally, Γ_C asserts that there exists a truth-evaluation to all subcircuits of C such that: the truth evaluation given to the output circuit is 0, the truth evaluation given to all input variables is 1, and the truth evaluation is consistent with all intermediate gate values. Clearly, this is an unsatisfiable formula since C is monotone.

Our next plan is to show that the price to infer the literal y_ℓ and hence get contradiction is nearly equal to $\gamma(C)$ multiplied by the price to infer any of "input" literals y_i, $i \leq p$.

Lemma 5.3.3 *Let C be an instance of Circuit MMSA and let γ equal the cardinality of the minimum satisfying assignment for C.*

(1) *Γ_C has a dag-like refutation with symbol-length (and hence step-length) equal to $O(\gamma m + n)$.*

(2) *Γ_C has a tree-like refutation of symbol-length $O(\gamma m + n^2)$ and step-length $O(\gamma m + n)$.*

Proof. (a) Let $I \subseteq \{x_1, \ldots, x_p\}$ specify a satisfying assignment for C of cardinality γ. The dag-like proof proceeds by first deriving the clause $\{y_i\}$ for each $x_i \in I$. Each $\{y_i\}$ is derived in m steps using the clauses associated with x_i; this part of the refutation takes γm steps. The refutation then derives clauses $\{y_i\}$ starting with smaller values of i and ending with $\{y_\ell\}$ (e.g., the subcircuits of C are processed in a bottom-up order). This takes $O(n)$ steps. One further resolution with the input clause $\{\overline{y_\ell}\}$ completes the refutation. Each clause in the refutation contains a constant number of (in fact, at most three) literals. Hence the symbol-length of the refutation is also $O(\gamma m + n)$.

(b) The above proof is clearly not tree-like. To form a tree-like refutation, we use a top-down procedure to generate the refutation. The first phase of the refutation starts with the clause $\{\overline{y_\ell}\}$ and derives successively clauses of the form $\{\overline{y_{k_1}}, \overline{y_{k_2}}, \ldots, \overline{y_{k_r}}\}$ with $k_1 > k_2 > \cdots > k_r$. Such a clause is resolved with one of the (at most two) clauses that contain y_{k_1} positively. This continues until we have a clause which contains only literals $\overline{y_i}$ corresponding to input x_i of C. It is possible to do this so that the remaining clause is just $\{\overline{y_i} : x_i \in I\}$. For the second phase of the refutation, derive the clauses $\{y_i\}$, for $x_i \in I$, with γm steps, and for the third phases, use γ resolutions to derive the empty clause.

There are obviously $O(n)$ steps in the first and third phases of the derivation, so the whole refutation has $O(\gamma m + n)$ steps. Furthermore, each clause in the first and third phase contains at most n literals. The second phase contains γm clauses each with a constant number of literals. Therefore, the symbol-length of the tree-like refutation is $O(\gamma m + n^2)$.■

Lemma 5.3.4 *Let C and γ be as above. Then any resolution refutation (dag-like or tree-like) must have step-length, and hence symbol-length, of at least γm.*

Proof. Let R be a resolution refutation. An input variable x_i is defined to be *R-analyzed* if every one of the $(m+1)$-clauses associated with x_i is used in the refutation R. Obviously it will suffice to prove that at least γ input variables are R-analyzed. In fact, if I is defined to equal the set of R-analyzed variables, then I implies a satisfying assignment for C.

This last fact is almost immediate. To prove it formally, we define a truth assignment τ as follows: (1) τ assigns truth values to variables y_i according to the value I assigns to C_i (2) τ assigns *True* to $x_{i,k}$ iff each clause $\{x_{i,1}\}$ and $\{\overline{x_{i,j}}, x_{i,j+1}\}$ for $1 \leq j < k$ is used in R. If I doesn't satisfy C, then τ would satisfy all the clauses used in the refutation R, which is impossible. Therefore, I is a satisfying assignment for C.■

Let us choose m sufficiently large with respect to n^2 say $m = n^3$. These two Lemmas establish that $C \mapsto \Gamma_C$ is an A-reduction; namely, it is easy to see that the constructed reduction transforms a sufficiently large gap $g(n)$ between hard and easy instances of Circuit MMSA into a gap $g_\varepsilon(n)$ in Minimum Length Resolution Refutation, hence if Minimum Length Resolution Refutation is approximable with some factor $f(n)$ then Circuit MMSA is approximable with $O(f(n^{O(1)}))$. This proves Theorem 5.3.1.

5.4 Main Results for Frege Systems

In this section we prove the existence of A-reduction from the Circuit MMSA to the Minimum (Step) Length Frege Proof problem. We prove it for both tree-like and dag-like versions. This will imply the hardness of approxiomation of Minimum (Step) Length Frege Proof within $2^{\log^{1-o(1)} n}$.

5.4.1 Preliminaries

We next define the notion of "active" formulas in a proof, which will be useful for proving lower bounds on the lengths of proofs. Recall that an inference in a proof must be a substitution instance of an axiom scheme, i.e., each inference must be of the form

$$\frac{A_1\sigma \quad \cdots \quad A_k\sigma}{A_{k+1}\sigma} \quad \text{(I)}$$

Consider a particular occurrence of a formula C as a subformula of a formula $A_i\sigma$ in the inference (I). If the principal connective of C is present already in the formula A_i, then we say C is *active* w.r.t. the inference (I). Otherwise, C occurs as a (not necessarily proper) subformula of $x\sigma$ for some variable x, and C is not active w.r.t. inference (I).

If a formula C has some occurrence in a proof P which is active with respect to some inference of P, then C is said to be *active* in P. (The terminology is potentially confusing: it is important to note that an active formula of P may never occur as a formula in the proof P, but instead only as a subformula of formulas in P.)

The next theorem lets us obtain a lower bound on the length of P, $|P|$, in terms of the lengths of the active formulas of P.

Theorem 5.4.1 (see [Bus95b]) *Let $\mathbb{F}$ be a Frege proof system. There is a constant ϵ such that if P is a Frege proof and we let φ range over active formula-occurrences in P, then*

$$|P| \geq \epsilon \cdot \sum_{\varphi} |\varphi|$$

Proof. A formula can be viewed as a tree with nodes labeled by connectives from L. The *depth* of a formula is defined to equal the height of this tree, namely the maximum number of connectives along any branch of the tree. Let d equal the maximum depth of the depths of the formulas which occur in the inference schemes of $\mathbb{F}$, and set $\epsilon = (1/d)$. Clearly, any active occurrence of a formula in P must have its principal connective at distance at $d-1$ from the root of the formula's tree. Thus, any given single symbol a occurring in P can occur inside at most d active occurrences of

subformulas. From this, we immediately get

$$d \cdot |P| \geq \sum_{\varphi} |\varphi|$$

and the theorem follows immediately.■

As mentioned earlier, the *step-length* of a proof P is equal to the number of steps or inferences (counting axioms as nullary inferences) in the proof. There is a linear relationship between the number of formulas active in P and the step length of P; namely,

Theorem 5.4.2 *Let $\mathbb{F}$ be a Frege proof system. Then there is a constant ϵ so that if P is a proof and m is the number of distinct active formulas in P, then the step-length of P is $\geq \epsilon m$.*

Proof. We let α equal the maximum number of subformulas that can be active in any given formula in P. For instance, if every inference scheme has depth bounded by d and r is the maximum arity of connectives in the language of F, then $\alpha = \sum_{i=0}^{d} r^i$ works. Clearly Theorem 5.4.2 is true with $\epsilon = 1/\alpha$.■

It is an interesting (albeit trivial) observation that if no formula is repeated in P, then the number of steps in P is also linearly upper bounded by the number of distinct formulas active in P.

5.4.2 Hardness of Approximation for Frege Systems

Theorem 5.4.3 *There is an A-reduction from the Circuit MMSA problem to the Minimum (Step) Length Frege Proof problems. This reduction works for both tree-like and dag-like inferences.*

Together with Theorem 5.2.2 this yields

Corollary 5.4.4 *If $\mathbf{P} \neq \mathbf{NP}$, then there is no polynomial time algorithm which can approximate Minimum (Step) Length Frege Proof to within $2^{\log^{1-o(1)} n}$ factor.*

These hardness results apply to both dag-like and tree-like Frege Systems.

The outline of our proof looks like the following: Suppose we have circuit C with inputs $x_1, \ldots, x_k$ given as a set of instructions $x_i := x_{j_{i,1}} OP x_{j_{i,2}}$ for $i \in \{k+1, \ldots, n\}$, where OP is $\wedge$ or $\vee$, where $j_{i,1}, j_{i,2} < i$ and where x_n is the output node. We construct a tautology

$$\psi_C = \left(\bigwedge_{i=k+1}^{n} ((x_{j_{i,1}} OP x_{j_{i,2}}) \to x_i) \right) \to x_n,$$

where $x_1, \ldots, x_k$ are replaced with some hard "independent" tautologies $\tau_1, \ldots, \tau_k$. Then we claim that the complexity of inference of ψ is about $\gamma(C)$ multiplied by the complexity of a single "hard" tautology τ_i (the intuition is that we need to spend "a lot" to infer some $\gamma(C)$ tautologies of $\tau_1, \ldots, \tau_k$ which force C to true and then we need "few" steps to infer ψ by evaluation of our circuit).

We use the construction of [Bus95a] to define this set of hard tautologies

Definition 5.4.5 *Let $p_0, p_1, \ldots$ be propositional variables. Let $\bot$ be the contradiction $p_0 \wedge \neg p_0$, and let τ_i^0 be the tautology $(\neg p_i \vee p_i)$. Define*

$$\tau_i^\ell = \underbrace{(\bot \vee (\bot \vee \cdots (\bot \vee \tau_i^0) \cdots))}_{\ell \text{ times}}$$

Note that $|\tau_i^l| = O(\ell)$.

The problem is that our language probably doesn't contain the connectives $\vee, \wedge, \neg$. Thus we need to express it with help of the connectives we have. Let us now fix some Frege proof system $\mathbb{F}$, with language L. We wish to construct L-formulas $\tau_i^{\ell,L}$ which are analogues of the formulas τ_i^ℓ. To do this with similar size bounds, we need the following lemma:

Lemma 5.4.6 (Reckhow [Rec76]) *There are L-formulas $NOT(x,z)$, $AND(x,y,z)$, $OR(x,y,z)$, and $IMP(x,y,z)$ such that*

(1) *$NOT(x,z)$ contains one occurrence of x, and $AND(x,y,z)$, $OR(x,y,z)$, and $IMP(x,y,z)$ contain exactly one occurrence of each of x and y.*

(2) *The four formulas represent the Boolean functions* $\neg x$, $(x \wedge y)$, $(x \vee y)$ *and* $(x \to y)$*; in particular, the truth values of the formulas are independent of the truth value of* z.

Proof. The side variable z acts merely as a placeholder whose truth value is irrelevant: in fact, if the language L contains a constant symbol, then the use of the variable z is unnecessary. We shall assume that the constant symbols $\top$ and $\bot$ are included in L; this may be assumed without loss of generality since the symbols $\top$ and $\bot$ may be replaced everywhere by L-formulas equivalent to the formulas $(z \vee \neg z)$ and $(z \wedge \neg z)$, respectively.

Let N be an L-formula containing only the variable x which represents the propositional function $\neg x$; N exists since L is a complete set of connectives. If N contains n occurrences of x, we write $N = N(x, x, \ldots, x)$ with each occurrence of x separately indicated. Let $\top^i$ denote a vector of i occurrences of $\top$, and define $\bot^i$ similarly. By choice of N, $N(\top^n)$ has value *False* and $N(\bot^n)$ has value *True*. Therefore, there is some $0 \leq i < n$ such that $N(\top^i, \bot^{n-i})$ has value *True* and $N(\top^{i+1}, \bot^{n-i-1})$ has value *False*. Thus, we can take $NOT(x)$ to be the formula $N(\top^i, x, \bot^{n-i-1})$.

To prove the remainder of the lemma, it will suffice to find an L-formula $X(x, y)$ which has one occurrence of each of x and y, and which has appearing in its truth table either three values *True* and one value *False*, or three values *False* and one value *True*. (Note that conjunction, disjunction and implication are three of the eight propositional functions whose truth table has this property.) This will suffice since *AND*, *OR* and *IMP* can be readily defined from such a formula X and from *NOT*.

