On the Algebraic Proof Complexity of Constraint Satisfaction

Conneryd, Jonas

2026

On the Algebraic Proof Complexity of Constraint Satisfaction

by Jonas Conneryd

LUND
UNIVERSITY

Thesis for the degree of Doctor of Philosophy in Engineering

Thesis advisors: Prof. Tatyana Turova, Dr. Michael Doggett
Faculty opponent: Prof. Alexander Razborov

To be presented, with the permission of the Faculty of Engineering of Lund University, for public criticism in room E:1406 at the Department of Computer Science on September 25th, 2026 at 13:00.

Organization
LUND UNIVERSITY

Department of Computer Science
Box 118
SE–221 00 Lund
Sweden

Author(s)
Jonas Conneryd

Document name
DOCTORAL DISSERTATION

Date of disputation
2026-09-25

Sponsoring organization
Partially supported by the Swedish Research Council and the Wallenberg AI, Autonomous Systems and Software Program (WASP) funded by the Knut and Alice Wallenberg Foundation.

Title and subtitle
On the Algebraic Proof Complexity of Constraint Satisfaction

Abstract
This thesis comprises three papers in the field of *proof complexity*, in which the objects of study are certificates of unsatisfiability. In *algebraic* proof complexity, the aim is to prove lower bounds on the complexity of certifying that there is no common root of a given set of polynomials. We will be especially concerned with the *polynomial calculus* proof system, where this is accomplished by iteratively deriving new polynomials in the ideal generated by the input until reaching the constant polynomial 1.

In Paper A, we prove asymptotically optimal lower bounds on the size and degree required for polynomial calculus to refute the k-colorability of a sparse random graph sampled either from the Erdős–Rényi distribution or the uniform distribution over regular graphs.

In Paper B, we show that the so-called *Alekhnovich–Razborov* method for proving polynomial calculus degree lower bounds also yields level lower bounds for an algorithm called *cohomological k-consistency*, which is a general-purpose method for solving constraint satisfaction problems (CSPs). Together with the degree lower bounds from Paper A, which are established using this method, this result establishes optimal cohomological k-consistency level lower bounds for approximate graph coloring. Through this connection, we also provide an alternative proof of an optimal level lower bound for random instances of so-called *lax and null-constraining* CSPs, originally due to Chan and Ng.

Finally, in Paper C, we systematically investigate polynomial calculus over other variable domains than $\{0, 1\}$. Over other domains, the usual methods for proving size lower bounds break down. Using a new, unified framework, we prove optimal size lower bounds for random graph coloring in Bayer's formulation over roots of unity as well as for the functional pigeonhole principle over $\{\pm 1\}$-valued variables. In addition, we prove that polynomial calculus where each $\{0, 1\}$-valued variable also has a $\{\pm 1\}$-valued counterpart is *non-automatable*, which informally means that efficiently searching for proofs in this proof system is impossible unless $\mathsf{P} = \mathsf{NP}$. As a complement to our lower bounds for variable domains consisting of roots of unity, we prove that polynomial calculus over *non*-roots of unity simulates the *cutting planes* proof system with polynomially bounded coefficients.

Key words
computational complexity theory, proof complexity

Classification system and/or index terms (if any)

Supplementary bibliographical information

Language
English

ISSN and key title
1404-1219

ISBN
978-91-90202-69-2 (print)
978-91-90202-70-8 (pdf)

Recipient's notes

Number of pages
xii+243

Price

Security classification

Signature _____ Date 2026-08-31

On the Algebraic Proof Complexity of Constraint Satisfaction

Jonas Conneryd

Doctoral Thesis, 2026

Department of Computer Science
Lund University

ISBN 978-91-90202-70-8 (electronic version)
ISBN 978-91-90202-69-2 (print version)
ISSN 1404-1219
Doctoral Thesis 89, 2026
LU-CS-DISS: 2026-06

Department of Computer Science
Lund University
Box 118
SE-221 00 Lund
Sweden

E-mail: jonas.conneryd@cs.lth.se

Typeset using LaTeX
Printed in Sweden by Tryckeriet i E-huset, Lund, 2026

Abstract

This thesis comprises three papers in the field of *proof complexity*, in which the objects of study are certificates of unsatisfiability. In *algebraic* proof complexity, the aim is to prove lower bounds on the complexity of certifying that there is no common root of a given set of polynomials. We will be especially concerned with the *polynomial calculus* proof system, where this is accomplished by iteratively deriving new polynomials in the ideal generated by the input until reaching the constant polynomial 1.

In Paper A, we prove asymptotically optimal lower bounds on the size and degree required for polynomial calculus to refute the k-colorability of a sparse random graph sampled either from the Erdős-Rényi distribution or the uniform distribution over regular graphs.

In Paper B, we show that the so-called *Alekhnovich–Razborov* method for proving polynomial calculus degree lower bounds also yields level lower bounds for an algorithm called *cohomological k-consistency*, which is a general-purpose method for solving constraint satisfaction problems (CSPs). Together with the degree lower bounds from Paper A, which are established using this method, this result establishes optimal cohomological k-consistency level lower bounds for approximate graph coloring. Through this connection, we also provide an alternative proof of an optimal level lower bound for random instances of so-called *lax and null-constraining* CSPs, originally due to Chan and Ng.

Finally, in Paper C, we systematically investigate polynomial calculus over other variable domains than $\{0, 1\}$. Over other domains, the usual methods for proving size lower bounds break down. Using a new, unified framework, we prove optimal size lower bounds for random graph coloring in Bayer's formulation over roots of unity as well as for the functional pigeonhole principle over $\{\pm 1\}$-valued variables. In addition, we prove that polynomial calculus where each $\{0, 1\}$-valued variable also has a $\{\pm 1\}$-valued counterpart is *non-automatable*, which informally means that efficiently searching for proofs in this proof system is impossible unless $\mathsf{P} = \mathsf{NP}$. As a complement to our lower bounds for variable domains consisting of roots of unity, we prove that polynomial calculus over *non*-roots of unity simulates the *cutting planes* proof system with polynomially bounded coefficients.

Acknowledgements

Ed è subito sera. It is hard to believe that this thesis is finally finished. There are many people to thank, without whom it certainly would not exist.

First and foremost, I want to thank Tatyana for taking me on during a difficult time, and for the extraordinary kindness and support she has subsequently offered; and Mike, for all his help during and after said time. I also want to thank Jakob and Susanna for introducing me to this wonderful area.

Second, I want to thank my family for all they have done for me, and Elsa for these years spent together and for many more to come. I love you, always.

Thank you Andy, Christophe, David, Duri, Gaia, Kilian, Noel, Rui, Shuo, Stephan, Wietze, and Yassine for all the discussions and good memories. Special thanks to Ioana for making me laugh, and for being kind to me. Thanks also to Esra, Faseeh, and Hashim for being such good officemates.

In Copenhagen I want to thank Rasmus for his advice, Mikkel and Anne for creating such a welcoming atmosphere, and everyone else for their friendship.

I am very grateful to my opponent and evaluation committee for agreeing to referee this thesis. Thank you for your time and effort. Thanks also to Nutan Limaye for agreeing to referee my licentiate thesis.

Outside of academia, thanks in no particular order to Aron, Isak, Wille, Elias, Emma, Niklas, and David. I am incredibly lucky to have friends like you.

In addition to learning a great deal, I have had the privilege to travel extensively during my Ph.D. studies, all because people have been kind enough to invite me all over the world to workshops and conferences. In this regard, I would like to thank Nutan Limaye for inviting me to the *Workshop on Algebra in Computation (WAC)* in Gothenburg in summer 2023; Robert Robere, for inviting me to spend a week with his group in Montréal in summer 2024, and Stefan Grosser and John Ahmed Dellas for their hospitality while there; Ján Pich, for inviting me to the proof complexity workshop in Oxford in fall 2024; Susanna de Rezende, for organizing and encouraging me to attend the *Complexity as a Kaleidoscope Research School* in spring 2025 at CIRM, Marseille, France, where the ideas behind the second paper in this thesis were worked out; Ioana Bercea, for inviting me to spend a month with the theory group at KTH in fall 2025; Robert Robere again, for hosting me on a visit in summer 2026; and most of all Jakob Nordström, for arranging for me to attend workshops in Dagstuhl in fall 2022 and Oberwolfach in spring 2024, and for the wonderful two months I got to spend at the Simons Institute for the Theory of Computing at UC Berkeley in spring 2023.

Thank you Idriss for sharing your thesis template with me, and for your help with related questions. Many thanks are also due to the administration and technical support at LTH Datavetenskap for their diligent and competent work.

Finally, I want to thank the Wallenberg AI, Autonomous Systems and Software Program (WASP): for offering many of the courses I have taken during my studies; for funding my stay in Berkeley and my second, longer visit to Montréal; and for awarding me a postdoctoral scholarship for the coming years.

Contribution Statement

The following papers are included in this thesis. The papers appear as in each respective reference apart from cosmetic edits to fit the thesis format and unify notation as well as fixing of minor typos and inaccuracies.

Paper A Jonas Conneryd, Susanna F. de Rezende, Jakob Nordström, Shuo Pang, and Kilian Risse. Graph Colouring Is Hard on Average for Polynomial Calculus and Nullstellensatz. 2025. arXiv:2503.17022. Full version of a paper with the same title that appeared in *Proceedings of the 64th Annual IEEE Symposium on Foundations of Computer Science (FOCS '23), pages 1–11, November 2023*.

Paper B Jonas Conneryd, Yassine Ghannane, and Shuo Pang. Lower Bounds for CSP Hierarchies Through Ideal Reduction. 2025. arXiv:2511.17272. Appeared in *Proceedings of the 37th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA '26), pages 6436–6463, January 2026*.

Paper C Jonas Conneryd, Yassine Ghannane, Jakob Nordström, Shuo Pang, Kilian Risse, and Dmitry Sokolov. On the Power of Polynomial Calculus over Non-Boolean Domains. Manuscript, July 2026.

The table below indicates the responsibilities Jonas Conneryd had in each paper:

Paper	*Concepts*	*Writing*
A	◑	◑
B	◑	●
C	◑	◑

The dark portion of the circle represents the amount of work and responsibilities assigned to Jonas Conneryd for each individual step:

- ◔ Jonas Conneryd was a minor contributor to the work.
- ◑ Jonas Conneryd was a contributor to the work.
- ◕ Jonas Conneryd led and did a majority of the work.
- ● Jonas Conneryd led and did almost all of the work.

Populärvetenskaplig Sammanfattning

För vissa beräkningsproblem finns det algoritmer som är så effektiva att problemet går att lösa så gott som omedelbart närhelst det behövs. För andra, till synes likartade problem känner vi inte till någon effektiv algoritm—de verkar behöva lösas med ren råstyrka. Vad beror detta på? Finns det en ännu oupptäckt algoritm, eller är dessa problem svåra till sin natur?

Inom ämnet *beräkningskomplexitet* undersöks dessa frågor ur en matematisk synvinkel. Värdet av effektiva algoritmer är uppenbart, men varför vill man även veta att vissa problem inte går att lösa effektivt? Ett av många svar på denna fråga möter vi dagligen inom modern kryptografi. I dagens kryptografiska protokoll behöver mottagaren lösa ett beräkningsproblem för att öppna ett krypterat meddelande. Dessa problem är lätta att lösa med en krypteringsnyckel men anses vara omöjliga att lösa effektivt utan nyckeln. Det är därför inte en underdrift att säga att hela dagens digitala infrastruktur för säker kommunikation, och därmed elektroniska banktransaktioner, datalagring i molnet, etc., möjliggörs av svåra beräkningsproblem.

Det vore önskvärt att med säkerhet fastslå att ingen algoritm kan lösa ett givet problem inom en rimlig tidsram, men så allmängiltiga resultat ligger dessvärre långt bortom vad som kan bevisas med nuvarande matematiska verktyg. I stället gör man ofta ett *komplexitetsantagande* som säger att dessa problem är svåra, motiverat av beprövad erfarenhet och teoretiska resultat som stöder denna slutsats. Komplexitetsantaganden är mycket vanliga inom beräkningskomplexitet i stort, eftersom de låter forskare förklara varför ingen har lyckats konstruera en effektiv algoritm för breda uppsättningar beräkningsproblem: ofta kan man visa att en sådan algoritm även skulle lösa det förmodat svåra problemet som komplexitetsantagandet bygger på. Denna metod kan däremot inte visa att ett beräkningsproblem faktiskt är svårt, bara att det är minst lika svårt som problemet i antagandet.

Denna avhandling handlar i stället om att *ovillkorligen* visa att beräkningsproblem är svåra för specifika slags algoritmer. Avhandlingen verkar i ett forskningsområde kallat *beviskomplexitet*, vilket klassificerar algoritmer enligt vilket slags logik de använder när de körs. Därmed kan man uttrycka alla algoritmer i en sådan klass i ett formellt bevissystem, och varje gång en sådan algoritm körs så genereras ett bevis för dess slutsats uttryckt i detta bevissystem. Följaktligen kan man bevisa att en algoritm är ineffektiv för ett beräkningsproblem genom att visa att algoritmen i vissa fall enbart kan nå sin slutsats genom långa, komplicerade bevis i sitt motsvarande bevissystem.

Det gemensamma temat i de tre artiklarna i denna avhandling är att genom beviskomplexitet fastslå att en bred uppsättning algoritmer inte kan lösa en mängd så kallade *constraint satisfaction problems*, som närmast översätts till (det något krystade) *villkorsuppfyllandeproblem* på svenska. Dessa problem härstammar från artificiell intelligens och hör till de mest studerade inom beräkningskomplexitet i stort. Villkorsuppfyllandeproblem handlar förenklat om att avgöra ifall en given uppsättning sinsemellan motstridiga villkor kan uppfyllas samtidigt.