Let A be an L-formula containing only the variables x and y which represents the propositional function $(x \wedge y)$. Assume $A = A(x, \ldots, x, y, \ldots, y)$ has m occurrences of x and n occurrences of y, each occurrence separately indicated. Define $A_{i,\top}$ to be the formula $A(\top^i, \bot^{m-i}, \top^n)$ and $A_{i,\bot}$ to be $A(\top^i, \bot^{m-i}, \bot^n)$. Now $A_{m,\top}$ and $A_{m,\bot}$ have different truth values, and $A_{0,\top}$ and $A_{0,\bot}$ have the same truth value. Clearly, there is a value $0 \leq i < m$ so that $A_{i,\top}$ and $A_{i,\bot}$ have the same truth value, but so that

$A_{i+1,\top}$ and $A_{i+1,\bot}$ have different truth values. Fix such an i and let $B = B(x, y, \ldots, y)$ be the formula $A(\top^i, x, \bot^{m-i-1}, y, \ldots, y)$. Note that B has one occurrence of x and n occurrences of y, each indicated separately. Let $B_i(x)$ be the formula $B(x, \top^i, \bot^{n-i})$. From the definition of B, the multiset of the four truth values of $B_0(\top)$, $B_0(\bot)$, $B_n(\top)$ and $B_n(\bot)$ contains either three *Trues* and one *False*, or one *True* and three *Falses*. Therefore, there is some value $0 \leq j < n$ so that the multiset of the truth values of $B_j(\top)$, $B_j(\bot)$, $B_{j+1}(\top)$ and $B_{j+1}(\bot)$ enjoys the same property. Letting $X(x, y)$ be the formula $B(x, \top^j, y, \bot^{n-j-1})$ gives the desired formula.■

For simplicity of notation, we shall henceforth suppress mentioning the occurrences of the side variable z.

Definition 5.4.7 *If φ is a formula over the basis $\{\neg, \wedge, \vee, \rightarrow\}$, then its* L-translation, *φ^L, is the L-formula obtained by replacing the connectives $\neg$, $\wedge$, $\vee$, and $\rightarrow$ with the formulas NOT, AND, OR and IMP in the obvious way. Because of the condition that x and y occur at most once in the formulas NOT, AND, OR and IMP, the size $|\varphi^L|$ of φ^L is $O(|\varphi|)$.*

We write $\tau_i^{\ell,L}$ to denote $(\tau_i^{\ell})^L$; thus $|\tau_i^{\ell,L}| = O(\ell)$.

The next lemma will be used to give upper bounds on the lengths of $\mathbb{F}$-proofs.

Lemma 5.4.8 ([Bus95a]) *$\tau_i^{\ell,L}$ has an $\mathbb{F}$-proof of length $O(\ell^2)$ (step-length $O(\ell)$).*

This lemma is proved by noting that $\mathbb{F}$ can derive successively $\tau_i^{0,L}$, $\tau_i^{1,L}$, $\tau_i^{2,L}$, etc., until $\tau_i^{\ell,L}$ is derived. □

Suppose that we are given the circuit C. Let us take the formula ψ_C defined in the begining and translate to our language:

$$\psi_C^L = \left[\left(\bigwedge_{i=k+1}^{n} ((x_{j_{i,1}} O P x_{j_{i,2}}) \rightarrow x_i) \right) \rightarrow x_n \right]^L,$$

where $x_1, \ldots, x_k$ are replaced with $\tau_1^{m,L}, \ldots, \tau_k^{m,L}$ for sufficiently large m (it will be enough to let $m = n^3$). We claim that ψ_C^L is the reduction of Circuit MMSA instance

C to Minimum (Step) Length Frege Proof problem. We are going to show that minimal proof length of ψ_C^L is about $\gamma(C) \cdot m^2$ (step-length of ψ_C^L is about $\gamma(C) \cdot m$).

Lemma 5.4.9 *ψ_C^L has a tree-like $\mathbb{F}$-proof of length $O(\gamma(C) \cdot m^2 + m \cdot n^2 \log n)$ (step-length $O(\gamma(C) \cdot m + n \log n)$).*

Proof. Suppose that $\langle v_j \rangle$ is the minimal satisfying assignment for C and that I is the index set of all v_i such that $v_i = \top$, $|I| = \gamma(C)$. First we infer all the tautologies $\tau_i^{m,L}$, $i \in I$ in length $\gamma(C) \cdot m^2$ (step length $\gamma(C) \cdot m$) by Lemma 5.4.8.

Let $r_1, r_2, \ldots, r_s = n$ be the increasing sequence of indices such that $x_{r_1}, x_{r_2}, \ldots, x_{r_s}$ are made true by the truth assignment $\vec{v}$. Then, it is straightforward to construct a tree-like Frege proof of the formulas

$$\psi_\ell = \left[\left(\bigwedge_{i=k+1}^{n} ((x_{j_{i,1}} OP x_{j_{i,2}}) \to x_i) \right) \to \bigwedge_{i=1}^{\ell} x_{r_i} \right]^L$$

which proceeds by proving these successively with $\ell = 1, 2, 3, \ldots, s$ using $\tau_i^{m,L}$, $i \in I$ as basis. To make our inference tree-like on each step ℓ we independently prove formulas

$$\left[\left(\bigwedge_{i=k+1}^{n} ((x_{j_{i,1}} OP x_{j_{i,2}}) \to x_i) \right) \to ((x_{j_{r_{\ell+1},1}} OP x_{j_{r_{\ell+1},2}}) \to x_{r_{\ell+1}}) \right]^L,$$

$$\left[\left(\bigwedge_{i=1}^{\ell} x_{r_i} \right) \to (x_{j_{r_{\ell+1},1}} OP x_{j_{r_{\ell+1},2}}) \right]^L$$

Together with ψ_ℓ it will infer $\psi_{\ell+1}$ in $O(1)$ steps. Since the conjunctions all have $O(n)$ inputs and since the formulas have symbol-length $O(m \cdot n)$ if one is careful and uses balanced conjunctions [Bon91], on each step the inference $\psi_{\ell+1}$ from ψ_ℓ can have length $O(m \cdot n \log n)$ (step-length $\log n$).

Finally the overall proof has length at most $O(\gamma(C) \cdot m^2 + m \cdot n^2 \log n)$ (step-length $O(\gamma(C) \cdot m + n \log n)$). Lemma follows.■

Definition 5.4.10 *Let ψ be a formula and consider a particular $\tau_i^{\ell,L}$. We define $\psi/(\tau_i^{\ell,L})$ to be the formula obtained from ψ by replacing every occurrence of $\tau_i^{\ell,L}$ as a subformula of ψ with the formula $\bot$. Note, that if $\ell_1 < \ell_2$ then $\tau_i^{\ell_2,L}/(\tau_i^{\ell_1,L})$ is not a tautology anymore.*

If P is a proof, then we define $P/(\tau_i^{\ell,L})$ be a sequence of formulas obtained by replacing every ψ in P with $\psi/(\tau_i^{\ell,L})$. Note that $P/(\tau_i^{\ell,L})$ will not, in general, be a valid proof.

Lemma 5.4.11 *Let P be a proof of ψ and suppose that the formula $\tau_i^{\ell,L}$ is not active in P. Then, $\psi/(\tau_i^{\ell,L})$ is a tautology.*

The proof of Lemma 5.4.11 is immediate by the fact that if $\tau_i^{\ell,L}$ is not active in P, then $P/(\tau_i^{\ell,L})$ is identical to P, except that it may use a different substitution of formulas for variables, and hence it is still a valid proof. □

Lemma 5.4.12 *Any $\mathbb{F}$-proof of ψ_C^L has length $\Omega(\gamma(C)\cdot m^2)$ (step-length $\Omega(\gamma(C)\cdot m)$).*

Proof. Suppose that $\gamma(C) = p$. Let P be some $\mathbb{F}$-proof of ψ_C^L. Let I be the index set of all i such that $\tau_i^{\ell,L}$ is active in P for all $0 \leq \ell \leq m$. For $j \notin I$, choose j_r so that $\tau_j^{j_r,L}$ is not active in P. By Lemma 5.4.11 we have that, after replacing all $\tau_j^{j_r,L}$ with $\bot$ for all $j \notin I$, the formula ψ_C^L remains a tautology. Hence the circuit C is satisfied by the truth assignment corresponding to the characteristic function of I, hence $|I| \geq p$. Thus $|P| = \Omega(p \cdot m^2)$ by Theorem 5.4.1 and the fact that the total length of the formulas $\tau_i^{\ell,L}$, for $i \in I$, $0 \leq \ell \leq m$, is $\Omega(p \cdot m^2)$. Analogously by Theorem 5.4.2 the step length of P is $\Omega(p \cdot m)$.■

Altogether Lemmas 5.4.9, 5.4.12 imply that the mapping $C \mapsto \psi_C^L$ is A-reduction. Theorem 5.4.3 follows.

Remark. All of our hardness results for approximating step-length and symbol-length of Frege proofs also apply to extended Frege systems. To see this, it suffices to note that all the upper and lower bounds on the length of Frege proofs which were obtained in the proof of Theorem 5.4.3, also apply to extended Frege proofs.

Of course it is obvious that the upper bounds apply since every Frege proof is an extended Frege proof. The lower bounds also apply, since Theorems 5.4.1 and 5.4.2 are also true for extended Frege systems ([Bus95b]).

5.5 Hardness results for long proofs

In the previous sections we proved that it is **NP**-hard to approximate the minimal propositional proof length by any constant factor, and that if **NP** is not in **P** then minimum proof-length cannot be approximated (in polynomial time) within a $2^{\log^{1-o(1)} n}$ factor. The tautologies used in the proofs of these results had "short" proofs (or refutations); that is, proofs whose length is polynomial in the size of the formula. However, if $\mathbf{NP} \neq \mathbf{coNP}$, then for any proof system $\mathcal{S}$, there are tautologies whose shortest $\mathcal{S}$-proof is of super-polynomial length. It is therefore interesting to ask whether better non-approximability results can be achieved when the proof lengths are not bounded, and when the run time of the algorithm is required to be polynomial time in the length of the input formula only.

The following simple intuition implies that in this case, no polynomial time algorithm can guarantee a polynomial time approximation for the shortest refutation of a given unsatisfiable formula, unless $\mathbf{NP} \not\subseteq \mathbf{P/poly}$:[2]

Given an input formula ψ of length n, reduce it to a formula $\varphi = \psi \wedge \eta$, such that η is unsatisfiable but its shortest refutation is larger than the refutation of any unsatisfiable formula of length n by a super-polynomial factor. Then ψ is satisfiable iff on input φ, a supposed polynomially bounded approximation algorithm returns a number smaller than the size of the shortest refutation of φ. This implies a polynomial time circuit for recognizing SAT. To make the above argument formal, we need few more definitions.

Definition 5.5.1 *For a proof system $\mathcal{S}$ and an unsatisfiable formula φ, $\min_{\mathcal{S}}(\varphi)$ is the minimum length of a refutation of φ in $\mathcal{S}$. For an integer n, $MAX_{\mathcal{S}}(n) =$*

[2] We present the results in terms of finding short refutations of unsatisfiable formulas, but equivalent definitions and results are easily obtained for finding short proofs of tautologies.

$\max\{\min_{\mathcal{S}}(\varphi)\}$, *where* φ *ranges over all unsatisfiable formulas of length* $\leq n$.

We say that a non-decreasing function f has *super-polynomial growth* if for every polynomial r, $f(n) > r(n)$ for almost all positive integers n. The function f has a *smooth* super-polynomial growth if in addition there is a constant D such that for each n there is $1 < d < D$ such that $f(n^d) > f^d(n)$. [If we write $f(n) = n^{e(n)}$, then the first condition states that $e(n)$ is not bounded from above, and the second condition states that for each n there is m, $n < m < n^D$, such that $e(m) > e(n)$.]

Assume, for simplicity, that $\mathcal{S}$ contains the connective $\wedge$. Formulas ψ and η are said to be *disjoint* if their underlying sets of variables are disjoint.