Ett naturligt villkorsuppfyllandeproblem som figurerar i alla tre artiklarna är *graffärgning*. I detta problem är man given en graf, det vill säga en uppsättning noder med streck mellan vissa av noderna, och ett antal tillåtna färger. Uppgiften är att avgöra ifall varje nod kan tilldelas en färg utan att några noder med streck mellan sig får samma färg. Detta problem är vad som kallas NP-*fullständigt*, vilket innebär att en effektiv algoritm för graffärgning även ger en effektiv algoritm för alla problem i klassen NP. Denna komplexitetsklass innehåller en mängd problem av intresse både teoretiskt och i praktiken, exempelvis olika former av schemaläggning, handelsresandeproblemet, heltalsfaktorisering, etc.

Gemensamt för alla NP-problem är att en given lösning är lätt att verifiera—exempelvis kan man enkelt bekräfta att en färgtilldelning uppfyller alla villkor i graffärgningsproblemet om så är fallet. Det centrala olösta problemet inom beräkningskomplexitet är om detta innebär att sådana problem även är lätta att lösa: har alla problem i klassen NP en effektiv algoritm? Det kanske vanligaste komplexitetsantagandet i området är att så *inte* är fallet, vilket alltså är ekvivalent med att anta att graffärgning i synnerhet inte kan lösas effektivt. Bland annat bekräftar avhandlingen detta antagande ovillkorligen för algoritmer som kan uttryckas i bevissystemet *polynomkalkyl*, vilket exempelvis innefattar de som löser problemet genom att beräkna Gröbnerbaser. I själva verket visar avhandlingen att om man väljer en graf att färga från en sannolikhetsfördelning över rätt sorts grafer så tar detta problem med mycket hög sannolikhet exponentiellt lång tid att lösa. Därmed är en stor majoritet av graferna i fördelningen så gott som maximalt svåra att färga för sådana algoritmer, vilket är ett betydligt starkare resultat än att problemet är svårt i värsta fall.

Contents

Introduction

What distinguishes mathematics from other intellectual activities is the notion of *proof*. A mathematical claim is only accepted as true when supplemented by an airtight argument of its validity. Compelling evidence, appeal to authority, or conventional wisdom are all moot; only absolute truth will do. Of course, the only reason mathematicians can impose such high standards is that they restrict the kind of statements they study to those expressible in a formal reasoning system, where what is true—that is, the consequences of the *axioms* of the theory—is agreed upon beforehand. This means, of course, that the truth of a statement is completely determined by the axioms; no proof is needed. Since 1931, when Kurt Gödel published his landmark first *incompleteness theorem*, it has even been known that any strong enough logical theory—in particular, any theory that attempts to capture conventional mathematical reasoning—must contain sentences that are true but *unprovable* within the theory. Thus, informally speaking, proof and truth are two different things.

So why do we need proofs? Their importance is perhaps better explained through a sociological perspective, namely that mathematics is an activity performed by *humans*, and that the purpose of a proof is to convince *other humans* of a mathematical claim. As summarized in an adage often attributed to A. M. Gleason: *"Proofs aren't there to convince you that something is true, but to show you why it's true."*[1] This description is probably more congenial to practitioners of mathematics, all of whom have surely encountered both proofs that are turgid and cumbersome and others they feel are surprising, insightful, or even beautiful.

If a proof is to convince a human of a mathematical claim, it must adhere to much more stringent demands than simply being logically sound. In fact, given our physical limitations, some mathematicians have argued that all theorems we can ever hope to prove must be either essentially trivial or possess an "insightful" proof, much in the spirit of Gleason's saying.[2] In a contrived sense, this is obviously true for theorems concerning infinite mathematical objects, simply because a proof is, by definition, a *finite* string of symbols. Therefore, if it manages to convince us that, say, some property holds for all members of some infinite set, then the proof must have identified some commonality among all members of that set, and reasoned about them all at once. With some good will, it is fair to call such a proof insightful.

[1] See Thurston [Thu94] for a lucid discussion of this and related matters.

[2] Wigderson [Wig19, §3.4] puts forward an accessible and, in this author's opinion, compelling argument in favor of this hypothesis.

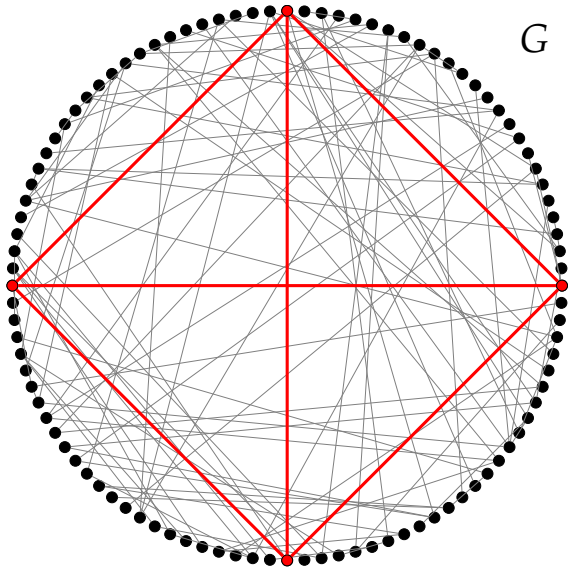

Figure 1.1: *A* 100*-vertex graph that contains a* 4*-clique, marked in red.*

What about what we might call *simple truths*, which concern finite objects and which, in principle, can be checked by exhaustive case analysis of these objects? Why would such theorems need insightful proofs? Let us see an example. Consider the graph G in Figure 1.1. It is a theorem, albeit not a very interesting one, that there is no way to label each vertex of G with one of three colors so that the endpoints of each edge in G receive different colors. In other words, G is not 3-colorable. To convince another person of this fact, one can simply tell them to try all 3^{100} possible assignments of colors to the vertices of G, and that upon doing so they will find that none work. This task takes much longer than the universe has existed, even with the help of a computer, and the argument is therefore unlikely to convince anyone. A better proof is to note that G contains a so-called 4-*clique* as a subgraph, which even by itself cannot be 3-colored.

While none of these two proofs are especially profound, the second is clearly preferable. The reason is *scaling*. The number of reasoning steps in the first proof grows exponentially with the size of the graph we are considering, while the second confines our reasoning to an object much smaller than G—a 4-clique. How specific was this example? Does there always exist an insightful proof—one that is significantly shorter than performing exhaustive case analysis—that a graph is not 3-colorable when this is the case? Seemingly more generally: *Do all simple truths possess insightful proofs?* These are the kinds of questions studied in the field of *proof complexity*, to which this thesis belongs.

Most researchers believe that the answer to all questions above is no; that is, just as certain complicated truths are unprovable, some simple, eminently provable theorems are true for no good reason. Somewhat ironically, this itself

seems to be an exceedingly complicated truth, one we are very far from proving. Suitably formalized, this question is equivalent to whether the complexity classes NP and coNP are equal; separating them would in particular separate P from NP and thus resolve one of the deepest, most well-known open problems in all of mathematics and computer science.

While we currently cannot hope to answer these questions in full generality, the *Cook–Reckhow program* in proof complexity has outlined a (long and arduous) path toward them. The idea is to start with very constrained proofs, and show that no short proofs of some simple truth exist in this particular, constrained form. Then, gradually lift some of the restrictions, see whether the techniques developed for the constrained case can be adapted to this new, more challenging setting, and rinse and repeat. Papers A and C in this thesis fit into this program: they establish lower bounds on the sizes of proofs of a particular form for a collection of simple truths for which such lower bounds were not known before. In fact, some of these simple truths are precisely those in our example—that certain graphs are not 3-colorable.

Paper B instead focuses on the interplay between algorithms and proof complexity. Informally speaking, many heuristic algorithms for 3-colorability and other so-called *constraint satisfaction problems (CSPs)* proceed by searching for evidence that the graph is not 3-colorable, and accept the instance if it cannot find any. Such *certificates of unsatisfiability* generally take two forms: either they are small, local obstructions such as the 4-clique in Figure 1.1, or they show that even some relaxed version of the problem, which has algebraic structure that allows for efficient algorithms, has no solution.

Since proof complexity aims to show that proofs of certain facts must be complex, it is not so surprising that techniques from this area can be useful for establishing limitations for these algorithms by showing that the certificates they search for must be infeasibly complicated in the worst case. Paper B uses techniques from proof complexity to prove lower bounds on the running time of a CSP algorithm called *cohomological k-consistency*, which searches for a combination of local and algebraic certificates of unsatisfiability.

Background

This chapter provides the background needed to understand the original contributions in this thesis. We begin with preliminaries from algebra, graph theory, and propositional logic, followed by an introduction to computational complexity theory and the subfield of proof complexity in particular. With the foundations in place, we conclude this chapter with an overview of the concrete proof systems and propositional formulas relevant to this thesis.

2.1 Preliminaries

This section contains the necessary preliminaries for understanding the original contributions of this thesis, along with some conventions and notation that we use.

2.1.1 Notation

For a natural number n, we write $[n]$ to denote the set $\{1, 2, \ldots, n\}$. We write $\log(\cdot)$ to denote the base 2 logarithm and $\ln(\cdot)$ for the natural logarithm. We use standard asymptotic notation: for functions $f, g \colon \mathbb{N} \to \mathbb{N}$, we write $f = O(g)$ if there exist nonnegative constants c and n_0 such that $f(n) \leq c \cdot g(n)$ for all $n \geq n_0$, and write $f = \Omega(g)$ if $g = O(f)$. If $f = O(g)$ and $g = O(f)$, we write $f = \Theta(g)$. We denote the set of binary strings of finite length by $\{0,1\}^*$; a subset of $\{0,1\}^*$ is called a *language*. We denote the length of a string $x \in \{0,1\}^*$ by $|x|$. For a set S, we denote the collection of subsets of S by 2^S, the collection of subsets of S of size k by $\binom{S}{k}$, and the collection of subsets of size at most k by $\binom{S}{\leq k}$.

2.1.2 Algebra

This section introduces the necessary material from algebra, most of which is standard. The perhaps slightly less familiar notion of reduction modulo ideals is covered at greater length than here by Mikša and Nordström [MN24, Section 2.3], still with a view toward proof complexity.

For a field $\mathbb{F}$, we denote the polynomial ring over $\mathbb{F}$ in n variables by $\mathbb{F}[x_1, \ldots, x_n]$. The set of variables appearing in a polynomial p is denoted $\mathit{Vars}(p)$. A *monomial* is a product of variables, and a *term* is a monomial multiplied by a field element. For a set of polynomials $\mathcal{P} = \{p_1, \ldots, p_m\}$ in $\mathbb{F}[x_1, \ldots, x_n]$, the set of polynomials of the form $\sum_{i \in [m]} q_i p_i$ for $q_i \in \mathbb{F}[x_1, \ldots, x_n]$ is called the *ideal generated by* $\mathcal{P}$ and is denoted by $\langle \mathcal{P} \rangle$. For a polynomial p and a partial mapping

$\rho\colon Vars(p) \to \mathbb{F}$, we denote the polynomial obtained by evaluating p according to ρ by $p\restriction_\rho$.

A total well-order $\prec$ on the monomials in $\mathbb{F}[x_1, \ldots, x_n]$ is called *admissible* if (1) for any monomial m it holds that $1 \prec m$ and (2) for any monomials m_1, m_2 and m such that $m_1 \prec m_2$, it holds that $m \cdot m_1 \prec m \cdot m_2$.[1] Fixing an admissible order $\prec$, the *leading term* of a polynomial p is the term in p with the largest monomial with respect to $\prec$. A polynomial p is *reducible modulo* an ideal I if there exists $q \in I$ with the same leading term as p; otherwise, p is *irreducible* modulo I.

It is straightforward to show that for any polynomial p and any ideal I, there exists a unique representation $p = q + r$ where $q \in I$ and r is a linear combination of irreducible monomials modulo I. We call r the *reduction* of p modulo I. The operator R_I on $\mathbb{F}[x_1, \ldots, x_n]$ that takes a polynomial p to its reduction modulo I is called the *reduction operator modulo* I. It is not hard to show that the reduction operator is $\mathbb{F}$-linear.

We need to establish one property of reduction operators, which we use in Section 3.1.3. The reader can safely skip it until then.

Fact 2.1.1. *For all ideals I and J in $\mathbb{F}[x_1, \ldots, x_n]$ such that $I \subseteq J$, and all monomials m and m', it holds that $R_J(m'R_I(m)) = R_J(m'm)$.*

Proof. Write $m = q + r$ where $q \in I$ and r is the reduction of m modulo I. Then $m'R_I(m) = m'r$, and also $m'R_I(m) = q' + r'$ where $q' \in J$ and r' is the reduction of $m'R_I(m)$ modulo J. Observe that $R_J(m'R_I(m)) = r'$, so the claim follows if we show that $R_J(m'm) = r'$. Note that $m'R_I(m) = m'(m - q) = q' + r'$, and hence $m'm = q' + r' - m'q$. Since $I \subseteq J$ it holds that $m'q \in J$. Therefore, we can write $m'm = q' - m'q + r'$ where $q' - m'q \in J$ and r' is irreducible modulo J. By uniqueness of this representation, it follows that $R_J(m'm) = r'$. □

2.1.3 Graph Theory

Unless otherwise stated, all graphs in this thesis are finite, simple, and undirected. A graph $G = (V, E)$ is *d-regular* if all its vertices have degree d. The *neighborhood* of a set $U \subseteq V$ is the set $N(U) = \{v \in V \setminus U \mid \exists u \in U : (u, v) \in E\}$. We write $E(U)$ to denote the set of edges in E whose endpoints are both in U, and $E(U, W)$ to denote the set of edges with one endpoint in U and one endpoint in $W \setminus U$. An *independent set* in G is a set $U \subseteq V$ such that no two vertices in U share an edge. The *subgraph of G induced by U* is $G[U] = (U, E(U))$. A *path of length $\ell > 0$* in G is a sequence $(v_1, \ldots, v_{\ell+1})$ of distinct vertices such that $(v_i, v_{i+1}) \in E$ for all $i \in [\ell]$. We follow the convention that there is a path of length 0 from a vertex to itself. A *cycle* is a path, but where the first and last vertices coincide. A graph is *connected* if there is a path between every pair of vertices in it. A *forest* is a graph without cycles, and a *tree* is a connected forest. The *girth* of a graph is the length of its shortest cycle, or ∞ if there are no cycles in it.

[1] This definition differs slightly (but insignificantly) from that used in [MN24].

A *proper k-coloring* of a graph $G = (V, E)$ is a map $\chi : V \to [k]$ such that for every edge $(u, v) \in E$, it holds that $\chi(u) \neq \chi(v)$. If there exists a proper k-coloring of G, we say that G is *k-colorable*. The *chromatic number* of G is the smallest k for which a proper k-coloring of G exists, and is denoted $\chi(G)$.

A graph G with edge set E is *bipartite* if its vertex set can be partitioned into two subsets U, V such that $E = E(U, V)$. We then write $G = (U \sqcup V, E)$ to emphasize this partition. Note that G is bipartite if and only if it is 2-colorable.

We consider two models of random graphs: the *Erdős-Rényi random graph model* $\mathbb{G}(n, p)$, which is the distribution over graphs on n vertices where each edge is included independently with probability p; and the *random d-regular graph model* $\mathbb{G}_{n,d}$, which is the uniform distribution over d-regular graphs on n vertices. A graph property P holds *asymptotically almost surely* for a random graph model $\mathbb{G} = \{\mathbb{G}_n : n \in \mathbb{N}\}$ if $\lim_{n\to\infty} \Pr_{G\sim\mathbb{G}_n}[G \text{ has property } P] = 1$.[2]

A graph $G = (V, E)$ is (s, ε)-*sparse* if every set $U \subseteq V$ such that $|U| \leq s$ satisfies $|E(U)| \leq (1+\varepsilon)|U|$, and G is an (s, a)-*expander* if every set $U \subseteq V$ such that $|U| \leq s$ satisfies $|E(U, V \setminus U)| \geq a|U|$. Note that a d-regular graph is (s, ε)-sparse if and only if it is an (s, a)-expander for $a = d - 2(1 + \varepsilon)$.

The condition on vertex sets in the definition of expander graphs in this section is technically that of *edge expansion*. Many variants of expander graphs have been defined and used in the literature. For instance, some applications require that the neighborhoods of small sets are large, not only that they contain many edges, and some require good *boundary expansion*, meaning that the neighborhood of each small set $U \subseteq V$ contains many vertices connected to exactly one vertex in U. Moreover, some definitions are phrased in terms of the eigenvalues of the adjacency matrix, and others are specialized to bipartite graphs. For an overview, see [HLW06]. In informal discussions, we will be somewhat lax about these differences and refer to all of them collectively as *expander graphs*.

2.1.4 Propositional Logic

We briefly review some standard notions from propositional logic. A *Boolean variable* x takes either the value `True` or `False`, which we identify with 1 and 0, respectively. A *literal* ℓ over a variable x is either x itself or its *negation* $\neg x$ (denoted also by $\overline{x}$), which is `True` when x is `False` and vice versa. A *(propositional) formula* is a well-formed logical expression involving literals of Boolean variables and the binary connectives OR($\vee$) and AND($\wedge$). We use $\bot$ and $\top$ to denote the constant false and true formulas, respectively. A *clause* $C = \ell_1 \vee \ldots \vee \ell_k$ is a disjunction of literals over distinct variables. A formula is in *conjunctive normal form (CNF)* if it is a conjunction of clauses. It is a standard fact that any propositional formula over a set of variables $x_1, \ldots, x_n$ is logically equivalent to (that is, has the same truth table as) a formula in CNF over the same variable set. For a formula F, we denote the set of variables appearing in it by $\mathit{Vars}(F)$.

[2] This definition is a bit problematic for random regular graphs, since d-regular graphs on n vertices can only exist if dn is even. In cases such as this, the limit is understood to be taken only over those n for which the distribution $\mathbb{G}_n$ is over a non-empty set.

A *(partial) assignment* to the variables of a formula F is a (partial) mapping $\rho\colon \mathit{Vars}(F) \to \{0,1\}$. An assignment that is not partial is called *total*. If there exists an assignment ρ such that F evaluates to `True` when the variables are assigned according to ρ, then F is *satisfiable*; otherwise, F is *unsatisfiable*. A formula F is a *tautology* if every assignment to its variables satisfies F.

2.1.5 Relational Structures and Homomorphisms

A *signature* σ is a finite set of symbols $\Gamma_1, \ldots, \Gamma_\ell$, each with respective arity $\mathrm{ar}(\Gamma_i) \in \mathbb{N}$. A *$\sigma$-structure* $\mathbf{A}$ consists of a domain A and, for each $\Gamma \in \sigma$, a relation $\Gamma^{\mathbf{A}} \subseteq A^{\mathrm{ar}(\Gamma)}$. A *relational structure* is a σ-structure for some σ. A relational structure $\mathbf{A}$ is *finite* if $|A|$ is finite. All relational structures in this thesis are finite unless otherwise stated.

For a σ-structure $\mathbf{A}$ and a set $X \subseteq A$, the *substructure of* $\mathbf{A}$ *on* X, denoted $\mathbf{A}[X]$, is the σ-structure with domain X where, for $\Gamma \in \sigma$, the relation $\Gamma^{\mathbf{A}[X]}$ consists of the tuples in $\Gamma^{\mathbf{A}}$ that mention only elements in X.

Given two σ-structures $\mathbf{A}$ and $\mathbf{B}$, a *homomorphism* from $\mathbf{A}$ to $\mathbf{B}$ is a map $\varphi\colon A \to B$ such that for each symbol $\Gamma \in \sigma$ and each tuple $\mathbf{a} = (a_1, \ldots, a_{\mathrm{ar}(\Gamma)}) \in \Gamma^{\mathbf{A}}$, it holds that $\varphi(\mathbf{a}) = (\varphi(a_1), \ldots, \varphi(a_{\mathrm{ar}(\Gamma)})) \in \Gamma^{\mathbf{B}}$. If there exists a homomorphism from $\mathbf{A}$ to $\mathbf{B}$, we say that $\mathbf{A} \to \mathbf{B}$; if not, we write $\mathbf{A} \not\to \mathbf{B}$. The restriction of a map $\varphi\colon A \to T$ to a set $X \subseteq A$ is denoted $\varphi{\restriction_X}$. Given sets $Y \subseteq X \subseteq A$ and a homomorphism $\varphi\colon \mathbf{A}[Y] \to \mathbf{T}$, if there exists a homomorphism $\psi\colon \mathbf{A}[X] \to \mathbf{T}$ such that $\psi{\restriction_Y} = \varphi$, we say that ψ *extends* φ (to X). We denote the set of homomorphisms from $\mathbf{A}$ to $\mathbf{B}$ by $\mathrm{Hom}(\mathbf{A}, \mathbf{B})$.

2.2 Computational Complexity Theory

In Chapter 1, we stated that this thesis studies (the lack of) *insightful* proofs of tautologies. A technically more accurate (albeit less evocative) term, which we will use going forward, is *efficient* proofs. The proper setting in which to study these questions is that of *computational complexity theory*, which we now introduce. A more comprehensive treatment of the material in this section can be found in any textbook on the subject; see, for instance, Arora and Barak [AB09, Chapters 1-2].

In a nutshell, the goal of computational complexity theory is to understand the capabilities and limits of *efficient computation*. The term *computation* here intuitively refers to the application of a set of mechanical rules, together with some form of *memory* for storing intermediate results for future use. An *algorithm* for computing a function $f\colon \{0,1\}^* \to \{0,1\}^*$ consists of a set of such rules as well as unambiguous instructions for how to apply them.[3] The *running time* of an algorithm is the total number of applications of these rules.

[3] The set of binary strings is simply a convenient and inessential choice of representation of a set of arbitrary finite objects, such as graphs, propositional formulas or natural numbers. What object a string encodes will be clear from context.

A mathematical formalization of computation as described above is provided by the *Turing Machine* [Tur36], which we will not describe formally here. The reason is that the notion of computation as captured by Turing Machines is extremely robust—whenever the word *algorithm* appears in this thesis, the reader can think of it as implemented in the computational model they find most intuitive, such as a program in their favorite programming language (albeit with no constraints on memory usage), and be sure that the algorithm can also be implemented by a Turing machine with overhead small enough to be insignificant for our considerations; see [AB09, Chapter 1] for a more detailed discussion.[4]

2.2.1 Efficient Computation and the Class NP

Next, we define what constitutes *efficient* computation (with respect to running time). We focus on *decision problems*, which for a language $L \subseteq \{0,1\}^*$ and a string $x \in \{0,1\}^*$ decide whether $x \in L$, or equivalently, compute the function $f_L \colon \{0,1\}^* \to \{0,1\}$ such that $f_L(x) = 1$ if and only if $x \in L$.

Definition 2.2.1 (Polynomial-time computability, the class P)**.** A language $L \subseteq \{0,1\}^*$ is *polynomial-time computable* if there exists an algorithm A deciding L and a constant $c \in \mathbb{N}$ such that for every string $x \in \{0,1\}^*$, the algorithm A runs in time $O(|x|^c)$. The class of polynomial-time computable languages is denoted P.

Much of computational complexity theory aims to understand the class P, and in particular which languages are *not* in P. The most prominent such example is the class NP, which captures those languages where membership can be efficiently *verified*.

Definition 2.2.2 (The class NP)**.** A language $L \subseteq \{0,1\}^*$ is in NP if there exists a polynomial-time algorithm $A_L \colon \{0,1\}^* \times \{0,1\}^* \to \{0,1\}$ and a polynomial $p \colon \mathbb{N} \to \mathbb{N}$ such that for every string $x \in L$ there exists $\pi_x \in \{0,1\}^*$ such that $|\pi_x| \leq p(|x|)$ and $A_L(\pi_x, x) = 1$, and such that $A_L(\pi, x) = 0$ for all $\pi \in \{0,1\}^*$ whenever $x \notin L$.

By definition, a language L is in NP precisely when its elements have *efficient* proofs of membership in L in the form of the strings $\{\pi_x : x \in L\}$, which can then also be checked efficiently (that is, in time polynomial in $|x|$) by the algorithm A_L.

Observe that P $\subseteq$ NP, since if a language L can be decided by a polynomial-time algorithm A_L, then A_L can be given an additional string π as input and simply disregard it. Determining whether also NP $\subseteq$ P is the P vs. NP question, the central open problem in computational complexity theory and one of the Clay Mathematics Institute's seven Millennium Prize Problems [CJW06]. We cannot hope to adequately cover the depth and breadth of the P vs. NP question and its implications here, and instead refer to Aaronson's in-depth survey [Aar16].

The language of interest to us in proof complexity is TAUT, which consists of all propositional tautologies. Asking whether every tautology has an efficient

[4] This is the venerable *Extended (or Complexity-Theoretic) Church–Turing Thesis.*

proof therefore becomes mathematically well-defined in the following form: *Is* TAUT *in* NP*?* We will soon see that this question is intimately related to whether P = NP, but first we need a few more definitions.

A language L is *polynomial-time (Karp) reducible* to a language L' if there exists a polynomial-time computable function $f : \{0,1\}^* \to \{0,1\}^*$ such that for every string $x \in \{0,1\}^*$ it holds that $x \in L$ if and only if $f(x) \in L'$. A language L is NP*-hard* if every language in NP is polynomial-time reducible to L, and NP*-complete* if also $L \in$ NP.

The NP-complete problems are the hardest problems in NP, in the sense that a polynomial-time algorithm deciding an NP-complete language L can be turned into a polynomial-time algorithm deciding any other language in NP through reductions. In two landmark papers, Cook [Coo71] and independently Levin [Lev73] found the first natural example of an NP-complete problem.

Theorem 2.2.3 (Cook–Levin)**.** *The language* SAT *of satisfiable Boolean formulas is* NP*-complete.*

Building on their work, Karp [Kar72] proved that 21 natural combinatorial problems are NP-complete. Following Cook, Levin, and Karp, thousands of computational problems coming from all over science, mathematics, operations research, and more have been proven NP-complete; a classic reference containing hundreds of examples is the book of Garey and Johnson [GJ79]. This is one reason that the P vs. NP question is important: if P ≠ NP, we cannot hope for a polynomial-time algorithm for *any* of these problems. Determining the complexity of the language TAUT only requires Cook's original result that SAT is NP-complete, however, and we do so next.

2.2.2 The Class coNP

The complexity class that characterizes the complexity of TAUT is called coNP.

Definition 2.2.4 (Complement, the class coNP)**.** The *complement* of a language $L \subseteq \{0,1\}^*$ is $\{0,1\}^* \setminus L$, which we denote by $\overline{L}$. The class coNP consists of all languages whose complement is in NP.

Note that for any language $L \subseteq$ coNP and any string $x \in \overline{L}$, there exists an efficient proof that $x \notin L$ since $\overline{L} \in$ NP. It is clear that P ⊆ coNP, for essentially the same reason that P ⊆ NP.

Polynomial-time reductions and completeness can be defined for coNP analogously as for NP. Using Theorem 2.2.3, it is not hard to show that TAUT is in fact coNP-complete.