Theorem 5.5.2 *Assume that* **NP** $\not\subseteq$ **P/poly**, *and let* $\mathcal{S}$ *be a proof system which satisfies:*

1. *For every pair of disjoint formulas* ψ *and* η, *where* η *is unsatisfiable, the following holds:*

   (a) *If* ψ *is unsatisfiable, then* $\min_{\mathcal{S}}(\psi \wedge \eta) < \min_{\mathcal{S}}(\psi) + r(|\psi| + |\eta|)$ *for some (fixed) polynomial* r.

   (b) *If* ψ *is satisfiable, than* $\min_{\mathcal{S}}(\psi \wedge \eta) \geq \min_{\mathcal{S}}(\eta)$;

2. $MAX_{\mathcal{S}}(n)$ *has a smooth super-polynomial growth.*

Then for any polynomial q, *there is no polynomial time* q*-approximation algorithm for the minimum length proof in* $\mathcal{S}$.

Observe that property 1 above holds trivially for all proof systems mentioned in this paper. Property 2 is known to hold for resolution, since in this case $MAX_{\mathcal{S}}(n) < 3^n$ for all n, and by [BP96], for each n there is an e, $1 < e < 3$ s.t. $MAX_{\mathcal{S}}(n^e) > 2^{\frac{n^e}{40}}$, thus property 2 holds for $D = 3$ for all large enough n's. We conjecture it to be valid for any known proof system in which the proof lengths are not polynomially bounded.

Proof. We show that the existence of a polynomial time q-approximation algorithm, AL, for $\mathcal{S}$, implies polynomial time circuits for solving SAT.

Let j be such that $q(n) < n^j$ for almost all n, and let D be the constant guaranteed by the smooth super-polynomial growth of $MAX_{\mathcal{S}}$. Since $MAX_{\mathcal{S}}$ has super-polynomial growth, for all large enough n it holds that $r(n + n^{2jD}) < MAX_{\mathcal{S}}(n)$. Fix an integer n_0 for which this inequality holds. Since the super-polynomial growth of $MAX_{\mathcal{S}}$ is smooth, there is a number d, $2j \leq d \leq 2jD$, such that $[MAX_{\mathcal{S}}(n_0)]^d < MAX_{\mathcal{S}}(m)$, where $m = {n_0}^d$. Let η_m be a formula of size $\leq m$ such that $\min_{\mathcal{S}}(\eta_m) = MAX_{\mathcal{S}}(m)$. An input formula ψ of size n_0 is reduced to $\varphi = \psi \wedge \eta_m$, where the variables of η_m are disjoint from these of ψ (note that φ is unsatisfiable and its size is polynomial in that of ψ). We claim that ψ is unsatisfiable if and only if AL on input φ will output a number $k < MAX_{\mathcal{S}}(m)$. To see this, observe that if ψ is unsatisfiable, then by property (1a) above, $\min_{\mathcal{S}}(\varphi) \leq \min_{\mathcal{S}}(\psi) + r(|\psi| + |\eta_m|) < 2MAX_{\mathcal{S}}(n_0)$. Hence, by the assumption on AL, AL must produce an output $k < (2MAX_{\mathcal{S}}(n_0))^j < MAX_{\mathcal{S}}(m) = \min_{\mathcal{S}}(\eta_m)$. On the other hand, if ψ is satisfiable, then, by property (1b), $\min_{\mathcal{S}}(\varphi) \geq \min_{\mathcal{S}}(\eta_m) = MAX_{\mathcal{S}}(m)$.■

Chapter 6

Automatizability of Resolution and Fixed Parameterized Complexity

6.1 Introduction

The analysis of potential usefulness of proof search heuristics and automated theorem proving procedures based on a proof system P amounts (on the theoretical level) to the following two basic questions:

Question 1. Which theorems in principle possess efficient P-proofs?

Question 2. How to find the optimal (or, at least, nearly the optimal) proof of a given theorem in P?

The traditional proof complexity mostly dealt, and still deals with the first question. Recently, however, there has been a growing interest in the second one, too. An additional motivation to study the complexity of finding optimal proofs comes from deep connections with efficient interpolation theorems; we refer the reader to the excellent survey [BP98] for more details about this and also as to a good starting point for learning more about propositional proof complexity in general.

One convenient framework for the theoretical study of Question 2 was proposed in [BPR00]. Namely, they called a proof system P *automatizable* if there exists a deter-

ministic algorithm A which, given a tautology τ, returns its P-proof in time polynomial *in the size of the shortest P-proof of τ*. The definition of a quasi-automatizable proof system is given in the same way, but we only require the algorithm A to run in time which is quasi-polynomial (in the same parameter).

One advantage of this definition is that it allows us to completely disregard the first basic question on the *existence* of efficient P-proofs and indeed concentrate on *finding* efficient proofs *provided* they exist. In particular, the notion of automatizability makes perfect sense for those (weak) proof systems for which hard tautologies are already known. Moreover, the weaker is our system the more likely it seems to be automatizable. One possible explanation of this phenomenon comes from the connection between automatizability and efficient interpolation (every automatizable proof system has efficient interpolation, and the property of having efficient interpolation is indeed anti-monotone w.r.t. the strength of the system). Anyway, given this connection, the results from [KP95], [BPR00] imply that Extended Frege and TC^0-Frege proof systems respectively are not automatizable assuming some widely believed cryptographic assumptions. [BDG$^+$99] extended the latter result to bounded-depth Frege but under a much stronger assumption.

We are primarily interested in the automatizability of Resolution and tree-like Resolution. It is worth noting that both systems possess efficient interpolation, therefore their non-automatizability can not be proved via techniques similar to [KP95, BPR00, BDG$^+$99]. Nonetheless, [Iwa97] proved that it is **NP**-hard to find the shortest resolution refutation. In Chapter5 we prove that if $\mathbf{P} \neq \mathbf{NP}$ then the length of the shortest resolution refutation can not be approximated to within a factor $2^{\log^{1-o(1)} n}$ (both for general and tree-like Resolution).

In the opposite direction, [BP96] observed that tree-like Resolution is quasi-automatizable. Thus, it is unlikely to show that this system is not automatizable modulo $\mathbf{P} \neq \mathbf{NP}$ conjecture, because it would imply quasi-polynomial algorithms for **NP** (in case of general Resolution this goal seems also tricky at the moment because there is only one known example [BG99] for which proof search algorithm of [BW99]

is working in more than quasi-polynomial time). Therefore, for any method of establishing non-automatizability which is applicable to tree-like Resolution, one needs to invoke some complexity framework in which the asymptotics $n^{O(1)}$ and $n^{\log n}$ are essentially different.

One natural example of such a framework is *parameterized complexity* by Downey and Fellows (see [DF98]) in which algorithms working in time $f(k)n^{O(1)}$ and n^k are considered different from the point of effectiveness (here k is an integer input parameter that is supposed to be an "arbitrarily large" constant). In this paper we prove that neither Resolution nor tree-like Resolution is automatizable unless the class $\mathbf{W[P]}$ lying very high in the hierarchy of parameterized problems is fixed-parameter tractable by a randomized algorithm with one-sided error (Theorem 6.4.3). Our proof goes by a reduction from the optimization problem MINIMUM MONOTONE CIRCUIT SATISFYING ASSIGNMENT (MMCSA for short) whose decision version is complete for the class $\mathbf{W[P]}$. An alternative hardness assumption is that there is no *deterministic* fixed-parameter algorithm which *approximates* MMCSA within any constant factor (Theorem 6.4.2). It is worth noting in this connection that we were able to relate to each other the hardness of finding *exact* and *approximate* solutions for MMCSA without using the PCP Theorem (see the proof of Theorem 6.4.3). This result can be interesting in its own.

The paper is organized as follows. Section 6.2 contains necessary preliminaries and definitions, in Section 6.3 we present our core reduction from MMCSA to automatizability of Resolution, and in Section 6.4 we use (sometimes non-trivial) self-improving techniques to prove our main results, Theorems 6.4.2 and 6.4.3. The paper is concluded with some open problems in Section 6.5.

6.2 Preliminaries

6.2.1 Resolution and automatizability

Recall that *Resolution* operates with clauses and has one rule of inference called *resolution rule*:

$$\frac{A \vee x \qquad B \vee \bar{x}}{A \vee B}.$$

A resolution proof is *tree-like* if its underlying graph is a tree. A *resolution refutation* of a CNF τ is a resolution proof of the empty clause from the clauses appearing in τ.

For an unsatisfiable CNF τ, $S(\tau)$ $[S_T(\tau)]$ is the minimal size of its resolution refutation [tree-like resolution refutation, respectively]. Clearly, $S(\tau) \leq S_T(\tau)$.

For a CNF τ, let $n(\tau)$ be the overall number of distinct variables appearing in it, and let $|\tau|$ be the overall number of occurrences of variables in τ, i.e., $|\tau| \stackrel{\text{def}}{=} \sum_{C \in \tau} w(C)$. For an unsatisfiable CNF τ, $w(\tau \vdash \emptyset)$ will denote the minimal width of its resolution refutation.

We will recall the general definition of automatizability from [BPR00] for the special cases of Resolution and tree-like Resolution.

Definition 6.2.1 *Resolution [tree-like Resolution] is* (quasi-)automatizable *if there exists a deterministic algorithm A which, given an unsatisfiable CNF τ, returns its resolution refutation [tree like resolution refutation, respectively] in time which is (quasi-)polynomial in $|\tau| + S(\tau)$ $[|\tau| + S_T(\tau)$, respectively].*

Remark 6.2.2 *Note that we do not require that all clauses from τ must necessarily appear in its refutations, therefore, we can not a priori expect the inequality $S(\tau) \geq |\tau|$. This is why we must introduce the term $|\tau|$ into the bound on the running time of A when adapting the general definition of automatizability from [BPR00] to the case of Resolution.*

6.2.2 Parameterized complexity and MMCSA problem

We refer the reader to [DF98, Fel01] for a good general introduction to the topic of parameterized complexity.

Definition 6.2.3 ([DF98, Definition 2.4]) *The class* **FPT** *(that stands for* fixed parameter tractable*) of parameterized problems consists of all languages* $L \subseteq \Sigma^* \times \mathbb{N}$ *for which there exists an algorithm* Φ*, a constant* c *and a recursive function* $f : \mathbb{N} \longrightarrow \mathbb{N}$ *such that:*

1. *the running time of* $\Phi(\langle x, k\rangle)$ *is at most* $f(k) \cdot |x|^c$*;*
2. $\langle x, k\rangle \in L$ *iff* $\Phi(\langle x, k\rangle) = 1$.

Thus an algorithm is considered to be feasible if it works in time polynomial in n and $f(k)$, where k is supposed to be much smaller than n, and f is an arbitrarily large (recursive) function. A similar feasibility issue arises in the theory of polynomial time approximation schemes (PTAS) for **NP**-hard problems: assume that we have an algorithm that approximates a given problem within arbitrary error $\epsilon > 0$ working in time $n^{O(1/\epsilon)}$. Is it possible to get rid of $1/\epsilon$ in the exponent and do it in time $f(1/\epsilon)n^{O(1)}$ (the algorithms which obey the latter bound on the running time are called EPTAS, efficient polynomial time approximation schemes)?

It turns out that this question is highly related to the fixed parameter tractability. Namely, the existence of EPTAS for a given problem implies an *exact* algorithm for the corresponding fixed parameter version (see [Baz95, CT97]).

To study the complexity of parameterized problems, special *parameterized* reductions (that preserve the property of being in **FPT**) are used. For any t the parameterized problem WEIGHTED t-NORMALIZED SATISFIABILITY is defined by restricting the classical SATISFIABILITY to a certain class of Boolean formulas, the parameter k bounding Hamming weights of the satisfying assignment we are searching for. The complexity class **W[t]** consists of all problems that can be reduced to WEIGHTED-t-NORMALIZED-SATISFIABILITY via parameterized reductions,

and the class **W[P]** (where P stands for polynomial) includes all problems reducible to WEIGHTED CIRCUIT SATISFIABILITY. These definitions lead to the following *parameterized hierarchy*, in which every inclusion is believed to be strict:

$$\mathbf{FPT} \subseteq \mathbf{W[1]} \subseteq \mathbf{W[2]} ... \subseteq \mathbf{W[P]}.$$

In our paper we construct a randomized parameterized reduction from the automatizability of Resolution to the following optimization problem (MMCSA in what follows) that was introduced in Section 5.