Theorem 2.2.5. *The language* TAUT *is* coNP*-complete.*

Proof. First, we show that TAUT ∈ coNP. Its complement $\overline{\text{TAUT}}$ is the set of Boolean formulas with at least one falsifying assignment. Therefore, membership in $\overline{\text{TAUT}}$ can be decided by a polynomial-time algorithm that takes as input a

formula F and an assignment π to the variables of F and accepts if and only if $F(\pi) = 0$. We conclude that $\overline{\text{TAUT}} \in \text{NP}$, and consequently $\text{TAUT} \in \text{coNP}$.

To see that TAUT is coNP-hard, recall from Theorem 2.2.3 that the language SAT is NP-complete. Therefore, for any language $L \in \text{coNP}$ there exists a polynomial-time computable function which on an input $x \in \{0,1\}^*$ produces a formula ϕ_x that is satisfiable if and only if $x \in \overline{L}$. It follows that $\neg\phi_x \in \text{TAUT}$ if and only if $x \in L$. □

We saw in Section 2.2.1 that our original question of whether all tautologies possess efficient proofs can be formalized as asking whether TAUT is in NP. Due to Theorem 2.2.5, this question is in turn equivalent to whether NP = coNP or not. Most researchers believe that NP ≠ coNP, but proving this seems hopelessly out of reach. Establishing that NP ≠ coNP would in particular imply that P ≠ NP: if a language is in P then so is its complement, so if P = NP then NP = coNP.

The observation that TAUT is coNP-complete and the consequences for the NP vs. coNP and P vs. NP questions were among the original motivations for the field of proof complexity, which is covered in Section 2.3.

2.3 Proof Complexity

This section introduces the main concepts in the field of *proof complexity*, to which this thesis belongs. We only cover the bare minimum and refer the reader to Krajíček [Kra19] for a much more in-depth treatment of this rich area.

2.3.1 The Cook–Reckhow Program

The field of proof complexity proper began with the seminal work of Cook and Reckhow [CR79], some of which appeared earlier in a preliminary version [CR74] and in Reckhow's Ph.D. thesis [Rec75]. The work of Cook and Reckhow explicitly links proving lower bounds on the sizes of proofs of tautologies to central questions in computational complexity theory, using the observation in Section 2.2.2 that NP ≠ coNP if and only if TAUT ∉ NP.

A key contribution of Cook and Reckhow was a way to classify different types of proofs of a given tautology through the notion of a *propositional proof system*, also known as a *Cook–Reckhow proof system*.

Definition 2.3.1 (Propositional proof system, P-proof)**.** A *propositional proof system* (or *proof system*) is a polynomial-time computable function $P\colon \{0,1\}^* \to \{0,1\}^*$ such that $\text{Rng}(P) = \text{TAUT}$. A string $x \in \{0,1\}^*$ such that $P(x) = F$ is called a *P-proof* of F.

Any Cook–Reckhow proof system P is sound, since it only proves tautologies, and is complete since the range of P is all of TAUT. The key difference to classical proof systems is that a proof in a Cook–Reckhow proof system must be able to be efficiently verified (that is, in time polynomial in the proof size), enforced by the function P being polynomial-time computable.

To formalize the notion of a proof system that can efficiently prove every tautology, Cook and Reckhow define *polynomially bounded* proof systems.

Definition 2.3.2 (Polynomially bounded)**.** A Cook–Reckhow proof system P is *polynomially bounded* if there exists a polynomial $p\colon \mathbb{N} \to \mathbb{N}$ such that for all strings $x \in \text{TAUT}$, there exists $\pi \in \{0,1\}^*$ such that $P(\pi) = x$ and $|\pi| \leq p(|x|)$.

Using that TAUT is coNP-complete, it is not hard to prove that the existence of a polynomially bounded proof system is *equivalent* to NP = coNP.

Theorem 2.3.3. *There exists a polynomially bounded proof system if and only if* NP = coNP*.*

Proof. From Theorem 2.2.5 we know that NP = coNP if and only if TAUT $\in$ NP. If a proof system P is polynomially bounded, then the function $A_P\colon \{0,1\}^* \times \{0,1\}^* \to \{0,1\}$ mapping (π, x) to 1 if and only if π is a P-proof of x witnesses that TAUT $\in$ NP.

Conversely, if TAUT $\in$ NP then there exists a polynomial $p\colon \mathbb{N} \to \mathbb{N}$ and a polynomial-time algorithm $A\colon \{0,1\}^* \times \{0,1\}^* \to \{0,1\}$ such that for all $x \in \text{TAUT}$ there exists $\pi \in \{0,1\}^*$ such that $|\pi| \leq p(|x|)$ and $A(\pi, x) = 1$. Define $P_A\colon \{0,1\}^* \to \{0,1\}^*$ by $P_A(y) = x$ if y encodes the running trace of A on some pair (π, x) such that $A(\pi, x) = 1$, and $P_A(y) = \top$ otherwise. Since A runs in time polynomial in the size of (π, x) and for every $x \in \text{TAUT}$ there exists π_x of size polynomial in x such that $A(\pi, x) = 1$, the function P_A is a polynomially bounded proof system. □

Theorem 2.3.3 formalizes our question of whether all tautologies possess an efficient proof as asking whether there exists a polynomially bounded proof system. Cook and Reckhow also suggest a long-term approach to this question through the notions of *simulation* and *p-simulation*, which we now define.[5]

Definition 2.3.4 (Simulation, p-simulation)**.** A proof system P *simulates* a proof system Q if there exists a function $f\colon \{0,1\}^* \to \{0,1\}^*$ and a polynomial $q\colon \mathbb{N} \to \mathbb{N}$ such that for all $x \in \{0,1\}^*$, it holds that $|f(x)| \leq q(|x|)$ and $P(f(x)) = Q(x)$. If, in addition, the function f is polynomial-time computable, then P is said to *p-simulate* Q.

If a proof system P simulates a proof system Q, then all tautologies with proofs in Q of polynomial size also have polynomial-size proofs in P, and if P p-simulates Q, then a given Q-proof of a tautology F can be converted to a P-proof of F in polynomial time. Simulations and p-simulations thus allow us to speak of stronger and weaker proof systems. Together with Theorem 2.3.3, these notions suggest an ambitious research program for proving that NP $\neq$ coNP, and hence that P $\neq$ NP: *For stronger and stronger proof systems, prove that some family of tautologies does not admit polynomial-sized proofs in these proof systems. Generalize*

[5]The definition of p-simulation also appears in earlier work of Cook [Coo75].

these results to eventually prove that for any *given proof system, there exists some family of tautologies with no polynomial-size proofs in that proof system.*[6]

As the field has developed, its scope has broadened and connections to other parts of logic and theoretical computer science have been discovered. In particular, the template above is much too narrow to encompass all kinds of questions studied in modern proof complexity. For further reading, we refer to the books of Krajíček [Kra95; Kra19] and Cook and Nguyen [CN10] for extensive treatments of the logical side of proof complexity, and to the survey articles of Buss and Nordström [BN21] and Fleming, Kothari, and Pitassi [FKP19] for accounts of the interplay between proof complexity and the design and analysis of algorithms.

2.3.2 Proofs, Refutations, and Algebraic Proof Systems

To conclude this section, we note that the theory developed in Section 2.3.1 works equally well for coNP-complete languages other than TAUT, and indeed other languages are often more convenient. For instance, it often turns out to be more natural in practice to consider *refutations* of formulas in the language UNSAT of *unsatisfiable* propositional formulas, which is easily seen to be coNP-complete since it consists of negations of formulas in TAUT.

We will also consider *algebraic proof systems*, which reason about sets of polynomials with no common root instead of unsatisfiable Boolean formulas.[7] For an algebraic proof system to reason about a CNF formula F in the variables $x_1, \ldots, x_n$, we translate it to a set of polynomials $\mathcal{P}_F$ as follows: for $i \in [n]$, each variable x_i is mapped to the polynomial $1 - x_i$, and each negation $\overline{x}_i$ is mapped to x_i. Each clause $C \in F$ is mapped to the product of the images of the literals in it. Finally, we add the *Boolean axioms* $\{x_i^2 - x_i : i \in [n]\}$ to $\mathcal{P}_F$ to ensure that its variables take Boolean values.

2.4 Proof Systems

This section introduces the propositional proof systems relevant to the original contributions of this thesis. Besides being of intrinsic interest, all three proof systems in this section have myriad applications all over computer science, perhaps especially in the design and analysis of algorithms. We cannot go into

[6] Note, however, that no such family $\mathcal{F} = \{\mathcal{F}_n : n \in \mathbb{N}\}$ can have a "simple description," i.e., be constructible in time polynomial in n. The reason is that we can then modify a given system P to compute $\{\mathcal{F}_n\}$ when given as input a formula of size n and then check whether the input is in $\mathcal{F}_n$. We could still hope to prove, e.g., that for a given proof system P it holds with nonzero probability that a randomly sampled k-CNF formula requires P-proofs of superpolynomial size. See [San21] for a discussion of this and related matters.

[7] For this problem to be in coNP, we require all variable domains to be fixed to finite sets, most often $\{0, 1\}$, which is achieved by strengthening the input with univariate polynomials whose roots are precisely each respective variable domain. Technically, we also need that all coefficients in the polynomials in the set have a reasonable encoding as a string; this is true for all sets considered in this thesis, and in proof complexity at large. Hence, we do not worry about this technical point going forward.

further detail here but refer the interested reader to the survey article of Buss and Nordström [BN21], where the algorithmic applications of all proof systems in this section are discussed.

2.4.1 Resolution

Without question, the most well-known and studied proof system in proof complexity is *resolution*. Predating the field of proof complexity itself, resolution was introduced by Blake [Bla37] in his Ph.D. thesis; other important early works on (variants of) resolution include those of Davis and Putnam [DP60], Davis, Logemann, and Loveland [DLL62], and Robinson [Rob65]. Given an unsatisfiable CNF formula F, a *resolution refutation* of F is a sequence of clauses $\pi = (D_1, \ldots, D_\tau)$ such that D_τ is the empty clause $\bot$ and each clause D_i in π is either a clause of F or a clause of the form $D_i = B \vee C$ derived from clauses $D_j = B \vee x$ and $D_k = C \vee \overline{x}$, where $D_j, D_k \in \pi$ and $j, k < i$, using the *resolution rule*

$$\frac{B \vee x \qquad C \vee \overline{x}}{B \vee C}. \tag{2.1}$$

The clause $B \vee C$ is called the *resolvent* of $B \vee x$ and $C \vee \overline{x}$ over the variable x. It is easy to see that resolution is a sound proof system, since the resolution rule is sound. That resolution is (refutationally) complete requires a short argument, which we record as Proposition 2.4.1.

Proposition 2.4.1. *Each unsatisfiable CNF formula F over n variables $x_1, \ldots, x_n$ has a resolution refutation with at most $2^{n+1} - 1$ clauses.*

Proof sketch. Create a binary decision tree of depth n where for $i \in [n]$ the internal vertices at depth i are labelled by x_i, so that each root-to-leaf path corresponds to a total assignment to $x_1, \ldots, x_n$. Starting from the root, as soon as a partial assignment on a path violates a clause in F, terminate that path and label the new leaf with a clause of F violated by the assignment. Since F is unsatisfiable, every root-to-leaf path in this new tree receives a label. Starting at maximum depth, resolve the leaf clauses on the variable their common parent is labelled by, label their parent by the resolvent, and proceed inductively to the root. □

The complexity measures of a resolution refutation most relevant to this thesis are *size* (sometimes also called *length*) and *width*. The size of a resolution refutation π is the number of clauses in it. The width of a clause is the number of variables in it, and the width of π is the maximum width among the clauses appearing in π. A celebrated result of Ben-Sasson and Wigderson [BW01] provides a lower bound on resolution size in terms of resolution width.

Theorem 2.4.2 ([BW01], Corollary 3.6)**.** *Let F be an unsatisfiable CNF formula over n variables, and suppose that all clauses in F have width at most w_0. If resolution requires width w to refute F, then resolution requires size $\exp(\Omega((w - w_0)^2/n))$ to refute F.*

In the other direction, a resolution width upper bound of w for refuting a formula F implies a size upper bound of $n^{O(w)}$ for refuting F simply because there are $\sum_{i=0}^{w} \binom{2n}{i} \leq n^{O(w)}$ distinct clauses of width at most w. Theorem 2.4.2 is a first example of an important paradigm in proof complexity, namely that a lower bound for an auxiliary measure such as width can often be translated to a size lower bound. Some of the original contributions of this thesis use similar reasoning; see Section 3.1.3 and Section 3.3.3.

2.4.2 Nullstellensatz

As a first example of algebraic proof systems, we now introduce *Nullstellensatz*. As its name suggests, this proof system is based on the fundamental *Hilbert's Nullstellensatz*, which predates the field of proof complexity by many decades.[8] The Nullstellensatz proof system was introduced by Beame et al. [BIKP+94], but similar ideas appear earlier in, for instance, a paper of Lovász [Lov82].

Let $\mathbb{F}$ be a field and $\mathcal{P} = \{p_1, \ldots, p_m\}$ a set of polynomials over $\mathbb{F}$ in the variables $x_1, \ldots, x_n$. If the polynomials in $\mathcal{P}$ have no common root, a *Nullstellensatz certificate* of this fact is a syntactic polynomial identity of the form

$$\sum_{i=1}^{m} p_i q_i = 1, \tag{2.2}$$

where $q_1, \ldots, q_m$ are arbitrary polynomials in $\mathbb{F}[x_1, \ldots, x_n]$.