Monotone Minimum Circuit Satisfying Assignment:
Instance: A monotone circuit C in n variables over the basis $\{\wedge, \vee\}$.
Solution: An assignment $a \in \{0,1\}^n$ such that $C(a) = 1$.
Objective function: $\gamma(a)$ defined as the number of a_i's which are equal to 1.

By $\gamma(C)$ we will denote the optimum value $\gamma(a)$ of a solution a for an instance C of MMCSA. We need the following easy observation ("self-improvement").

Proposition 6.2.4 *For every fixed integer $d > 0$ there exists a poly-time computable function π which maps monotone circuits into monotone circuits and such that $\gamma(\pi(C)) = \gamma(C)^d$ for all C.*

Proof. For a circuit C define a "composition" circuit

$$C \star C \stackrel{\text{def}}{=} C(C(x_1, ..., x_n), C(x_1, ..., x_n), ..., C(x_1, ..., x_n)).$$

It is easy to verify that $\gamma(C \star C) = \gamma^2(C)$. It is left to iterate this construction d times.■

The decision version of MMCSA was considered in [DF98] (under the name WEIGHTED MONOTONE CIRCUIT SATISFIABILITY) in the context of parameterized complexity and was shown to be complete in the class **W[P]**.

In order to formulate our main result, we need to introduce the obvious hybrid of the classes **R** and **FPT**:

Definition 6.2.5 *The class* **FPR** *of parameterized problems consists of all languages* $L \subseteq \Sigma^* \times \mathbb{N}$ *for which there exists a probabilistic algorithm* Φ*, a constant* c *and a recursive function* $f : \mathbb{N} \longrightarrow \mathbb{N}$ *such that:*

1. *the running time of* $\Phi(\langle x, k\rangle)$ *is at most* $f(k) \cdot |x|^c$*;*
2. *if* $\langle x, k\rangle \in L$ *then* $\mathbf{P}[\Phi(\langle x, k\rangle) = 1] \geq 1/2$*;*
3. *if* $\langle x, k\rangle \notin L$ *then* $\mathbf{P}[\Phi(\langle x, k\rangle) = 1] = 0$.

6.3 Main reduction from MMCSA to automatizability of Resolution

This section is entirely devoted to the proof of the following lemma.

Lemma 6.3.1 *There exists a poly-time computable function* τ *which maps any pair* $\langle C, 1^m\rangle$*, where* C *is a monotone circuit and* m *is an integer, to an unsatisfiable CNF* $\tau(C, m)$ *such that:*

$$S_T(\tau(C, m)) \leq |C| \cdot m^{O(\min\{\gamma(C), \log m\})}$$

and

$$S(\tau(C, m)) \geq m^{\Omega(\min\{\gamma(C), \log m\})}. \tag{6.1}$$

We begin the proof of Lemma 6.3.1 with describing CNFs that form the main building block in $\tau(C, m)$ and establishing their necessary properties. From now on fix a monotone circuit C in n variables $p_1, \ldots, p_n$. Let ℓ be another parameter, and let $\mathcal{A} \subseteq \{0, 1\}^\ell$. We will call vectors from $\mathcal{A}$ (usually represented as columns) *admissible* and call an 0-1 matrix with ℓ rows $\mathcal{A}$*-admissible* if all its columns are so. Consider the following combinatorial principle $\mathcal{P}_{C,\mathcal{A}}$:

$\mathcal{P}_{C,\mathcal{A}}$: *every* $(\ell \times n)$ *0-1* $\mathcal{A}$*-admissible matrix* A *contains a row* $i \in [\ell]$ *such that* $C(a_{i1}, a_{i2}, \ldots, a_{in}) = 1$.

Let us formulate one sufficient condition for $\mathcal{P}_{C,\mathcal{A}}$ to be true in the real world.

Definition 6.3.2 *$d_1(\mathcal{A})$ is the maximal d such that for every d vectors from $\mathcal{A}$ there exists a position $i \in [\ell]$ in which all these vectors have 1.*

$d_1(\mathcal{A})$ can be also easily characterized in terms of minimum covers. Namely, if we associate with every $\begin{pmatrix} a_1 \\ \vdots \\ a_\ell \end{pmatrix} \in \mathcal{A}$ the subset $\{i \in [\ell] \mid a_i = 0\}$ of $[\ell]$ then $d_1(\mathcal{A})+1$ is exactly the minimal number of such sets needed to cover the whole $[\ell]$.

Lemma 6.3.3 *If $\gamma(C) \leq d_1(\mathcal{A})$ then $\mathcal{P}_{C,\mathcal{A}}$ is true.*

Proof. Let A be an $(\ell \times n)$ 0-1 $\mathcal{A}$-admissible matrix. Let $a = (a_1, \ldots, a_n)$ be such that $C(a_1, \ldots, a_n) = 1$ and $\gamma(a) = \gamma(C)$. Let $\mathcal{A}_0 \stackrel{\text{def}}{=} \left\{ \begin{pmatrix} a_{1j} \\ \vdots \\ a_{\ell j} \end{pmatrix} \middle| \, a_j = 1 \right\}$. Since $|\mathcal{A}_0| \leq \gamma(a) = \gamma(C) \leq d_1(\mathcal{A})$, there exists $i \in [\ell]$ such that $a_{ij} = 1$ whenever $a_j = 1$. This means $a_{ij} \geq a_j$ for all $j \in [n]$ and implies $C(a_{i1}, \ldots, a_{in}) = 1$ since A is monotone.■

The proof of Lemma 6.3.3 suggests that the optimal propositional proof of the principle $\mathcal{P}_{C,\mathcal{A}}$ should work by the exhaustive search through all $|\mathcal{A}|^{\gamma(C)}$ possible placements of admissible vectors to the columns $\{j \mid a_j = 1\}$ and thus have size roughly $|\mathcal{A}|^{\gamma(C)}$. Our task is to find an encoding of (the negation of) $\mathcal{P}_{C,\mathcal{A}}$ as a CNF so that we can prove tight upper and lower bounds on $S_T(\tau(C,\mathcal{A}))$ and $S(\tau(C,\mathcal{A}))$ of (roughly) this order. The encoding will also involve auxiliary surjective functions $F_1, \ldots, F_n : \{0,1\}^s \longrightarrow \mathcal{A}$, $f_1, \ldots, f_\ell : \{0,1\}^s \longrightarrow [r]$, where f_is are possibly partial, and s, r are some new parameters. They will be used for encoding vectors from $\mathcal{A}$ and elements from another finite set of r *controls* by binary strings of length s. These complications will be needed for the lower bounds purposes.

In order to not obstruct the proof with irrelevant details, we will define our CNFs $\tau(C, \mathcal{A}, \vec{F}, \vec{f})$ and establish their necessary properties in a situation which is more general than what will be actually needed for completing the proof of Lemma 6.3.1 (see page 124). If the reader prefers, he/she may think during the course of the proof

that $\ell = m$ and $\mathcal{A}$ is an arbitrary set of vectors such that $d_1(\mathcal{A}) \geq \Omega(\log m)$ and (see Definition 6.3.7) $d_0(\mathcal{A}) \geq \Omega(\log m)$. Furthermore, $r = \log m$, $s = O(\log m)$ and F_j, f_i will be $(\log m)$-surjective in the sense of Definition 6.3.6.

Definition 6.3.4 *Let $C(p_1, \ldots, p_n)$ be a monotone circuit, $\mathcal{A} \subseteq \{0,1\}^\ell$ be a set of vectors and $F_1, \ldots, F_n : \{0,1\}^s \longrightarrow \mathcal{A}$, $f_1, \ldots, f_\ell : \{0,1\}^s \longrightarrow [r]$ be surjective functions, where f_is are possibly partial. For every $j \in [n]$ and $\nu \in [s]$ we introduce a propositional variable x_j^ν, for every $i \in [\ell]$ and $\nu \in [s]$ introduce a variable y_i^ν, and for every $i \in [\ell]$, every "control" $c \in [r]$ and every vertex v of the circuit C introduce the variable z_{iv}^c.*

For $j \in [n]$ and $\vec{a} \in \mathcal{A}$, let us denote by $[Column_j = \vec{a}]$ the predicate $F_j(x_j^1, \ldots, x_j^s) = \vec{a}$. Likewise, for $i \in [\ell]$ and $c \in [r]$, let $[Control_i = c]$ denote the predicate $f_i(y_i^1, \ldots, y_i^s) = c$.

The CNF $\tau(C, \mathcal{A}, \vec{F}, \vec{f})$ consists of all clauses that result from the expansion of the following Boolean predicates as CNFs:

$$(y_i^1, \ldots, y_i^s) \in dom(f_i), \text{ for all } i \in [\ell]; \tag{6.2}$$

$$\left.\begin{array}{c} ([Column_j = \vec{a}] \wedge [Control_i = c]) \supset z_{i,p_j}^c \\ \text{for all } \vec{a} \in \mathcal{A},\ i \in [\ell] \text{ such that } a_i = 1 \text{ and all } j \in [n],\ c \in [r]; \end{array}\right\} \tag{6.3}$$

$$\left.\begin{array}{c} \left([Control_i = c] \wedge (z_{i,v'}^c * z_{i,v''}^c)\right) \supset z_{iv}^c \\ \text{for all } i \in [\ell],\ c \in [r] \text{ and all internal nodes } v \\ \text{corresponding to the instruction } v \leftarrow v' * v'',\ * \in \{\wedge, \vee\}; \end{array}\right\} \tag{6.4}$$

$$[Control_i = c] \supset \bar{z}_{i,v_{\text{fin}}}, \text{ where } v_{\text{fin}} \text{ is the output node of } C. \tag{6.5}$$

Informally, every assignment to x-variables encodes an $\mathcal{A}$-admissible matrix A and every assignment to y-variables satisfying (6.2) encodes controls $c_1, \ldots, c_\ell$ (one control per row). The axioms (6.3) and (6.4) inductively state that $z_{iv}^{c_i}$ is greater or equal than the result of computation of the node v on the ith row of the matrix A. Thus, $\tau(C, \mathcal{A}, \vec{F}, \vec{f})$ is unsatisfiable (for arbitrary surjective $\vec{F}, \vec{f}$) if and only if $P_{C,\mathcal{A}}$ is true.

Remark 6.3.5 *If we are interested in general Resolution only, the construction of $\tau(C, \mathcal{A}, \vec{F}, \vec{f})$ can be a bit simplified. Namely, we do not need the controls, and we can encode the value attained by a node v on the ith row in the same simple way as we did with $[Column_j = \vec{a}]$. Then lower bounds from Lemma 6.3.8 will still go through, but the upper bound will become problematic for the tree-like case (although, it will still hold for general Resolution).*

The only thing we need from $\vec{F}, \vec{f}$ is that their ontoness is preserved under restricting not too many variables.

Definition 6.3.6 *We say that an onto (possibly partial) function $g : \{0,1\}^k \longrightarrow D$ is* r-surjective *if for any restriction ρ with $|\rho| \leq r$ the function $g|_\rho$ is still onto.*

Finally, for the lower bound purposes we need a notion dual to $d_1(\mathcal{A})$.

Definition 6.3.7 *$d_0(\mathcal{A})$ is the maximal d such that for every d positions $i_1, \ldots, i_d \in [\ell]$ there exists $\vec{a} \in \mathcal{A}$ such that $a_{i_1} = \ldots = a_{i_d} = 0$.*

Now we are ready to formulate our main technical lemma.

Lemma 6.3.8 *Let C be a monotone circuit in n variables, $\mathcal{A} \subseteq \{0,1\}^\ell$, $|\mathcal{A}| = m$ and $F_1, \ldots, F_n : \{0,1\}^s \longrightarrow \mathcal{A}$, $f_1, \ldots, f_\ell : \{0,1\}^s \longrightarrow [r]$ be r-surjective functions, where f_is are possibly partial; ℓ, m, r, s arbitrary integer parameters. Then the following bounds hold.*

a) *If $\gamma(C) \leq d_1(\mathcal{A})$ then $S_T(\tau(C, \mathcal{A}, \vec{F}, \vec{f})) \leq O\left(|C| \cdot 2^{s(\gamma(C)+1)}\right)$.*

b) $w(\tau(C, \mathcal{A}, \vec{F}, \vec{f}) \vdash \emptyset) \geq \frac{r}{2} \cdot \min\{\gamma(C), d_0(\mathcal{A})\}$.

c) $S(\tau(C, \mathcal{A}, \vec{F}, \vec{f})) \geq \exp\left(\Omega\left(\frac{\tau^2}{s} \cdot \min\{\gamma(C), d_0(\mathcal{A})\}\right)\right)$.

Proof of Lemma 6.3.8. Part a). By formalizing the proof of Lemma 6.3.3. Let $k \stackrel{\text{def}}{=} \gamma(C)$ and $a_1, \ldots, a_n$ be such that $C(a_1, \ldots, a_n) = 1$ and $\gamma(a) = k$. Assume for simplicity that $a_1 = \ldots = a_k = 1$, $a_{k+1} = \ldots = a_n = 0$. Fix arbitrary admissible vectors $\vec{a}_1 \stackrel{\text{def}}{=} \begin{pmatrix} a_{i1} \\ \vdots \\ a_{\ell 1} \end{pmatrix}, \ldots, \vec{a}_k \stackrel{\text{def}}{=} \begin{pmatrix} a_{ik} \\ \vdots \\ a_{\ell k} \end{pmatrix}$ and, using the inequality $d_1(\mathcal{A}) \geq k$, pick up an arbitrary $i \in [\ell]$ (depending in general on $\vec{a}_1, \ldots, \vec{a}_k$) such that $a_{i1} = a_{i2} = \ldots = a_{ik} = 1$. We want to infer from $\tau(C, \mathcal{A}, \vec{F}, \vec{f})$ all clauses in the CNF expansion of

$$[\text{Column}_1 \neq \vec{a}_1] \vee \ldots \vee [\text{Column}_k \neq \vec{a}_k] \vee [\text{Control}_i \neq c] \tag{6.6}$$

for all $c \in [r]$. It is fairly obvious how to do this efficiently in general Resolution. Namely, let V be the set of all nodes of the circuit C that are evaluated to 1 by the assignment $(1^k, 0^{n-k})$. Then we may proceed by induction on the construction of C and subsequently infer

$$([\text{Column}_1 = \vec{a}_1] \wedge \ldots \wedge [\text{Column}_k = \vec{a}_k] \wedge [\text{Control}_i = c]) \supset z^c_{iv}$$

for all $v \in V$ until we reach v_{fin}.

In order to get a *tree-like* proof, however, we should employ a dual (top-down) strategy. Namely, enumerate the set of vertices in some order which is consistent with the topology of C: $V = \langle v_1 = p_1, v_2 = p_2, \ldots, v_k = p_k, v_{k+1}, v_{k+2}, \ldots, v_t = v_{\text{fin}} \rangle$; all wires between vertices in V go from the left to the right. Then, by a reverse induction on $\mu = t, t-1, \ldots, k$ we infer (all clauses in the CNF expansion of) $[\text{Control}_i = c] \supset \left(\bar{z}^c_{1,v_1} \vee \ldots \vee \bar{z}^c_{\mu,v_\mu}\right)$. For $\mu = t$ this is (a weakening of) (6.5), and for the inductive step we resolve with the appropriate axiom in (6.4). Finally, when we descend to $[\text{Control}_i = c] \supset \left(\bar{z}^c_{1,p_1} \vee \ldots \vee \bar{z}^c_{k,p_k}\right)$, we consecutively resolve with the corresponding axioms (6.3) to get rid of $\bar{z}^c_{j,p_j}$ and arrive at (6.6). Clearly, this resolution inference of every individual clause in (6.6) is tree-like and has size $O(|C|)$.

Finally, for every $i \in [\ell]$, every clause in the variables $\{y_i^\nu \mid 1 \leq \nu \leq s\}$ appears in one of the CNFs resulting from the predicates $\{[\text{Control}_i \neq c] \mid c \in [r]\}$ or in (6.2), and every clause in the variables $\{x_j^\nu \mid 1 \leq \nu \leq s\}$ appears in one of $[\text{Column}_j \neq \vec{a}_j]$. This gives us an obvious tree-like refutation of the set of clauses (6.6) that has size $O(2^{s(k+1)})$. Combining this refutation with previously constructed inferences of (6.6) from $\tau(C, \mathcal{A}, \vec{F}, \vec{f})$, we get the desired upper bound.

Part b). We follow the general strategy proposed in [BW99]. Let Row_i be the set of axioms in $\tau(C, \mathcal{A}, \vec{F}, \vec{f})$ that correspond to this particular row i. For a clause C, let $\mu(C)$ be the smallest cardinality of $I \subseteq [\ell]$ such that $\{\text{Row}_i \mid i \in I\}$ (semantically) implies C. As in the previous proofs, $\mu(C)$ is subadditive, that is $\mu(C) \leq \mu(C_1) + \mu(C_2)$ whenever C is obtained from C_1, C_2 via a single application of the resolution rule. It is also obvious that $\mu(A) = 1$ for any axiom A in $\tau(C, \mathcal{A}, \vec{F}, \vec{f})$.

We claim that $\mu(\emptyset) > d_0(\mathcal{A})$. Indeed, fix any $I \subseteq [\ell]$ with $|I| \leq d_0(\mathcal{A})$. We need to construct an assignment that satisfies all axioms in $\{\text{Row}_i \mid i \in I\}$. Pick $\vec{a}$ accordingly to Definition 6.3.7 in such a way that $\forall i \in I(a_i = 0)$. Assign every x_j^ν to α_j^ν, where $\alpha_j^1, \ldots, \alpha_j^s$ is an arbitrary vector such that $F_j(\alpha_j^1, \ldots, \alpha_j^s) = \vec{a}$; assign y_i^ν in an arbitrary way with the only requirement that they satisfy (6.2), and assign all z-variables to 0. This assignment will satisfy all axioms in $\{\text{Row}_i \mid i \in I\}$ which proves $\mu(\emptyset) > d_0(\mathcal{A})$.

Thus, any resolution refutation of $\tau(C, \mathcal{A}, \vec{F}, \vec{f})$ must contain a clause C with $\frac{1}{2}d_0(\mathcal{A}) \leq \mu(C) \leq d_0(\mathcal{A})$, and we only need to show that this implies $w(C) \geq \frac{r}{2} \cdot \min\{k, d_0(\mathcal{A})\}$. Fix $I \subseteq [\ell]$ such that $\frac{1}{2}d_0(\mathcal{A}) \leq |I| \leq d_0(\mathcal{A})$, $\{\text{Row}_i \mid i \in I\}$ semantically implies C and I is minimal with this property.

If for every $i \in [I]$ the clause C contains at least r variables amongst $\{y_i^\nu \mid \nu \in [s]\} \cup \{z_{iv}^c \mid c \in [r],\ v \text{ is a node}\}$, we are done. Thus, suppose that this is not the case for some $i_0 \in I$. Fix an arbitrary assignment α that satisfies all axioms in $\{\text{Row}_i \mid i \in I \setminus \{i_0\}\}$ and falsifies C (such an assignment exists due to the minimality of I).

Let J_0 consist of those $j \in [n]$ for which the clause C contains at least r variables from $\{x_j^\nu \mid \nu \in [s]\}$. If $|J_0| \geq k$, we are also done. If this is not the case, we will show how to alter the assignment α so that it will satisfy all axioms in $\{\text{Row}_i \mid i \in I\}$

(including Row_{i_0}) but still falsify C, and this will give us the contradiction.

According to Definition 6.3.7, there exists $\vec{a} \in \mathcal{A}$ such that $a_i = 0$ for all $i \in I$. We alter α as follows.

1. For every $j \notin J_0$ we, using that F_j is r-surjective, change the values of the variables $\{ x_j^\nu \mid \nu \in [s] \}$ not appearing in C in such a way that $F_j(x_j^1, \ldots, x_j^s) = \vec{a}$.

2. Pick any control $c_0 \in [r]$ such that no variable $z_{i_0,v}^{c_0}$ appears in C. Using the fact that f_{i_0} is r-surjective, change the values of variables $\{ y_{i_0}^\nu \mid \nu \in [s] \}$ not appearing in C in such a way that $f_{i_0}(y_{i_0}^1, \ldots, y_{i_0}^s) = c_0$. Finally, re-assign every $z_{i_0,v}^{c_0}$ to the value computed by the node v on the characteristic vector of the set J_0. Note that $z_{i_0,p_j}^{c_0}$ is set to 1 for $j \in J_0$ whereas $z_{i_0,v_{\mathrm{fin}}}^{c_0}$ is set to 0 since $|J_0| < \gamma(C)$.

With the remarks made in this description, it is easy to see that the altered assignment will satisfy all axioms in $\{\mathrm{Row}_i \mid i \in I\}$. Also it will falsify C since we have not touched variables appearing in C. This contradiction with the fact that $\{\mathrm{Row}_i \mid i \in I\}$ implies C completes the proof of part b).

Part c). We apply the standard argument with width-reducing restrictions (cf. [BP96]). For doing this we observe that the CNFs of the form $\tau(C, \mathcal{A}, \vec{F}, \vec{f})$ well behave with respect to certain restrictions. Namely, let $d \leq r$ and $R \subseteq [r]$ be an arbitrary set of controls. Denote by $\mathcal{R}_{d,R}$ the set of all restrictions that arbitrarily assign to a Boolean value d variables in every one of the groups $\{ x_j^\nu \mid \nu \in [s] \}, \{ y_i^\nu \mid \nu \in [s] \}$ with $j \in [n], i \in [\ell]$ as well as all variables z_{iv}^c with $c \notin R$. Then it is easy to see that for $\rho \in \mathcal{R}_{d,R}$, every non-trivial clause in $\tau(C, \mathcal{A}, \vec{F}, \vec{f})|_\rho$, after a suitable re-enumeration of variables and controls, contains a subclause from $\tau(C, \mathcal{A}, \vec{F}|_\rho, (\vec{f}|_\rho)|_R)$ (the partial function $(f_i|_\rho)|_R$ is obtained from $f_i|_\rho$ by restricting its domain to $\{y_i \mid f_i|_\rho(y_i) \in R\}$ and range to R).

Pick now $\boldsymbol{\rho}$ uniformly and at random from $\mathcal{R}_{r/2,\boldsymbol{R}}$, where $\boldsymbol{R}$ is picked at random from $[r]^{r/2}$. Then $F_j|_{\boldsymbol{\rho}}, (f_i|_{\boldsymbol{\rho}})|_{\boldsymbol{R}}$ will be $(r/2)$-surjective. Therefore, by the already

proven part b), for every refutation P of $\tau(C, \mathcal{A}, \vec{F}, \vec{f})$, $\boldsymbol{\rho}(P)$ will contain a clause of width $\Omega\left(r \cdot \min\{\gamma(C), d_0(\mathcal{A})\}\right)$ with probability 1.

It is easy to see, however, that every clause of this width is killed by $\boldsymbol{\rho}$ with probability $\geq 1 - \exp\left(-\Omega\left(\frac{r^2}{s} \cdot \min\{\gamma(C), d_0(\mathcal{A})\}\right)\right)$. Therefore, the size of P must be at least $\exp\left(\Omega\left(\frac{r^2}{s} \cdot \min\{\gamma(C), d_0(\mathcal{A})\}\right)\right)$ since otherwise a random restriction $\boldsymbol{\rho}$ would have killed all such clauses with non-zero probability, which is impossible.

Lemma 6.3.8 is completely proved.■

Proof of Lemma 6.3.1. Our construction of $\tau(C, m)$ proceeds in polynomial time as follows.

1. We may assume w.l.o.g. that m is a prime. Let P_m be the $(m \times m)$ 0-1 *Paley matrix* given by $a_{ij} = 1$ if and only if $j \neq i$ and $(j - i)$ is a quadratic residue mod m. Set $\ell \stackrel{\text{def}}{=} m$, and let $\mathcal{A} \subseteq \{0,1\}^m$ consist of all columns of P_m. Then $|\mathcal{A}| = m$ and $d_0(\mathcal{A}), d_1(\mathcal{A}) \geq \frac{1}{4}\log m$ (see e.g. [Alo95]).