Nullstellensatz is a sound proof system: no certificate as in (2.2) can exist if the polynomials in $\mathcal{P}$ have a common root, since evaluating both sides of (2.2) on that common root would result in the contradictory equality $0 = 1$. The completeness of Nullstellensatz, meaning that an identity as in (2.2) always exists if the polynomials in $\mathcal{P}$ have no common root, is precisely the statement of the weak version of Hilbert's Nullstellensatz, provided that $\mathbb{F}$ is algebraically closed. In proof complexity, the domain of each of the variables $x_1, \ldots, x_n$ is always finite. In this case, Hilbert's Nullstellensatz holds even if $\mathbb{F}$ is not algebraically closed (for instance, when $\mathbb{F}$ is a finite field) and hence the Nullstellensatz proof system is complete over every field. This theorem has surely been folklore for many decades, but a version of it was explicitly stated in an influential paper by Alon [Alo99][9], who also used it to derive various combinatorial results. We include a short proof of this result below with no claim of originality.

Theorem 2.4.3 (Finite-Domain Nullstellensatz)**.** *Let g be a polynomial and $\mathcal{Q} \subseteq \mathbb{F}[x_1, \ldots, x_n]$, and let $S \subseteq \mathbb{F}$ be a finite set. Finally, for $i \in [n]$, let r_i denote the polynomial $\prod_{\alpha \in S}(x_i - \alpha)$. If $\mathcal{Q}$ contains the set of polynomials $\{r_i : i \in [n]\}$, then g vanishes on all common roots of $\mathcal{Q}$ if and only if $g \in \langle \mathcal{Q} \rangle$. In particular, $1 \in \langle \mathcal{Q} \rangle$ if and only if there is no common root of the polynomials in $\mathcal{Q}$.*

[8] Hilbert proved the Nullstellensatz in 1893 [Hil93].

[9] Note, however, that the paper [BIKP+94] appeared before [Alo99].

Proof. If $g \in \langle Q \rangle$, then $g = \sum_j f_j q_j$ for $f_j \in \mathbb{F}[x_1, \ldots, x_n]$ and $q_j \in Q$ and hence vanishes on all common roots of the polynomials in Q. For the other direction, let $\beta = (\beta_1, \ldots, \beta_n) \in S^n$ and let $R \subseteq S^n$ be the set of common roots of the polynomials in Q. Denote $|S| = k$, define the *coordinate degree* of a polynomial to be the maximum degree among the variables in its terms and consider the polynomial

$$p_\beta = \prod_{i \in [n]} \prod_{\substack{\alpha \in S \\ \alpha \neq \beta_i}} (x_i - \alpha) \tag{2.3}$$

which as a function $S^n \to \mathbb{F}$ is nonzero if and only if $x = \beta$. Note that p_β is a linear combination of monomials of coordinate degree at most $k - 1$. Write $\mathbf{1}_\beta \coloneqq p_\beta / p_\beta(\beta)$ to denote the polynomial p_β normalized to output 1 on input β.

Each function $f \colon S^n \to \mathbb{F}$ can be expressed as a polynomial of coordinate degree at most $k - 1$ through the identity

$$f = \sum_{\beta \in S^n} f(\beta) \cdot \mathbf{1}_\beta \,. \tag{2.4}$$

Moreover, the vector space V of polynomials in $\mathbb{F}[x_1, \ldots, x_n]$ of coordinate degree at most $k - 1$ has dimension k^n, and the k^n polynomials $\{\mathbf{1}_\beta : \beta \in S^n\}$ are linearly independent over V since all linear combinations $\sum_j \alpha_j \cdot \mathbf{1}_{(\beta)_j}$ evaluate to 0 on all β outside $\{(\beta)_j\}$. Therefore, the polynomials $\{\mathbf{1}_\beta : \beta \in S^n\}$ form a basis of V and hence the representation in (2.4) is unique.

Modulo the polynomials $\{r_i : i \in [n]\}$, each polynomial $p \in \mathbb{F}[x_1, \ldots, x_n]$ can be written as a polynomial $\widetilde{p}$ with coordinate degree at most $k - 1$ that agrees with p on S^n. In particular, since g vanishes on R we can use (2.4) to write

$$\widetilde{g} = \sum_{\beta \in S^n \setminus R} g(\beta) \cdot \mathbf{1}_\beta \,. \tag{2.5}$$

If $R = S^n$, it follows immediately that $g \in \langle Q \rangle$ since $\{r_i : i \in [n]\} \subseteq Q$ by assumption. If not, it suffices to show that for each element $\beta \in S^n \setminus R$ it holds that $\mathbf{1}_\beta \in \langle Q \rangle$ to conclude the proof.

Since R is the set of common roots of Q, it holds for every element $\beta \in S^n \setminus R$ that there exists a polynomial $q \in Q$ such that $q(\beta) \neq 0$. Note that the polynomial $(q(\beta))^{-1} q \cdot \mathbf{1}_\beta \in \langle Q \rangle$ coincides with $\mathbf{1}_\beta$ on all of S^n, and hence $q^{-1}(\beta) q \cdot \mathbf{1}_\beta = \mathbf{1}_\beta$ modulo the ideal $\langle \{r_i : i \in [n]\} \rangle$. Since $\{r_i : i \in [n]\} \subseteq Q$, it follows that $\mathbf{1}_\beta \in \langle Q \rangle$. □

For simplicity, Theorem 2.4.3 assumes that the domain of all variables is the same. It is clear that a similar argument works when the variables have different (finite) domains, but requires more cumbersome notation.

This thesis considers two complexity measures of a Nullstellensatz certificate as in (2.2). The first and most commonly studied is *degree*, which is the maximum degree among the polynomials $p_i q_i$ appearing in it. The second is *size*, which is the quantity $\sum_{i \in [m]} |\{\text{ monomials in } q_i\}| \cdot |\{\text{monomials in } p_i\}|$. Other size measures

are also used in the literature, but all the usual definitions are polynomially equivalent; see [BN21, Section 7.5.1] for a discussion.

For Nullstellensatz to be able to reason about a CNF formula F, it must first be translated to a set $\mathcal{P}$ of polynomials (see Section 2.3.2). The usual translation maps each variable x to the polynomial $(1 - x)$. Therefore, a very wide clause can turn into a polynomial of exponential monomial size. To avoid this technical problem, a set of *dual variables* $\overline{x}_1, \ldots, \overline{x}_n$ along with polynomials $\{x_i + \overline{x}_i - 1 : i \in [n]\}$ is often added to sets $\mathcal{P}$ coming from a CNF formula.

2.4.3 Polynomial Calculus

The Nullstellensatz proof system introduced in Section 2.4.2 differs from resolution in that a proof consists of a single equality rather than a sequence of derivations. The *polynomial calculus* proof system defined in this section can be seen as a common generalization of the two, in which proofs are sequences of derivations like in resolution but the underlying reasoning is algebraic. Polynomial calculus was originally introduced by Clegg, Edmonds, and Impagliazzo [CEI96].[10]

Just like Nullstellensatz, polynomial calculus proves that there are no common roots of the polynomials in a set $\mathcal{P} = \{p_1, \ldots, p_m\} \subseteq \mathbb{F}[x_1, \ldots, x_n]$ through showing that the constant polynomial 1 is in the ideal generated by $\mathcal{P}$. Polynomial calculus has two derivation rules:

$$\textit{Linear combination: } \frac{p \quad q}{\alpha p + \beta q},\ \alpha, \beta \in \mathbb{F}; \tag{2.6a}$$

$$\textit{Multiplication: } \frac{p}{x_i p},\ x_i \text{ any variable.} \tag{2.6b}$$

As the reader may have noted already, these are precisely the operations under which an ideal is closed. A *polynomial calculus derivation* of a polynomial q starting from $\mathcal{P}$ is a sequence of polynomials $(p_1, \ldots p_\tau)$ such that $p_\tau = q$ and each polynomial p_i is either contained in $\mathcal{P}$ or obtained via linear combination or multiplication applied to polynomials p_j such that $j < i$. A *polynomial calculus refutation* of $\mathcal{P}$ is a derivation of the constant polynomial 1 from $\mathcal{P}$. By the same arguments as for Nullstellensatz, polynomial calculus is sound, and complete over every field when the domains of all variables are finite.

This thesis concerns two complexity measures of polynomial calculus refutations: *size* and *degree*. They can be thought of as algebraic analogs of the resolution size and width measures in Section 2.4.1. The *size* of a polynomial p is its number of terms when expanded into a linear combination of distinct monomials, and the *degree* of p is the maximal degree among all its monomials. The size of a

[10] They called it the *Gröbner proof system* due to its connections with the Gröbner basis algorithm. Polynomial calculus seems to have received its current name from Buss et al. [BIKP+97].

polynomial calculus refutation π is the sum of the sizes of the polynomials in it, and the degree of π is the maximal degree among its polynomials.[11]

Polynomial calculus p-simulates Nullstellensatz with respect to both size and degree: from a Nullstellensatz refutation $1 = \sum_{i=1}^{m} q_i p_i$ for a set $\mathcal{P} = \{p_i\}_{i=1}^{m}$, we can obtain the same refutation in polynomial calculus by first deriving each polynomial $q_i p_i$ and then adding the polynomials together. However, as defined above polynomial calculus does not p-simulate, or even simulate, resolution with respect to size [dRLNS21, §1.2]. Therefore, most papers about polynomial calculus in fact study a stronger proof system called *polynomial calculus resolution (PCR)* [ABRW02, §4.1], which is polynomial calculus where additionally each variable x_i has a formal negation $\overline{x}_i$, enforced by adding the polynomials $x_i + \overline{x}_i - 1$ to $\mathcal{P}$.

Polynomial calculus degree and PCR degree are equivalent, since the substitution $\overline{x}_i \mapsto 1 - x$ turns a PCR refutation into a polynomial calculus refutation of the same degree. This substitution does not preserve size, however, and [dRLNS21, Theorem 1] shows that a size blowup is sometimes unavoidable. PCR p-simulates resolution with regard to both size and degree (width), since a resolution refutation can be simulated in PCR step-by-step. By using dual variables, clauses can be translated to monomials, and resolving two clauses $C_1 \vee x_i$ and $C_2 \vee \overline{x}_i$ on a variable x_i can be simulated by adding $m_{C_1 \vee x}$ and $m_{C_2 \vee \overline{x}}$ and then subtracting the polynomial $m_{C_1 \vee C_2} \cdot (x + \overline{x} - 1)$, which is derivable through repeated multiplication in size $O(w(C_1 \vee C_2))$ and degree $w(C_1 \vee C_2) + 1$.

Just as for resolution size and width, a strong enough lower bound on polynomial calculus degree for a set $\mathcal{P}$ over Boolean variables implies a lower bound on polynomial calculus size. Notably, this result was established before the analogous size-width relation for resolution of Ben-Sasson and Wigderson [BW01].

Theorem 2.4.4 ([IPS99], Corollary 6.3)**.** *Let $\mathcal{P}$ be an unsatisfiable set of polynomials of degree D_0 over n Boolean variables. If polynomial calculus requires degree D to refute $\mathcal{P}$, then PCR requires size* $\exp(\Omega((D - D_0)^2/n))$ *to refute $\mathcal{P}$.*

The statement and proof of Theorem 2.4.4 in [IPS99] are for polynomial calculus, but the same proof works for PCR.

2.5 Formulas in Proof Complexity

The discussion so far establishes that the central goal of proof complexity is to prove lower bounds on the size required for a proof system to refute a given unsatisfiable formula (or set of polynomials with no common root). The goal of this section is to introduce some of the most well-studied formulas in proof complexity, as well as the formulas relevant for the original contributions of this

[11] It might seem more natural to define the size of a polynomial calculus refutation to be its *length*, that is, the number of polynomials in it. This definition turns out to be problematic: if the polynomials are represented modulo the Boolean axioms, as is common in proof complexity, then *every* unsatisfiable set $\mathcal{P}$ of polynomials turns out to have a polynomial calculus refutation of length linear in the number of variables (but of exponential monomial size); see [MN24, Proposition 2.4] for details.

thesis. Of course, the formulas mentioned in this section constitute only a very small fraction of those that have been studied in proof complexity. We refer the reader to [Kra19, Part III] and references therein for a more extensive treatment.

2.5.1 The Pigeonhole Principle

The most studied formula in proof complexity, introduced already by Cook and Reckhow [CR79], is the (negation of the) *pigeonhole principle* stating that there is no way to fit $m \geq n+1$ pigeons into n holes without some hole containing at least two pigeons. Expressed as a formula in CNF, the pigeonhole principle PHP_n^m takes the form

$$\bigvee_{j=1}^{n} x_{i,j} \qquad i \in [m] \tag{2.7a}$$

$$\overline{x}_{i,j} \vee \overline{x}_{i',j} \qquad i \neq i' \in [m],\ j \in [n] \tag{2.7b}$$

where (2.7a) states that each pigeon gets at least one hole and (2.7b) ensures that no two pigeons get the same hole. The pigeonhole principle can be strengthened by the additional clauses

$$\overline{x}_{i,j} \vee \overline{x}_{i,j'} \qquad i \in [m],\ j \neq j' \in [n] \tag{2.8a}$$

$$\bigvee_{i=1}^{m} x_{i,j} \qquad j \in [n] \tag{2.8b}$$

where (2.8a) is called the set of *functionality axioms* and (2.8b) the *onto axioms*. The pigeonhole principle augmented by (2.8a) is called the *functional pigeonhole principle* $FPHP_n^m$, and states that there exists an injective function $[m] \to [n]$. Adding (2.8b) in turn results in the *onto pigeonhole principle*, denoted by $ontoPHP_n^m$, and adding both (2.8a) and (2.8b) to PHP_n^m results in the *onto functional pigeonhole principle*, claiming that there exists a bijective function $[m] \to [n]$. Since adding additional axioms can only make a formula easier to refute, a lower bound for refuting a more constrained principle implies the same lower bound for less constrained versions.