2. Fix any poly-time computable $\mathbb{F}_2$-linear code $L \subseteq \{0,1\}^{h\lceil \log m \rceil}$ of dimension $\lceil \log m \rceil$ and with the minimal distance $\geq \lceil \log m \rceil$, where $h > 0$ is an absolute constant. Then the linear mapping $G : \{0,1\}^{h\lceil \log m \rceil} \longrightarrow \{0,1\}^{\lceil \log m \rceil}$ which is dual to the inclusion $L \to \{0,1\}^{h\lceil \log m \rceil}$ is $\lceil \log m \rceil$-surjective. Set $r \stackrel{\text{def}}{=} \lceil \log m \rceil$ and $s \stackrel{\text{def}}{=} h\lceil \log m \rceil$. Consider arbitrary (poly-time computable) surjective mappings $\Pi : \{0,1\}^{h\lceil \log m \rceil} \longrightarrow \mathcal{A}$, $\pi : \{0,1\}^{h\lceil \log m \rceil} \longrightarrow [r]$, and let $F_j \stackrel{\text{def}}{=} G \circ \Pi, f_i \stackrel{\text{def}}{=} G \circ \pi$ for all i, j.

3. Construct $\tau(C, \mathcal{A}, \vec{F}, \vec{f})$. Note that the size of this CNF is polynomial in $|C|, \ell, 2^s$ which is polynomial in $|C|, m$ due to our choice of parameters.

At this point, Lemma 6.3.8 c) already implies (6.1) for $\tau(C, \mathcal{A}, \vec{F}, \vec{f})$. The only remaining problem is that a priori we do not have the condition $\gamma(C) \leq d_1(\mathcal{A})$ needed for part a) of Lemma 6.3.8. We circumvent this by the following trick.

Let τ_m be a fixed unsatisfiable (poly-time constructible) CNF with $S(\tau_m), S_T(\tau_m) = m^{\theta(\log m)}$ (for example, one can take the Pigeonhole principle with $\sqrt{\log m}$ pigeons) and such that its set of variables is disjoint from the set of variables of $\tau(C, \mathcal{A}, \vec{F}, \vec{f})$. We finally set $\tau(C, m) \stackrel{\text{def}}{=} \tau(C, \mathcal{A}, \vec{F}, \vec{f}) \wedge \tau_m$.

Since both $\tau(C, \mathcal{A}, \vec{F}, \vec{f})$ and τ_m satisfy the lower bound in (6.1), Efficient Interpolation Property for Resolution implies that $\tau(C, m)$ satisfies this bound, too.

Lastly, if $\gamma(C) \leq \frac{1}{4}\log m$ then, since $d_1(\mathcal{A}) \geq \frac{1}{4}\log m$, we can apply Lemma 6.3.8 a) to get the required upper bound $S_T(\tau(C, m)) \leq |C| \cdot m^{O(\gamma(C))}$. If, on the other hand, $\gamma(C) \geq \frac{1}{4}\log m$, the necessary upper bound $S_T(\tau(C, m)) \leq m^{O(\log m)}$ simply follows from the upper bound for τ_m. This completes the proof of Lemma 6.3.1.■

6.4 Self-improvement and main results

In this section we combine Lemma 6.3.1 with some non-trivial self-improvement technique. This will show non-automatizability of Resolution and tree-like Resolution modulo plausible assumptions about the intractability of the parameterized hierarchy. First we need to get rid of the dummy parameter m in the statement of Lemma 6.3.1.

Lemma 6.4.1 *If either Resolution or tree-like Resolution is automatizable then there exists an absolute constant $h > 0$ and an algorithm Φ working on pairs $\langle C, k\rangle$, where C is a monotone circuit and k is an integer such that:*

1. *the running time of $\Phi(\langle C, k\rangle)$ is at most $\exp(O(k^2)) \cdot |C|^{O(1)}$;*
2. *if $\gamma(C) \leq k$ then $\Phi(\langle C, k\rangle) = 1$;*
3. *if $\gamma(C) \geq hk$ then $\Phi(\langle C, k\rangle) = 0$.*

Proof. Combining the reduction in Lemma 6.3.1 with an automatizing algorithm for either Resolution or tree-like Resolution, we get an integer-valued function $S(C, m)$ computable in time $\left(|C| \cdot m^{\min\{\gamma(C), \log m\}}\right)^{h_0}$ and such that

$$m^{\epsilon \cdot \min\{\gamma(C), \log m\}} \leq S(C, m) \leq \left(|C| \cdot m^{\min\{\gamma(C), \log m\}}\right)^{h_1}$$

for some absolute constants $\epsilon, h_0, h_1 > 0$. Set the constant h in the statement in such a way that

$$h^2 > \frac{h_1}{\epsilon}(h+1). \tag{6.7}$$

Our algorithm Φ works as follows. We set

$$m \stackrel{\text{def}}{=} \begin{cases} 2^{hk} & \text{if } k \geq \sqrt{\log |C|}, \\ 2^{h(\log |C|/k)} & \text{otherwise.} \end{cases}$$

Note that in any case $\log m \geq hk$. Φ simulates $\left(|C| \cdot m^k\right)^{h_0}$ steps in the computation of $S(C,m)$, outputs 1 if the computation halts within this time and its result $S(C,m)$ satisfies the inequality $S(C,m) \leq \left(|C| \cdot m^k\right)^{h_1}$, and outputs 0 in all other cases.

Our choice of m ensures that $m^k \leq \max\{2^{k^2}, |C|\}^h$, which implies property 1).

The upper bounds we have imposed on the running time and the output value of the algorithm Φ are the same as the bounds known for the underlying algorithm computing $S(C,m)$, the only difference is that we have replaced $\gamma(C)$ by k. This observation implies property 2).

Finally, since $\log m \geq hk$, $\gamma(C) \geq hk$ implies that $S(C,m) \geq m^{\epsilon kh}$, and elementary calculations show that, along with (6.7), this gives us $S(C,m) > \left(|C| \cdot m^k\right)^{h_1}$. Thus, if $\gamma(C) \geq hk$, the algorithm Φ outputs the value 0.

Lemma 6.4.1 is proved.■

Combining this lemma with Proposition 6.2.4, we immediately get our first main result.

Theorem 6.4.2 *If either Resolution or tree-like Resolution is automatizable then for any fixed $\epsilon > 0$ there exists an algorithm Φ working on monotone circuits C which runs in time $\exp\left(\gamma(C)^{O(1)}\right) \cdot |C|^{O(1)}$ and approximates the value of $\gamma(C)$ to within a factor $(1+\epsilon)$.*

Proof. Firstly we extract from Lemma 6.4.1 an algorithm which meets the bound on the running time and achieves the ratio of approximation h. For doing that we con-

secutively run the algorithm Φ from that lemma on the inputs $\langle C, 1\rangle, \ldots, \langle C, k\rangle, \ldots$, and output the first value k for which we get the answer 0.

Combining this algorithm with the self-improving reduction from Proposition 6.2.4 (for $d = \lceil \frac{1}{\epsilon} \ln h \rceil$), we get approximating algorithm with the required properties.■

In the established terminology, what we have seen so far under the assumption of automatizability of (tree-like) Resolution is a PTAS (polynomial time approximation scheme) for MMCSA in the context of parameterized complexity (the latter referring to the term $\exp\left(\gamma(C)^{O(1)}\right)$ in the bound on the running time). As we noted in Section 6.2.2, in this context *efficient* polynomial time approximation schemes lead to *exact* algorithms. The task of converting an arbitrary PTAS into an EPTAS seems to be hopeless in general even in the context of parameterized complexity. We nonetheless can perform it for the specific problem MMCSA using a much trickier self-improvement construction. This construction which appears in the proof of the next theorem might be of independent interest.

Recall Definition 6.2.5 of the class **FPR**.

Theorem 6.4.3 *If either Resolution or tree-like Resolution is automatizable then* $\mathbf{W[P]} \subseteq co - \mathbf{FPR}$.

Proof. Let C be a monotone circuit in n variables and k be an integer such that

$$10 \leq k \leq \epsilon(\log n / \log\log n)^2 \tag{6.8}$$

for a sufficiently small constant $\epsilon > 0$ (we will remark later how to get rid of this condition). Our goal is to construct in polynomial time a randomized monotone circuit $\pi(C, k)$ and an integer $\alpha(k)$ (deterministically depending only on k) such that α is recursive and the following conditions hold:

$$\gamma(C) \leq k \implies \mathbf{P}[\gamma(\pi(C,k)) \leq \alpha(k)] = 1; \tag{6.9}$$

$$\gamma(C) \geq k+1 \implies \mathbf{P}[\gamma(\pi(C,k)) \geq 2\alpha(k)] \geq 1/2. \tag{6.10}$$

Firstly we apply to C the reduction from Proposition 6.2.4 with $d = 2$. Redenoting k^2 back to k, we may assume w.l.o.g. that in (6.10) we have a stronger premise:

$$\gamma(C) \geq k + 2\sqrt{k} \implies \mathbf{P}[\gamma(\boldsymbol{\pi}(C,k)) \geq 2\alpha(k)] \geq 1/2. \tag{6.11}$$

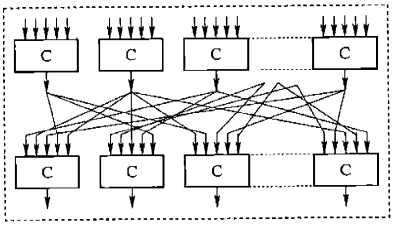

Figure 6-1: One layer of $\boldsymbol{\pi}(C, N, d)$

Now comes our main reduction. Let N, d be two parameters (to be specified later). The randomized circuit $\boldsymbol{\pi}(C, N, d)$ in (nN) variables consists of d layers. Each layer consists of N independent copies of the circuit C (see Fig. 6-1); thus, it has (nN) inputs and N outputs. We connect input nodes at the $(i+1)$st level to output nodes at the ith level at random. Finally, we pick up an arbitrary output node at the last dth level and declare it to be the output of the whole circuit $\boldsymbol{\pi}(C, N, d)$.

Clearly, this construction is polynomial in $|C|, N, d$. Also, an obvious induction on d shows that $\gamma(\boldsymbol{\pi}(C, N, d)) \leq \gamma(C)^d$ with probability 1. In order to get a lower bound on $\gamma(\boldsymbol{\pi}(C, N, d))$, we need the following easy lemma.

Lemma 6.4.4 *Let $\boldsymbol{\chi} : [N] \times [n] \longrightarrow [N]$ be a randomly chosen function, and k, a be any parameters. Then $\mathbf{P}\left[\exists V \in [N]^k (|\boldsymbol{\chi}(V \times [n])| \leq kn - a)\right] \leq N^k \cdot \left(\frac{4k^2n^2}{N}\right)^a$.*

Proof of Lemma 6.4.4. This event takes place if and only if there exist $V \in [N]^k$ and disjoint $D_1, \ldots, D_r \subseteq V \times [n]$ such that $|D_1|, \ldots, |D_r| \geq 2$, $\sum_{i=1}^r (|D_i| - 1) = a$ and $\boldsymbol{\chi}|_{D_i} = \text{const}$ for all $i \in [r]$. Since the two first properties imply $\sum_{i=1}^r |D_i| \leq 2a$, the overall number of all choices of $\langle D_1, \ldots, D_r \rangle$ does not exceed $N^k \cdot (2kn)^{2a}$. On the other hand, for every fixed choice we have

$$\mathbf{P}[\boldsymbol{\chi}|_{D_1} = \text{const}, \ldots, \boldsymbol{\chi}|_{D_r} = \text{const}] = \frac{N^r}{N^{(\sum_{i=1}^r |D_i|)}} = N^{-a}.$$

Lemma 6.4.4 follows.■

Now we can complete the description of our reduction. Namely, we set $N \stackrel{\text{def}}{=} n^3$, $d \stackrel{\text{def}}{=} \sqrt{k}$ and let $\pi(C,k) \stackrel{\text{def}}{=} \pi(C,n^3,\sqrt{k})$, $\alpha(k) \stackrel{\text{def}}{=} k^{\sqrt{k}}$.

(6.9) follows from the remark made above.