Another way to constrain the pigeonhole principle is to limit which holes each pigeon is allowed to fly to by defining the formula on a bipartite graph $G = (U \sqcup V, E)$, with the left side representing the pigeons and the right side representing the holes. The graph pigeonhole principle $PHP(G)$ and its functional and onto variants are then defined as in (2.9), but only including adjacent pairs (i, j) in G. Note that for any graph G, the pigeonhole principle on G can be obtained from the ordinary version by restricting variables. Hence, for any proof system closed under variable restrictions, and in particular those considered in this thesis, a lower bound for the graph version implies the same lower bound for the full pigeonhole principle.

For Nullstellensatz and polynomial calculus to reason about the pigeonhole principle, it must be translated to a set of polynomials. The translation in Section 2.3.2 turns PHP_n^m into a set of polynomials of degree n (and exponential monomial size), which is undesirable for proof systems where degree is the primary complexity measure. Therefore, the translation

$$\sum_{j=1}^{n} x_{i,j} - 1 \qquad i \in [m] \tag{2.9a}$$

$$x_{i,j}x_{i',j} \qquad i \neq i' \in [m],\ j \in [n] \tag{2.9b}$$

$$x_{i,j}^2 - x_{i,j} \qquad i \in [m],\ j \in [n] \tag{2.9c}$$

is more commonly used. The functionality and onto axioms then take the form

$$x_{i,j}x_{i,j'} \qquad i \in [m],\ j \neq j' \in [n] \tag{2.10a}$$

$$\sum_{i=1}^{m} x_{i,j} - 1 \qquad j \in [n]\,. \tag{2.10b}$$

It is relevant for this thesis to consider the pigeonhole principle over variable bases other than $\{0,1\}$. If the variables instead take the values $\{\pm 1\}$, then the translation analogous to (2.9a) can run into issues if the field characteristic divides $m-1$. Hence, in this case one mostly considers the translation of the CNF graph pigeonhole principle over low-degree graphs, which then takes the form

$$\prod_{j \in N(i)} (x_{i,j} + 1) \qquad i \in U \tag{2.11a}$$

$$(x_{i,j} - 1) \cdot (x_{i',j} - 1) \qquad i \neq i' \in U,\ j \in N(i) \cap N(i') \tag{2.11b}$$

$$x_{i,j}^2 - 1 \qquad i \in U,\ j \in N(V) \tag{2.11c}$$

with the same semantics as the $\{0,1\}$-valued version.

The pigeonhole principle is an obvious mathematical fact and is provable by basic counting. The formula PHP_n^m is therefore often thought of as gauging the ability of a proof system to count, since if a proof system requires exponential size to refute PHP_n^m it is probably equally poor at performing more complicated counting arguments.

Over Boolean variables, the proof complexity of the pigeonhole principle and its variants is reasonably well-understood for all proof systems introduced so far. Haken [Hak85] proved that resolution requires size $\exp(\Omega(n))$ to refute PHP_n^{n+1}, and his proof can be extended to work also when including the functionality and onto axioms. Later work by Raz [Raz04a] and Razborov [Raz03; Raz04b] showed that resolution requires size $\exp(\Omega(n^\varepsilon))$ to refute all variants of the pigeonhole principle, for arbitrary m and for ε that does not depend on m. This was extended to the graph version by de Rezende, Nordström, Risse, and Sokolov [dRNRS25].

For polynomial calculus, and hence also Nullstellensatz, Razborov [Raz98] proved that $FPHP_n^m$ requires degree $\Omega(n)$ to refute over any field and for all

values of m. Alekhnovich and Razborov [AR03] proved $\exp(\Omega(n))$ size lower bounds for $FPHP_n^m$ where $m = O(n)$, but for a different encoding. Mikša and Nordström [MN24] obtained the same degree and size lower bounds for the usual encoding of $FPHP_n^m$.

For the onto and onto functional version, the situation is different. Over fields of characteristic either 0 or p that does not divide $m - n$, it is easy to see that Nullstellensatz can refute $ontoPHP_n^m$ in degree 1: simply sum the axioms (2.9a) and (2.10b) over m and n, respectively, and then subtract the results from each other. Moreover, Riis [Rii93] proved that Nullstellensatz over fields of characteristic p can refute $ontoFPHP_n^m$ in degree $D + 1$ whenever $\binom{m}{D} \neq \binom{n}{D} \mod p$.[12] Therefore, over fields of characteristic p, Nullstellensatz can refute $ontoFPHP_n^{n+p^\ell}$ in degree $p^\ell + 1$. Interestingly, Beame and Riis [BR98, Theorem 8.1] later proved a matching degree lower bound of 2^ℓ for refuting $ontoFPHP_n^{n+p^\ell}$ in Nullstellensatz. It is currently an open problem to extend their result to polynomial calculus [Raz26, Open Problem 21].

Over $\{\pm 1\}$-valued variables, degree lower bounds follow from the Boolean case by simple variable substitutions, while Sokolov [Sok20] proved an $\exp(\Omega(|V(G)|))$ polynomial calculus size lower bound for $PHP(G)$ on expander graphs. Paper C in this thesis establishes the same exponential lower bound for the functional version.

2.5.2 Tseitin Formulas

The *Tseitin formula* is defined on a graph $G = (V, E)$ over Boolean variables $\{x_e : e \in E\}$. Each vertex $v \in V$ is given a *charge* $b_v \in \{0, 1\}$ such that $\sum_{v \in V} b_v$ is odd. The Tseitin formula, which we denote by $Tse(G)$, contains (a suitable encoding of) the following axioms:

$$\sum_{e \ni v} x_e = b_v \mod 2 \qquad v \in V\,. \tag{2.12}$$

The formula $Tse(G)$ also has a natural encoding as a set of polynomials where the variables and charges instead take values in the set $\{-1, 1\}$. The axioms are then

$$\prod_{e \ni v} x_e - b_v \qquad v \in V \tag{2.13a}$$

$$x_e^2 - 1 \qquad e \in E \tag{2.13b}$$

where (2.13a) corresponds to (2.12) and (2.13b) ensures that the variables are $\{-1, 1\}$-valued.

[12] This result also appears as Lemma 8.7 in [BR98] (which is easier to access than [Rii93]) in a slightly more specialized form than here. The proof of the special case in [BR98] is easy to extend to the general case.

The Tseitin formulas are almost as well-studied as the pigeonhole principle. Tseitin [Tse68][13] introduced these formulas and proved that resolution requires size $\exp(\Omega(n))$ to refute them on the $n \times n$ grid[14] provided the refutation is *regular*, which is a technical condition we do not describe further here. Galil [Gal77] built on Tseitin's result to show a (strongly exponential) $\exp(\Omega(n))$ size lower bound for refuting Tseitin formulas on n-vertex expander graphs of constant degree. Urquhart [Urq87] in turn extended Galil's result by removing the regularity assumption, thus proving the first strongly exponential size lower bounds for resolution.

The Tseitin formulas can be refuted by Nullstellensatz in $O(n)$ size and constant degree over fields of characteristic 2 (where the constraints become affine functions), and were shown by Grigoriev [Gri98] to require $\Omega(n)$ degree to refute over other fields when defined on expander graphs. This lower bound was later extended to polynomial calculus by Buss, Grigoriev, Impagliazzo, and Pitassi [BGIP01] and further simplified by Beame and Impagliazzo [BI10].

2.5.3 Graph Coloring

Many of the original contributions of this thesis pertain to the proof complexity of *graph k-colorability*, which is one of the original 21 problems proved NP-complete in the seminal work of Karp [Kar72]. It therefore differs from the pigeonhole principle and Tseitin formulas, since these formulas express basic reasoning primitives while graph coloring is thought to be a genuinely difficult computational problem.

For a graph $G = (V, E)$ with chromatic number strictly larger than k, we define an unsatisfiable CNF formula $Col(G, k)$ stating that G has a proper k-coloring as follows:

$$\bigvee_{i=1}^{k} x_{v,i} \qquad v \in V \tag{2.14a}$$

$$\overline{x}_{v,i} \vee \overline{x}_{v,j} \qquad v \in V,\ i \neq j \in [k] \tag{2.14b}$$

$$\overline{x}_{u,i} \vee \overline{x}_{v,i} \qquad (u, v) \in E,\ i \in [k]\,. \tag{2.14c}$$

Equations (2.14a) and (2.14b) together state that every vertex $v \in V$ receives exactly one color, and (2.14c) ensures that the coloring is proper.

To study the proof complexity of k-coloring in the algebraic proof systems Nullstellensatz and polynomial calculus in Sections 2.4.2 and 2.4.3, we must translate $Col(G, k)$ to a set of polynomials similarly as for the pigeonhole principle:

[13] This paper precedes that of Cook and Reckhow [CR79] by more than a decade!

[14] Note that n is here the square root of the total number of variables.

$$\sum_{i=1}^{k} x_{v,i} - 1 \qquad v \in V \tag{2.15a}$$

$$x_{v,i}x_{v,i'} \qquad v \in V,\ i \neq i' \in [k] \tag{2.15b}$$

$$x_{u,i}x_{v,i} \qquad (u,v) \in E,\ i \in [k] \tag{2.15c}$$

$$x_{v,i}^2 - x_{v,i} \qquad v \in V,\ i \in [k]\,. \tag{2.15d}$$

The semantics of (2.15a)–(2.15c) are the same as in the CNF encoding of $Col(G,k)$, while (2.15d) ensures that the variables take Boolean values. Note that the set (2.15a)–(2.15d) is not the straightforward translation of the CNF encoding in (2.14a)–(2.14c), but these sets of polynomials are efficiently derivable from each other.

Both polynomial calculus and Nullstellensatz can reason about more general systems of polynomials than those coming from translations of CNF formulas. Therefore, we can also consider the proof complexity of encodings of k-colorability where the variables are not Boolean. The most well-known such encoding is seemingly due to Bayer [Bay82, Chapter III, §3] and appears in (2.16a)–(2.16b).

$$x_v^k - 1 \qquad v \in V \tag{2.16a}$$

$$\sum_{j=0}^{k-1} x_u^j x_v^{k-1-j} \qquad (u,v) \in E \tag{2.16b}$$

Equation (2.16a) ensures that each variable x_v is assigned a k-th root of unity in the underlying field (which is then of characteristic that either does not divide k or is 0), each of which corresponds to a color. Equation (2.16b) in turn asks that adjacent vertices in G are assigned different colors. This is easiest seen by assigning the variables x_u and x_v in (2.16b) to roots of unity ξ^i and ξ^ℓ, respectively; the result is then

$$\sum_{j=0}^{k-1} \xi^{ij+\ell(k-1-j)} = \xi^{-\ell} \sum_{j=0}^{k-1} \xi^{(i-\ell)j} = \begin{cases} \xi^{-\ell}k, & i = \ell \\ 0, & \text{otherwise.} \end{cases} \tag{2.17}$$

Equations (2.16a)–(2.16b) are usually referred to as the *Fourier encoding* or *roots-of-unity encoding* of k-colorability, and we denote it by $Col_\xi(G,k)$.

2.5.4 (Promise) Constraint Satisfaction Problems and Proof Complexity

The k-colorability problem is a special case of a more general class of so-called *constraint satisfaction problems (CSPs)*. An *ℓ-CSP* is defined over a set $x_1,\dots,x_n$ of variables, each taking values in respective domains $D_1,\dots,D_n$, together with a set $C_1,\dots,C_m$ of *constraints*. Each constraint C_i specifies a tuple $(x_{i_1},\dots,x_{i_\ell})$ of distinct variables as well as a set $S_i \subseteq \bigtimes_{j=1}^{\ell} D_{i_j}$ of allowed simultaneous

assignments to the variables $x_{i_1}, \ldots, x_{i_\ell}$. The task is to decide whether all constraints can be simultaneously satisfied by some assignment to the variables. Expressed in this language, the k-colorability problem on a graph G is a CSP where the domains of the variables are all $[k]$, each constraint corresponds to an edge of G, and the allowed assignments are those where the endpoints of the edge receive different colors. An instance of SAT on a formula in CNF is also a CSP, where the constraints are the clauses in the formula.

It has long been known [Jea98; FV98] that every constraint satisfaction problem can be formulated as the task of determining, given two relational structures $\mathbf{A}$ and $\mathbf{T}$, whether there is a homomorphism $\mathbf{A} \to \mathbf{T}$. Probably the most common variant is when the structure $\mathbf{T}$ is kept fixed, in which case we say that $\mathbf{T}$ is the *template* of the CSP. More formally, given a σ-structure $\mathbf{T}$, an *instance of the constraint satisfaction problem with template* $\mathbf{T}$ is a σ-structure $\mathbf{A}$ such that, for all $\Gamma \in \sigma$, there are no tuples in $\Gamma^{\mathbf{A}}$ with repeated elements.[15] We define CSP($\mathbf{T}$) as the computational problem where the input is an instance $\mathbf{A}$ and the task is to determine whether $\mathbf{A} \to \mathbf{T}$. For instance, the template of graph k-colorability is the complete graph on k vertices. In an *ℓ-CSP*, all relations in the template have the same arity ℓ.