In order to check (6.11), denote by $\chi_i : [N] \times [n] \longrightarrow [N]$ the function used for connecting input nodes at the $(i+1)$st level of $\pi(C,n^3,\sqrt{k})$ to the output nodes at the ith level. Let $k_i \stackrel{\text{def}}{=} (k+\sqrt{k})^{d-i}$. Let us call $\pi(C,N,d)$ *bad* if for at least one of these functions χ_i there exists a set V of circuits at the $(i+1)$st level such that $|V| = k_{i+1}$ and $|\chi_i(V \times [n])| \leq k_{i+1}(n-\sqrt{k})$. Using Lemma 6.4.4 and (6.8), we get the bound

$$\begin{aligned}\mathbf{P}[\pi(C,N,d) \text{ is bad }] &\leq \sum_{i=1}^{d-1} N^{k_{i+1}} \cdot \left(\frac{4k_{i+1}^2 n^2}{N}\right)^{\sqrt{k}\cdot k_{i+1}} \\ &= \sum_{i=1}^{d-1} \left(\frac{4k_{i+1}^2}{n^{1-3/\sqrt{k}}}\right)^{\sqrt{k}\cdot k_{i+1}} \leq 1/2.\end{aligned}$$

On the other hand, it is easy to see by induction on $i = d,\ldots,1$ that if $\gamma(C) \geq k+2\sqrt{k}$ and $\pi(C,N,d)$ is good then every satisfying assignment a should satisfy at least k_i output nodes at the ith level. Indeed, the base $i=d$ is obvious. For the inductive step, assume that a satisfies the output nodes of a set V of circuits at the $(i+1)$th level, $|V| = k_{i+1}$. Then at least $(k+2\sqrt{k})\cdot k_{i+1}$ *input* nodes to these circuits should be satisfied. Since χ_i is good, there are at most $\sqrt{k}\cdot k_{i+1}$ collisions between the $(k+2\sqrt{k})\cdot k_{i+1}$ wires leading to these nodes from the ith level. Therefore, at least $(k+2\sqrt{k})\cdot k_{i+1} - \sqrt{k}\cdot k_{i+1} = k_i$ output nodes at the ith level should be satisfied.

In particular, at the first level we will have $\geq (k+\sqrt{k})^{d-1}$ satisfied circuits and $\geq (k+2\sqrt{k})\cdot(k+\sqrt{k})^{\sqrt{k}-1} > 2\alpha(k)$ satisfied input nodes. This completes the proof that our probabilistic reduction $\pi(C,k)$ has the properties (6.9), (6.11).

Now we finish the proof of Theorem 6.4.3. Suppose that either Resolution or tree-like Resolution is automatizable. Since WEIGHTED MONOTONE CIRCUIT SATISFIABILITY is $\mathbf{W}[\mathbf{P}]$-complete (see [DF98, Chapter 13]), we only have to show that the language $\{\langle C,k\rangle \mid \gamma(C) \leq k\}$ is in $co-\mathbf{FPR}$. Given an input $\langle C,k\rangle$ we check

the condition (6.8). If it is violated, we apply the straightforward brute-force algorithm with running time $O(|C| \cdot n^k) \leq |C| \cdot f(k) \cdot n^9$. Otherwise we simply combine our probabilistic reduction $\langle \pi, \alpha \rangle$ with the deterministic algorithm for deciding whether $\gamma(\pi(C,k)) \leq \alpha(k)$ or $\gamma(\pi(C,k)) \geq 2\alpha(k)$ provided by Theorem 6.4.2. Theorem 6.4.3 is completely proved.■

6.5 Open Problems

The main problem left open by this paper is whether general Resolution is quasi-automatizable. Since the width algorithm by Ben-Sasson and Wigderson [BW99] finds a resolution refutation of any unsatisfiable CNF τ in time $n^{O(w(\tau \vdash \emptyset))}$, a negative solution to this problem must involve a construction of a broad and "tractable" family of CNF τ for which $S(\tau)$ is much smaller than $2^{w(\tau \vdash \emptyset)}$. Such families are not so easy to come by (e.g. our techniques involve showing the *opposite* in the proof of Lemma 6.3.8 c)), and it seems the only tautologies of this kind we have at the moment come from the somewhat isolated example [BG99].

We were not able to de-randomize the proof of Lemma 6.4.4. In the terminology of Chapters 3, 4, we need explicit constructions of $(N \times N)$ 0-1 matrices which would be $(k, n, n - O(k))$-expanders for $n \geq N^{\Omega(1)}$ and an arbitrary function $k = k(N)$ tending to infinity. Explicit constructions based on Ramanujan graphs seem to give only $(k, n, n - k^{1+\epsilon})$-expanders for any *fixed* ϵ which is not sufficient for our purposes. Can we weaken the hardness assumption in Theorem 6.4.3 to $\mathbf{W[P]} \neq \mathbf{FPT}$ by an explicit construction of better expanders (or by using any other means)?

Chapter 7

Conclusion and Recent Developments

In the first part of the thesis we give more questions than answers. In particular, the hardness of constructed tautologies is believed to have a bigger potential, than illustrated in the thesis. Thus, more research in this direction is anticipated in the future. Since the preliminary version of [ABSRW00] was disseminated, many open problems asked there have been solved, and many other related developments have occurred.

The principles expressing that Nisan-Wigderson generators are not onto studied in this paper bear a striking similarity to the pigeonhole principle PHP_n^m (with the same meaning of the parameters m, n). At the time the research was done, one of the most interesting open problems, both for NW-generators and for PHP_n^m, was to break through the quadratic barrier $m \geq n^2$ for (at least) the resolution size. This has been solved in both contexts.

The pigeonhole principle PHP_n^m was the first to yield. Raz [RanRaz02] proved exponential lower bounds on the size of its resolution refutations when $m \gg n$. Razborov [Raz02a] gave a simpler proof of a somewhat better bound that also holds for the more general functional onto version of this principle.

The quadratic barrier for pseudorandom generators did not stand for much longer. Razborov [Raz02b] constructed Nisan generators (that is, when the base functions

g_i are $\mathbb{F}_2$-linear forms) that allow $m \geq n^{\Omega(\log n)}$ output bits and are exponentially hard not only for Resolution, but also for its extensions $\mathrm{Res}(\epsilon \log n)$ (operating with $(\epsilon \log n)$-DNF instead of clauses) and PCR when $char(F) \neq 2$.

Another question asked in the earlier version of [] was whether any structural theory of pseudorandom generators is possible in the framework of proof complexity. In particular, we asked if it is possible to formulate and prove any reasonable statement that would say, possibly in a restricted way, that the composition of hard generators is hard (for a given propositional proof system). This was satisfactorily answered by Krajíček [Kra02] who showed that this is indeed the case provided hardness is replaced by a stronger notion of s-iterability (inspired by the so-called counter-example interpretation).

It was also conjectured in the earlier version that such a composition result might provide an alternate approach to the quadratic barrier problem (but for more complicated generators). This has indeed turned out to be the case. Krajíček [Kra02] proved (independently of [Raz02b]) that our generator from Section 3.4 can be iterated with itself once, which immediately allowed him to get as many as $m = n^{3-\epsilon}$ output bits. The Nisan generator from [Raz02b] turned out to be particularly suitable for Krajíček's notion of s-iterability, and it can be composed with itself exponentially many times while preserving hardness. In this way [Raz02b] constructed a function generator with $m = 2^{n^\epsilon}$ outputs which is hard for $\mathrm{Res}(\epsilon \log n)$ and for PCR with $char(F) \neq 2$. Along the lines outlined in the discussion after Example 3, this immediately implied that neither of these systems possess efficient proofs of $\mathbf{NP} \not\subseteq \mathbf{P}/poly$ (the same conclusion for Resolution had already followed from [RanRaz02, Raz02a]).

Finally, [CRVW02] took an important step toward constructing explicit expanders (called there and in [Raz02b] "lossless") with very good expansion properties (even if not sufficient yet for many of our purposes).

There also has been a future progress on better understanding the automatizability of weak systems since the results of Chapters 5 and 6 were published. Pavel Pudlák [Pud01] suggested the following natural question: does there exist an extension of

Resolution which is automatizable. He considers the problem from the point of view of canonical NP-pairs, which is an important characteristics of propositional systems. This direction was further investigated by Atserias and Bonet [AB02], who showed that there exists an automatizable extension of Resolution if and only if the system Res(2) has feasible interpolation property. Also, [AB02] constructed more examples of tautologies that require large refutation width, but small size in Resolution, that we have asked for in Section 6.5.

Bibliography

[Ajt83] M. Ajtai. Σ_1^1-formulae on finite structures. *Annals of Pure and Applied Logic*, 24:1–48, May 1983.

[Alo95] N. Alon. Tools from higher algebra. In *Handbook of Combinatorics, vol. II*, chapter 32. Elsevier, 1995.

[Alo98] N. Alon. Spectral techniques in graph algorithms. In C. L. Lucchesi and A. V. Moura, editors, *Lecture Notes in Computer Science* 1380, pages 206–215, Berlin, 1998. Springer-Verlag.

[AB02] Albert Atserias, Maria Luisa Bonet. On the automatizability of Resolution and Related Propositional Proof Systems. In *Proc. of Annual Conference of the European Association for Computer Science Logic CSL'02*, 569–583, 2002.

[ABFR94] J. Aspnes, R. Beigel, M. Furst, and S. Rudich. The expressive power of voting polynomials. *Combinatorica*, 14(2):1–14, 1994.

[ABMP01] M. Alekhnovich, S. Buss, S. Moran, and T. Pitassi. Minimum propositional proof length is **NP**-hard to linearly approximate. *Journal of Symbolic Logic* 66:171–191, 2001.

[ABSRW00] M. Alekhnovich, E. Ben-Sasson, A. Razborov, and A. Wigderson. Pseudorandom generators in propositional complexity. In *Proceedings of the 41st IEEE FOCS*, 2000. Journal version to appear in *SIAM Journal on Computing.*

[ABRW02] M. Alekhnovich, E. Ben-Sasson, A. Razborov, and A. Wigderson. Space complexity in propositional calculus. *SIAM Journal on Computing*, 31(4):1184–1211, 2002.

[ABSS93] S. Arora, L. Babai, J. Stern, and Z. Sweedyk. The hardness of approximate optima in lattices, codes, and systems of linear equations. *Journal of Computer and System Sciences*, 54, pp. 317–331, 1997.

[AL96] S. Arora and C. Lund. Hardness of approximations. In *Approximation Algorithms for NP-hard Problems*, D. S. Hochbaum, ed., PWS Publishing Co., Boston, 1996.

[AR01a] M. Alekhnovich and A. Razborov. Lower bounds for the polynomial calculus: non-binomial case. In *Proceedings of the 42nd IEEE Symposium on Foundations of Computer Science*, pages 190–199, 2001. Journal version to appear in *Proceedings of the Steklov Institute of Mathematics.*

[AR01b] M. Alekhnovich and A. Razborov. Resolution is Not Automatizable Unless W[P] is Tractable. In *Proc. of the 42nd IEEE FOCS*:210-219, 2001.

[Baz95] C. Bazgan. Schémas d'approximation et complexité paramétrée. Rapport de stage de DEA d'Informatique à Orsay, 1995.

[Bel96] M. Bellare Proof checking and approximation: Towards tight results. *SIGACT News*, 27, pp. 2–13, 1996.

[Bon91] M. L. Bonet. The Lengths of Propositional Proofs and the Deduction Rule. PhD thesis, U.C. Berkeley, 1991.

[Bus95a] S. R. Buss. On Gödel's theorems on lengths of proofs II: Lower bounds for recognizing k symbol provability. In *Feasible Mathematics II*, P. Clote and J. Remmel, eds., Birkhäauser-Boston, pp. 57–90, 1995.

[Bus95b] S. R. Buss. Some remarks on lengths of propositional proofs. *Archive for Mathematical Logic*, 34, pp. 377–394 (1995).

[BCS78] F. C. Bussemaker, D. M. Cvetković, and J. J. Seidel. Graphs related to exceptional root systems. In A. Hajnal and V. T. Sós, editors, *Combinatorics, Coll. Math. Soc. J. Bolyai, Vol. 18*, pages 185–191. North-Holland, Amsterdam, 1978.

[BDG+99] M. Bonet, C. Domingo, R. Gavaldá, A. Maciel, and T. Pitassi. Non-automatizability of bounded-depth Frege proofs. In *Proceedings of the 14th Annual IEEE Conference on Computational Complexity*, pages 15–23, 1999.