We will also discuss so-called *promise constraint satisfaction problems (PCSPs)*, which generalize CSPs. For a pair $(\mathbf{S}, \mathbf{T})$ of fixed σ-structures such that $\mathbf{S} \to \mathbf{T}$, the *promise constraint satisfaction problem with template* $(\mathbf{S}, \mathbf{T})$, denoted by PCSP($\mathbf{S}, \mathbf{T}$), is the computational problem of deciding whether $\mathbf{A} \to \mathbf{S}$ or $\mathbf{A} \not\to \mathbf{T}$ for a given instance $\mathbf{A}$, promised that one of the two holds. Note that PCSP($\mathbf{T}, \mathbf{T}$) = CSP($\mathbf{T}$). Perhaps the most natural PCSP is *(c, ℓ)-approximate graph coloring* [GJ76], in which the task is to determine whether an input graph G is either c-colorable or not even ℓ-colorable, for $\ell > c$. The template of this PCSP consists of a pair of complete graphs on c and ℓ vertices, respectively.

The definition above is the *decision* version of PCSP. In the *search* version, the input is an instance $\mathbf{A}$ such that $\mathbf{A} \to \mathbf{S}$, and the task is to find a homomorphism from $\mathbf{A}$ to $\mathbf{T}$. For fixed-template CSPs it is known that the search version reduces to the decision version [BJK05], and hence that the two are equally hard. However, no generic search-to-decision reduction is known for PCSPs, and recent work [Lar25] shows some partial evidence against the existence of such a reduction.

Similarly to the encoding (2.15a)–(2.15c) of graph k-colorability, given an instance $\mathbf{A}$ of CSP($\mathbf{T}$) there is a canonical encoding of the statement that $\mathbf{A} \to \mathbf{T}$ as the set of polynomials $\mathcal{P}_{\mathbf{A}\to\mathbf{T}}$ in (2.18a)–(2.18d) over Boolean variables $x_{a,i}$. The common roots of $\mathcal{P}_{\mathbf{A}\to\mathbf{T}}$ correspond precisely to the homomorphisms $\mathbf{A} \to \mathbf{T}$

[15] This corresponds to having no repeated variables in the constraint variable tuples as defined above.

(encoded as Boolean indicator vectors).

$$\sum_{i \in T} x_{a,i} - 1 \qquad a \in A \qquad \begin{bmatrix}\text{every vertex } a \in A \\ \text{mapped to some } i \in T\end{bmatrix} \tag{2.18a}$$

$$x_{a,i}x_{a,i'} \qquad a \in A,\ i \neq i' \in T \qquad \begin{bmatrix}\text{no } a \in A \text{ mapped to} > 1 \\ \text{vertex } i \in T\end{bmatrix} \tag{2.18b}$$

$$\prod_{a \in \mathbf{a}} x_{a,f(a)} \qquad \begin{matrix}\Gamma \in \sigma,\ \mathbf{a} \in \Gamma^{\mathbf{A}}, \\ f\colon \mathbf{a} \mapsto f(\mathbf{a}) \in T^{\mathrm{ar}(\Gamma)} \setminus \Gamma^{\mathbf{T}}\end{matrix} \qquad \begin{bmatrix}\text{no tuple in } \Gamma^{\mathbf{A}} \text{ mapped to} \\ \text{tuple not in } \Gamma^{\mathbf{T}}\end{bmatrix} \tag{2.18c}$$

$$x_{a,i}^2 - x_{a,i} \qquad a \in A,\ i \in T \qquad \text{[Boolean axioms]} \tag{2.18d}$$

Note that for graph coloring, the monomials in (2.18c) are precisely the degree-2 monomials $x_{u,i}x_{v,i}$ that encode that two adjacent vertices u, v in the input graph both receive color i.

Both k-colorability and CSPs in general have a rich history in proof complexity, especially so-called *random CSPs* where the constraints are picked at random from a distribution. We will discuss known proof complexity lower bounds for k-colorability and other constraint satisfaction problems in Section 3.1.1.

Contributions

This chapter discusses the contributions of the three papers included in this thesis. The proof complexity of random graph coloring plays an important part in all three papers. Paper A and part of Paper C together show that refuting $Col(G, k)$ or $Col_\xi(G, k)$ on a random graph G sampled from either of the distributions $\mathbb{G}(n, d/n)$ or $\mathbb{G}_{n,d}$ asymptotically almost surely requires polynomial calculus degree $n \cdot \exp(-d^{O(1)})$ and size $\exp(n \cdot \exp(-d^{O(1)}))$. For constant d, these degree lower bounds are optimal up to constant factors and the size lower bounds are optimal up to constant factors in the exponent. Paper B shows the same lower bound of $n \cdot \exp(-d^{O(1)})$ for the so-called *level* required for an algorithm called *cohomological k-consistency* to solve graph coloring, which is also asymptotically optimal for d constant. Both Paper B and Paper C contain multiple results other than for random graph coloring, which will be discussed in each respective section.

3.1 Graph Coloring Is Hard on Average for Polynomial Calculus

This paper establishes optimal lower bounds on the polynomial calculus size and degree required to refute $Col(G, k)$ on random graphs. This is the full version of a conference paper [CdRNP+23] that appeared in the *64th Annual IEEE Symposium on Foundations of Computer Science (FOCS '23)*.

3.1.1 Background

The history of the graph coloring problem traces back at least to 1852, when Augustus de Morgan wrote a letter to his friend William Rowan Hamilton containing a formulation of the *Four Color Conjecture* (now a theorem due to Appel and Haken [AH89]) that every planar graph is 4-colorable [Kub04, pp. x-xi]. Ever since then, graph coloring has been a central subarea of graph theory.[1] Graph coloring is also widely studied in theoretical computer science; for instance, as previously noted, it was one of the 21 problems proved NP-complete in the seminal work of Karp [Kar72].

Because graph coloring is such a vast subject both in mathematics and computer science, it is impossible to provide an accurate overview of the whole area here. Hence, this section discusses only the proof complexity of graph coloring and

[1] It has even been called "the most popular area of the subject" by leading experts [Alo00, p. 5], [Alo99, p. 18].

related problems. For more extensive treatments of graph coloring in theory and practice, see for instance [BW15] and [Lew21], respectively.

The study of graph coloring in proof complexity, especially when instantiated on random graphs, has historically followed that of random *constraint satisfaction problems (CSPs)* (see Section 2.5.3), of which random graph coloring is a special case. The study of random CSPs in proof complexity in turn started in earnest with the landmark result of Chvátal and Szemerédi [CS88], showing that randomly sampled k-CNF formulas require exponential size to refute in resolution with overwhelming probability. The work of Chvátal and Szemerédi led to numerous follow-up results, such as resolution size lower bounds for variations of random k-CNF formulas (see for instance [Fu96; BKPS02; ABM04]); extensions of their result to polynomial calculus (and hence also Nullstellensatz) [BI10; AR03]; and the formulation of generally applicable frameworks for proving resolution size lower bounds [BP96; BW01].

Parallel to the work of Chvátal and Szemerédi, influential empirical studies [CKT91; MSL92] found that more general families of random CSPs seemed to provide hard instances for numerous algorithmic paradigms. Mitchell [Mit02a; Mit02b] generalized the framework in [BW01] for proving resolution size lower bounds to many of these CSPs. Roughly concurrently, resolution size lower bounds were proved for important concrete families of random CSPs. Some examples include the independent set problem on random graphs by Beame, Impagliazzo, and Sabharwal [BIS07] and, most importantly for us, for random graph k-colorability by Beame, Culberson, Mitchell, and Moore [BCMM05]. Many of these results were later unified by Molloy and Salvatipour [MS07] and Chan and Molloy [CM13] into fine-grained characterizations of the resolution complexity of different families of random CSPs.

Proving strong size and degree lower bounds for refuting k-colorability in algebraic proof systems remained an open problem for some time after the results of [BCMM05], even though this question was well-motivated by empirical works such as [DLMO09; DLMM11; DMPR+15] showing that algorithms using reasoning captured by algebraic proof systems perform surprisingly well on many challenging benchmarks. A Nullstellensatz degree lower bound of $k+1$ for refuting k-colorability proved by De Loera et al. [DMPR+15] was the best known result until Lauria and Nordström [LN17] proved linear polynomial calculus degree lower bounds, and hence exponential size lower bounds by Theorem 2.4.4. Their hard instance was not a random graph as in [BCMM05], but rather an instance obtained from a reduction to the *functional pigeonhole principle* in Section 2.5.1, for which Mikša and Nordström [MN24] proved a linear degree lower bound. A more general reduction framework due to Atserias and Ochremiak [AO19] later established degree lower bounds for k-colorability in other proof systems, such as *Sherali–Adams* [SA90] and *sum-of-squares* [Sho87; Nes00; Par00; GV01; Las01] capturing linear and semidefinite programming hierarchies respectively, as well as weakly exponential size lower bounds for the *bounded-depth Frege* system [Rec75; CR79].

The proof methods of Lauria and Nordström and Atserias and Ochremiak

do not apply to random graphs, however, and indeed Lauria and Nordström mention extending their result to random graphs as an open problem [LN17, Open Problem 5.2]. The same question was also asked by, for instance, Mikša and Nordström [MN15, Section 5], Lauria [Lau18, Section 4], and Buss and Nordström [BN21, Open Problem 7.6]. Resolving this open problem optimally is the main result of the first paper [CdRNP+25] in this thesis. The results of [CdRNP+25] crucially build on techniques introduced by Romero and Tunçel [RT24]; see Section 3.1.3 for further discussion.

3.1.2 Main Results

The main result of this paper is Theorem 3.1.1.

Theorem 3.1.1. *Let k, d be natural numbers satisfying $3 \leq k < d/4 \log d$. If G is a graph sampled either from the Erdős-Rényi distribution $\mathbb{G}(n, d/n)$ or the uniform distribution over d-regular graphs on n vertices, then asymptotically almost surely, polynomial calculus over any field requires degree $n \cdot \exp(-d^{O(1)})$ to refute $Col(G, k)$.*

Together with the size-degree relation in Theorem 2.4.4, Theorem 3.1.1 implies strongly exponential size lower bounds for refuting $Col(G, k)$ when d is constant.

Corollary 3.1.2. *Let k, d be natural numbers satisfying $3 \leq k < d/4 \log d$. If G is a graph sampled either from the Erdős-Rényi distribution $\mathbb{G}(n, d/n)$ or the uniform distribution over d-regular graphs on n vertices, then asymptotically almost surely, polynomial calculus over any field requires size $\exp(n \cdot \exp(-d^{O(1)}))$ to refute $Col(G, k)$.*

A trivial upper bound on the size required to refute $Col(G, k)$ in polynomial calculus is $k^{O(n)}$, essentially by case analysis of all possible colorings of G. Therefore, when d and k are constant, our size lower bounds are optimal up to coefficients in the exponent. The strength of our lower bound deteriorates rapidly as d grows larger, but our results are non-trivial as long as $d = O(\log^{\varepsilon} n)$ where ε is a small enough positive constant.

As a byproduct of the techniques we introduce, we can also reproduce the lower bounds for resolution of Beame, Culberson, Mitchell, and Moore [BCMM05] using a different proof. Our results for resolution have better dependence on d than in Theorem 3.1.1 and Corollary 3.1.2, almost matching the parameters in [BCMM05]; see the paper for details.

3.1.3 Proof Techniques

To explain what is novel about the proof of Theorem 3.1.1, we first describe the paradigm we follow for proving polynomial calculus degree lower bounds. This approach was introduced by Razborov [Raz98] and refined by Alekhnovich and Razborov [AR03], and is centered around defining a so-called *pseudo-reduction operator* on polynomials. These objects are more commonly called *R-operators* in the literature, but we prefer the terminology above for reasons that will become clear.

Definition 3.1.3 (Pseudo-reduction operator)**.** Let $\mathbb{F}$ be a field, let $D \in \mathbb{N}^+$ and let $\mathcal{P} \subseteq \mathbb{F}[x_1, \ldots, x_n]$. An $\mathbb{F}$-linear operator R on $\mathbb{F}[x_1, \ldots, x_n]$ is a *degree-D pseudo-reduction operator for $\mathcal{P}$* if

1. $R(1) = 1$,
2. $R(p) = 0$ for every polynomial $p \in \mathcal{P}$, and
3. $R(x_i m) = R\big(x_i R(m)\big)$ for every monomial m of degree at most $D - 1$ and every variable x_i.

Intuitively, the kernel of a pseudo-reduction operator provides an overapproximation of what can be derived from $\mathcal{P}$ in degree at most D that is still fine-grained enough to not contain 1. It is not hard to see that the existence of a degree-D pseudo-reduction operator for a set $\mathcal{P}$ of polynomials implies a polynomial calculus degree lower bound of $D + 1$ for refuting $\mathcal{P}$.[2]

Lemma 3.1.4 ([Raz98], Lemma 3.2)**.** *Let $D \in \mathbb{N}^+$ and $\mathcal{P}$ be a set of polynomials over $\mathbb{F}[x_1, \ldots, x_n]$. If there exists a degree-D pseudo-reduction for $\mathcal{P}$, then any polynomial calculus refutation of $\mathcal{P}$ over $\mathbb{F}$ requires degree strictly greater than D.*

Proof sketch. Apply R to all polynomials in a purported polynomial calculus refutation of $\mathcal{P}$ of degree at most D and conclude that it is impossible to reach the constant polynomial 1. □

Proving polynomial calculus degree lower bounds for refuting an unsatisfiable set $\mathcal{P}$ over $\mathbb{F}$ thus reduces to defining a pseudo-reduction operator for $\mathcal{P}$. There is no known way to do this systematically; the operator must be defined case-by-case using the structure of $\mathcal{P}$. However, the notion of reduction modulo ideals in Section 2.1.2 provides a starting point, described next.