[BFNW93] L. Babai, L. Fortnow, N. Nisan, and A. Wigderson. BPP has subexponential time simulations unless EXPTIME has publishable proofs. *Complexity*, 3:307–318, 1993.

[BG99] M. Bonet, N. Galesi. A Study of Proof Search Algorithms for Resolution and Polynomial Calculus. In *Proceedings of the 40th IEEE FOCS*, pages 422-432, 1999.

[BGIP01] S. Buss, D. Grigoriev, R. Impagliazzo, and T. Pitassi. Linear gaps between degrees for the Polynomial Calculus modulo distinct primes. *Journal of Computer and System Sciences*, 62:267–289, 2001.

[BI99] E. Ben-Sasson and R. Impagliazzo. Random CNF's are Hard for the Polynomial Calculus. In *Proceedings of the 40th IEEE FOCS*, pages 415–421, 1999.

[BIK+94] P. Beame, R. Impagliazzzo, J. Krajicek, T. Pitassi, and P. Pudlak. Lower bounds on Hilbert's Nullstellensatz and propositional proofs. In *Thirty-fifth Annual Symposium on Foundations of Computer Science*, IEEE Press, pp. 794-806, 1994.

[BIK+96] S. Buss, R. Impagliazzo, J. Krajíček, P. Pudlák, A. Razborov, and J. Sgall. Proof complexity in algebraic systems and bounded depth Frege systems with modular counting. *Computational Complexity* **6**(3) (1996/1997), 256–298.

[BKPS98] P. Beame, R. Karp, T. Pitassi, and M. Saks. On the complexity of unsatisfiability of random k-cnf formulas. In *Proceedings of the 30th ACM Symposium on Theory of Computing*, pages 561–571, 1998.

[BP96] P. Beame and T. Pitassi. Simplified and improved resolution lower bounds. In *Proceedings, 37th Annual Symposium on Foundations of Computer Science*, Los Alamitos, California, IEEE Computer Society, pp. 274–282, 1996.

[BP98] P. Beame and T. Pitassi. Propositional proof complexity: Past, present and future. Technical Report TR98-067, Electronic Colloquium on Computational Complexity, 1998.

[BPR97] M. L. Bonet, T. Pitassi, and R. Raz. No feasible interpolation for TC^0-Frege proofs. In *Proceedings of the 38th Annual Symposium on Foundations of Computer Science*, Piscataway, New Jersey, IEEE Computer Society, pp. 264–263, 1997.

[BPR00] M. Bonet, T. Pitassi, and R. Raz. On interpolation and automatization for Frege systems. *SIAM Journal on Computing*, 29(6):1939–1967, 2000.

[BST98] P. Beame, M. Saks, and J. S. Thathachar. Time-space tradeoffs for branching programs. In *Proceedings of the 39th IEEE FOCS*, pages 254–263, 1998.

[BW99] E. Ben-Sasson and A. Wigderson. Short proofs are narrow - resolution made simple. In *Proceedings of the 31st ACM STOC*, pages 517–526, 1999.

[CEI96] M. Clegg, J. Edmonds, and R. Impagliazzo. Using the Groebner basis algorithm to find proofs of unsatisfiability. In *Proceedings of the Twenty-eighth Annual ACM Symposium on the Theory of Computing*, Association for Computing Machinery, pp. 174–183, 1996.

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

[CRVW02] M. Capalbo, O. Reingold, S. Vadhan, and A. Wigderson. Randomness conductors and constant-degree expansion beyond the degree/2 barrier. In *Proceedings of the 34th ACM Symposium on the Theory of Computing*, pages 659–668, 2002.

[CS88] V. Chvátal, E. Szemerédi. Many hard examples for resolution. *Journal of the ACM*, 35 (4):759-768, October 1988.

[CT97] M. Cesati and L. Trevisan. On the efficiency of polynomial time approximation schemes. *Information Processing Letters*, 64(4): 165–171, 1997.

[DF98] R. Downey and M. Fellows. *Parameterized Complexity.* Springer-Verlag, 1998.

[DS99] I. Dinur, S. Safra. On the Hardness of Approximating Label Cover. Manuscript, 1999.

[FSS84] M. Furst, J. B. Saxe, and M. Sipser. Parity, circuits and the polynomial time hierarchy. *Math. Syst. Theory*, 17:13–27, 1984.

[Gre00] F. Green. A complex-number Fourier technique for lower bounds on the MOD-m degree. *Computational Complexity*, 9(1):16–38, 2000.

[Gri98] D. Grigoriev. Tseitin's tautologies and lower bounds for nullstellensatz proofs. In *Proceedings of the 39th IEEE FOCS*, pages 648–652, 1998.

[Gri01] D. Grigoriev. Linear lower bounds on degrees of Postivestellensatz calculus proofs for the parity. *Theoretical Computer Science*, 259:613–622, 2001.

[GZ97] O. Goldreich and D. Zuckerman. Another proof that BPP$\subseteq$PH (and more). *Electronic Colloquium on Computational Complexity*, TR97-045, 1997.

[Fel01] M. Fellows. Parameterized complexity: new developments and research frontiers. To appear in *Aspects of Complexity*, R. Downey and E. Hirshfeldt, editors, De Gruyter, 2001.

[Hak85] A. Haken. The intractability or resolution. *Theoretical Computer Science*, 39:297–308, 1985.

[Hås86] J. Håstad. *Computational limitations on Small Depth Circuits.* PhD thesis, Massachusetts Institute of Technology, 1986.

[Iwa97] K. Iwama. Complexity of finding short resolution proofs. In *Mathematical Foundations of Computer Science*, I. Prívara and P. Ruzicka, eds., Lecture Notes in Computer Science #1295, Springer-Verlag, pp. 309–318, 1997.

[IKW01] R. Impagliazzo, V. Kabanets and A. Wigderson. In Search for an Easy Witness: Exponential time vs. probabilistic polynomial time. In Proceedings of the 16th annual IEEE Conference on Computational Complexity, pages 2-12, 2001.

[IM95] K. Iwama and E. Miyano. Intractibility of read-once resolution In *Proceedings of the Tenth Annual Conference on Structure in Complexity Theory*, Los Alamitos, California, IEEE Computer Society, pp. 29–36, 1995.

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

[IW97] R. Impagliazzo and A. Wigderson. P=BPP if E requires exponential circuits: Derandomizing the XOR Lemma. In *Proceedings of the Twenty-Ninth Annual ACM Symposium on Theory of Computing*, pages 220–229, 1997.

[Kra97] J. Krajíček. Interpolation theorems, lower bounds for proof systems, and independence results for bounded arithmetic. *Journal of Symbolic Logic*, 62(2): pp.457–486, 1997.

[Kra01a] J. Krajíček. On the weak pigeonhole principle. *Fundamenta Mathematicae*, 170(1-3):123–140, 2001.

[Kra01b] J. Krajíček. On the degree of ideal membership proofs from uniform families of polynomials over a finite field. *Illinois J. of Mathematics*, 45(1):41–73, 2001.

[Kra02] J. Krajíček. Dual weak pigeonhole principle, pseudo-surjective functions, and provability of circuit lower bounds. Manuscript, 2002.

[KP95] J. Krajíček and P. Pudlák. Some consequences of cryptographic conjectures for S_2^1 and EF. In *Logic and Computational Complexity*, D. Leivant, ed., Berlin, Springer-Verlag, pp. 210–220, 1995.

[KST97] S. Khanna, M. Sudan, and L. Trevisan. Constraint satisfaction: The approximability of minimization problems. In *Twelfth Annual Conference on Computational Complexity*, IEEE Computer Society, pp. 282–296, 1997.

[LPS88] A. Lubotsky, R. Philips, and P. Sarnak. Ramanujan graphs. *Combinatorica*, 8:261–277, 1988.

[LY94] C. Lund and M. Yannakakis. On the hardness of approximating minimization problems. *Journal of the Association for Computing Machinery*, 41, pp. 960–981, 1994.

[Mar88] G. A. Margulis. Explicit group-theoretical constructions of combinatorial schemes and their application to the design of expanders and superconcentrators. *Problemy Peredachi Informatsii (in Russian)*, 24:51–60, 1988. English translation in *Problems of Information Transmission, Vol. 24, pages 39-46.*

[N91] N. Nisan. Pseudo-random bits for constant-depth circuits. *Combinatorica*, 11(1):63-70, 1991.

[NW94] N. Nisan and A. Wigderson. Hardness vs. randomness. *Journal of Computer and System Sciences*, 49(2):149–167, October 1994.

[Pud01] P. Pudlák. On Reducibility and Symmetry of Disjoint NP-Pairs. In *Proc. 26th International Symposium MFCS*, 621-632, 2001.

[RanRaz02] R. Raz. Resolution lower bounds for the weak pigeonhole principle. In *Proceedings of the 34th ACM Symposium on the Theory of Computing*, pages 553–562, 2002.

[Raz95a] A. Razborov. Bounded Arithmetic and lower bounds in Boolean complexity. In P. Clote and J. Remmel, editors, *Feasible Mathematics II. Progress in Computer Science and Applied Logic, vol.* 13, pages 344–386. Birkhaüser, 1995.

[Raz95b] A. Razborov. Unprovability of lower bounds on the circuit size in certain fragments of bounded arithmetic. *Izvestiya of the RAN*, 59(1):201–224, 1995.

[Raz96] A. Razborov. Lower bounds for propositional proofs and independence results in Bounded Arithmetic. In F. Meyer auf der Heide and B. Monien, editors, *Proceedings of the 23rd ICALP, Lecture Notes in Computer Science,* 1099, pages 48–62, New York/Berlin, 1996. Springer-Verlag.

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

[Raz02a] A. Razborov. Resolution lower bounds for perfect matching principles. In *Proceedings of the 17th IEEE Conference on Computational Complexity*, pages 29–38, 2002.

[Raz02b] A. Razborov. Pseudorandom generators hard for k-DNF resolution and polynomial calculus resolution. Manuscript available at http://www.genesis.mi.ras.ru/~razborov, 2002.

[Rec76] R. A. Reckhow. On the Lengths of Proofs in the Propositional Calculus. PhD thesis, Department of Computer Science, University of Toronto, 1976.

[RR97] A. Razborov and S. Rudich. Natural proofs. *Journal of Computer and System Sciences*, 55(1):24–35, 1997.

[RS97] R. Raz, S. Safra. A sub-constant error-probability low-degree test, and a sub-constant error-probability PCP characterization of NP. In *Proceedings of the Twenty-Nineth Annual ACM Symposium on Theory of Computing*, Association for Computing Machinery, pp. 475–484, 1997.

[Sta77] R. Statman. Complexity of derivations from quantifier-free Horn formulae, mechanical introduction of explicit definitions, and refinement of completeness theorems. In *Logic Colloquium'76*, R. Gandy and M. Hyland, eds., Amsterdam, North-Holland, pp. 505–517, 1977.

[Tsa96] S.-C. Tsai. Lower bounds on representing Boolean functions as polynomials in $\mathbb{Z}_m$. *SIAM Journal on Discrete Mathematics*, 9:55–62, 1996.

[Tse68] Г. С. Цейтин. О сложности вывода в исчислении высказываний. In А. О. Слисенко, editor, *Исследования по конструктивной математике и математической логике*, II; *Записки научных семинаров ЛОМИ, т. 8*, pages 234–259. Наука, Ленинград, 1968. Engl. translation: G. C. Tseitin, On the complexity of derivations in propositional calculus, in: *Studies in mathematics and mathematical logic, Part II*, ed. A. O. Slissenko, pp. 115-125.

[Uma99] C. Umans. Hardness of Approximating Sigma_2 Minimization Problems. In *Proceedings of the 40th Annual IEEE Symposium on Foundations of Computer Science*: 465–474, 1999.

[Urq87] A. Urquhart. Hard examples for resolution. *Journal of the ACM*, 34(1):209–219, 1987.

[Yao82] A. Yao. Theory and applications of trapdoor functions. In *Proceedings of the 23rd IEEE FOCS*, pages 92–99, 1982.

[Yao85] A. Yao. Separating the polynomial-time hierarchy by oracles. In *Proceedings of the 26th IEEE FOCS*, pages 1–10, 1985.