Consider the ideal $\langle \mathcal{P} \rangle \subseteq \mathbb{F}[x_1, \ldots, x_n]$ and let $R_{\langle \mathcal{P} \rangle}$ be the reduction operator modulo $\langle \mathcal{P} \rangle$. Our goal in the first place is to prove that the polynomials in $\mathcal{P}$ have no common root, but suppose for a moment that such a common root *does* exist. If so, the operator $R_{\langle \mathcal{P} \rangle}$ is the best pseudo-reduction operator one could hope for. Indeed, $R_{\langle \mathcal{P} \rangle}$ is $\mathbb{F}$-linear by definition; moreover $1 \notin \langle \mathcal{P} \rangle$, so $R_{\langle \mathcal{P} \rangle}(1) = 1$; by definition $R_{\langle \mathcal{P} \rangle}(p) = 0$ for all polynomials in $\mathcal{P}$; and finally, it follows from Fact 2.1.1 that for every variable x_i and every monomial m, of any degree, we have $R_{\langle \mathcal{P} \rangle}(x_i m) = R_{\langle \mathcal{P} \rangle}\big(x_i R_{\langle \mathcal{P} \rangle}(m)\big)$. The upshot is that a pseudo-reduction operator for $\mathcal{P}$ should behave like reduction modulo $\langle \mathcal{P} \rangle$ would *if the polynomials in $\mathcal{P}$ had a common root*. This explains their name and also provides a heuristic for how to define them, which we now describe.

To make the pseudo-reduction operator R as similar to $R_{\langle \mathcal{P} \rangle}$ as possible while ensuring that $R(1) = 1$, Razborov [Raz98] and later Alekhnovich and Razborov [AR03] suggest defining R as follows: to each monomial m, associate

[2] In fact, it can be shown that such an operator exists if and only if refuting $\mathcal{P}$ requires degree strictly greater than D.

a *satisfiable* set $S(m) \subseteq \mathcal{P}$ and define R on m to be reduction modulo $\langle S(m)\rangle$.[3] Finally, define R on arbitrary polynomials by linear extension.

This definition yields an operator R that is $\mathbb{F}$-linear and satisfies $R(1) = 1$. The challenge is now to ensure that Properties 2 and 3 hold, meaning that R maps the polynomials in $\mathcal{P}$ to 0 and that $R(x_i m) = R\big(x_i R(m)\big)$ for low-degree monomials. A central contribution of Alekhnovich and Razborov was to identify a technical condition that ensures both: the set $S(m)$ should be defined to contain all polynomials in $\mathcal{P}$ that are "relevant" to reducing m modulo $\mathcal{P}$, in the sense that adding a "small" set of polynomials to $S(m)$ does not change the result of reduction. More formally, we want that for any set $T \subseteq \mathcal{P}$ such that $T \cup S(m)$ is satisfiable, we have $R_{\langle T \cup S(m)\rangle}(m) = R_{\langle S(m)\rangle}(m)$. We call this condition a *reducibility lemma* for $\mathcal{P}$.

To see the reducibility lemma in action, we now briefly show how it can be used to establish Property 2 of a pseudo-reduction operator. Thinking of $S(m)$ as the set of polynomials in $\mathcal{P}$ relevant to reducing m modulo $\mathcal{P}$, it is natural to ask that $S(m)$ contain all polynomials in $\mathcal{P}$ that share a variable with m. In this case, for a polynomial $p = \sum_j \alpha_j m_j$ in $\mathcal{P}$ we have[4]

$$
\begin{aligned}
R(p) &= \sum_j \alpha_j R_{\langle S(m_j)\rangle}(m_j) && \text{[by definition of } R \text{ and } p\text{]} && \text{(3.1a)}\\
&= \sum_j \alpha_j R_{\langle \bigcup_j S(m_j)\rangle}(m_j) && \text{[by the reducibility lemma]} && \text{(3.1b)}\\
&= R_{\langle \bigcup_j S(m_j)\rangle}\Big(\sum_j \alpha_j m_j\Big) && \text{[by linearity of } R_{\langle \bigcup_j S(m_j)\rangle}\text{]} && \text{(3.1c)}\\
&= R_{\langle \bigcup_j S(m_j)\rangle}(p) && \text{[by definition of } p\text{]} && \text{(3.1d)}\\
&= 0\,. && \text{[since } p \in \textstyle\bigcup_j S(m_j)\text{]} && \text{(3.1e)}
\end{aligned}
$$

Property 3 follows from similar but more complicated reasoning.

Alekhnovich and Razborov managed to prove the reducibility lemma for $\mathcal{P}$ corresponding to so-called *ℓ-immune* polynomials whose polynomial-variable incidence graph is an expander, and Galesi and Lauria [GL10b] used a similar method to obtain optimal separations between size and degree for polynomial calculus. Galesi and Lauria [GL10a] also used the Alekhnovich–Razborov method to establish hardness for *automating* (as defined in [BPR00]) polynomial calculus—that is, to efficiently search for a small polynomial calculus refutation whenever one exists—under reasonable complexity-theoretic assumptions. Mikša and Nordström [MN24] generalized the proof method in [AR03] and obtained a

[3] This heuristic was present already in the paper [Raz98], but there Razborov proves a stronger result that makes pseudo-reduction operators redundant in his particular case. This is best illustrated by the work of Impagliazzo, Pudlák, and Sgall [IPS99], who simplified the proof of the main result of [Raz98] to completely avoid the use of pseudo-reduction operators.

[4] For this calculation we are tacitly assuming that $\bigcup_j S(m_j)$ is satisfiable. This assumption is rather mild and is satisfied by the sets considered in [AR03] and in our results.

widely applicable framework for proving polynomial calculus degree lower bounds, where many prior results follow as special cases. In general, however, the framework of Mikša and Nordström is incomparable to that of [AR03].

After these works, it remained unclear how to prove a reducibility lemma for $Col(G, k)$ when defined over random graphs. Instead, as noted in Section 3.1.1, Lauria and Nordström [LN17] prove linear degree lower bounds for k-colorability through a reduction to the *functional pigeonhole principle* (see Section 2.5), to which the framework in [MN24] applies. Atserias and Ochremiak [AO19] also prove their lower bounds through reductions.

Finally, almost two decades after the resolution lower bounds for $Col(G, k)$ over random graphs by Beame et al. [BCMM05], new ideas by Romero and Tunçel [RT24] enabled progress. Inspired by Erdős' classical result showing that there exist graphs with large girth and chromatic number (see for instance [AS16, pp. 43–44]), they proved polynomial calculus degree lower bounds for refuting $Col(G, k)$ that are linear in the *girth* of the graph G.[5] En route, they found a definition of $S(m)$ that turned out to be extendable to random graphs.

The set $S(m)$ in [RT24] is constructed iteratively through an algorithm that starts with the set of vertices mentioned by m, identifies small, problematic subgraphs in G, adds them to $S(m)$, and then repeats until no such subgraphs remain. Their key insight is that these subgraphs add up to produce an increasingly shorter cycle in the graph. As we increase the girth, the algorithm constructing $S(m)$ terminates quicker, and thus $S(m)$ can be defined for monomials with higher degree.

We use a strengthening of Romero and Tunçel's definition of $S(m)$, but we cannot prove that the algorithm constructing it terminates in few steps in the same way they do since random graphs contain small cycles with constant probability. Instead of assuming that G has large girth, we use that random graphs are asymptotically almost surely (s, ε)-*sparse* (see Section 2.1.3) with excellent parameters, meaning that every small induced subgraph has very low edge density—much lower than that of the whole graph. Our key observation is that adding the problematic subsets to $S(m)$ appends a rather dense subgraph to the current iteration. Even if the set we start with is extremely sparse, these subgraphs rapidly increase the overall density of $S(m)$ to the point where it becomes too dense to exist in G. Thus, we can prove that there are not too many iterations in total.

Given the definition of $S(m)$, our proof of the reducibility lemma follows the same outline as that of Romero and Tunçel. However, demanding that it hold for all $k \geq 3$ on random graphs introduces significant technical complications that Romero and Tunçel do not encounter, which we deal with using novel arguments.

[5] They stated their result as a Nullstellensatz degree lower bound, but it also applies to polynomial calculus.

3.2 Lower Bounds for CSP Hierarchies Through Ideal Reduction

This paper establishes optimal level lower bounds for solving approximate graph coloring with an algorithm called *cohomological k-consistency* due to Ó Conghaile [Ó C22]. This is the publicly available version of a conference paper [CGP26] that appeared in the *37th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA '26)*.

3.2.1 Background

Constraint satisfaction problems are a venerable topic in computer science, with origins in artificial intelligence going back at least to the work of Waltz [Wal72], Montanari [Mon74], and Mackworth [Mac77] and with countless applications in both theory and practice. In theoretical computer science the study of CSPs is a vast area, with natural variants forming complete problems for many of the central complexity classes, but yet this class of problems is still structured enough to often allow for general complexity classifications. In particular, variants of constraint satisfaction problems are among the largest subclasses of NP that are structured enough to avoid NP-*intermediate* problems, i.e., those problems in NP that, supposing P ≠ NP, are not in P but also not NP-complete. If P ≠ NP, then such problems exist by Ladner's Theorem [Lad75].

The first example of such a result was in the seminal paper of Schaefer [Sch78], which showed that all CSPs with Boolean variable domains (such as satisfiability of CNF formulas) are either in P or NP-complete. Hell and Nešetřil [HN90] proved a similar dichotomy theorem for the graph homomorphism problem: they showed that the problem is in P when the template graph is bipartite (in which case the problem is equivalent to 2-coloring) and NP-hard otherwise. Inspired by these results, Feder and Vardi [FV98] conjectured that a similar dichotomy should hold for all fixed-template constraint satisfaction problems.

The two decades following the formulation of the dichotomy conjecture saw an intense investigation into this problem, with important partial results such as an extension of Schaefer's dichotomy to three-element domains [Bul06], a dichotomy for so-called *conservative* CSPs [Bul11], and a characterization of the CSPs solvable by local consistency methods [BK14]. The full conjecture was finally resolved in 2017 by Bulatov [Bul17] and independently Zhuk [Zhu20]. Both papers, as well as the partial results mentioned above, make extensive use of the so-called *algebraic approach* to constraint satisfaction, which studies CSPs through the algebraic structure of their template; for an overview of this area, see the (pre-dichotomy) survey paper [BKW17] and the treatise [Bra22]. Indeed, Bulatov's and Zhuk's results show that each such problem is either in P or NP-complete, and that such problems are tractable precisely when their template exhibits strong enough symmetries, or *polymorphisms*. It was previously known that a lack of a certain type of polymorphism implies hardness, and the contribution of both Bulatov and Zhuk was to provide efficient algorithms when such a polymorphism is present.

Both Bulatov's and Zhuk's algorithms are quite complicated, and their correctness proofs even more so. Moreover, their algorithms take exponential time in the size of the template, which therefore needs to be fixed in advance for the algorithm to be efficient. An outstanding question since their work (see, e.g., [Zhu20, Section 10.1]) is therefore whether there exists a *uniform* CSP algorithm, that is, one that takes the template as part of the input and is efficient whenever the CSP is tractable. Ideally, such an algorithm should also be simpler to analyze than those of Bulatov and Zhuk. Besides its intrinsic importance, the search for such an algorithm has also been motivated by the study of PCSPs, which vastly generalize CSPs and for which a complexity classification remains a distant open problem. Studying the performance of a uniform CSP algorithm on PCSPs would likely help to determine a plausible conjecture regarding the border of tractability for these problems; at the moment, even this remains an active area of research.

In addition to the foundational papers [AGH17; BG21] laying the groundwork for a theory of PCSPs, a considerable body of work, e.g., [BG17; BG19; BGWŽ20; Ó C22; AD22; CŽ23; CŽ25a; CŽ24; CŽ25b; CNP24; DO24; CN25; LP25] has defined and analyzed simpler algorithms for solving (P)CSPs than those of Bulatov and Zhuk. All these algorithms are *hierarchies*, defined with respect to a parameter k called the *level*. Larger values of k yield stronger algorithms but weaker guarantees on worst-case running time, typically $m^{O(k)}$ for instances of size m. While these algorithms are defined for CSPs, they can also be used for solving PCSPs: a CSP algorithm solves a PCSP with template $(\mathbf{S}, \mathbf{T})$ if the algorithm run on $\mathrm{CSP}(\mathbf{S})$ accepts every instance $\mathbf{A}$ such that $\mathbf{A} \to \mathbf{S}$ and rejects each $\mathbf{A}$ such that $\mathbf{A} \not\to \mathbf{T}$ (which is then also an unsatisfiable instance for $\mathrm{CSP}(\mathbf{S})$). Among these hierarchies, one called *cohomological k-consistency* [Ó C22] has recently gained prominence due to the work of Lichter and Pago [LP25]. They define a tractable CSP that most algorithms studied so far fail to solve, thus disqualifying them as candidates for uniform CSP algorithms, and show in addition that cohomological k-consistency correctly solves it.

As previously mentioned, perhaps the most well-studied PCSP is *approximate graph coloring*, also known as *c vs. ℓ-coloring*, where the task is to decide whether an input graph is either c-colorable or not even ℓ-colorable. This problem dates back at least to 1976 [GJ76] and is conjectured to be NP-hard for all $\ell \geq c \geq 3$, but currently we know only that c vs. $(2c-1)$-coloring is NP-hard for $c \geq 3$ [BBKO21] and that c vs. $\binom{c}{\lfloor c/2 \rfloor} - 1$)-coloring is NP-hard for $c \geq 4$ [KOWŽ23]. A central goal in many recent works, e.g., [AD22; CŽ24; CŽ25a; CŽ25b; CNP24; CN25], is to show that the hierarchies defined thus far do not efficiently solve c vs. ℓ-coloring and generalizations, thus providing evidence that the problem is indeed hard. Some partial hardness results for cohomological k-consistency were recently shown by Chan and Ng [CN25] for the same range of c and ℓ for which the problem is known to be NP-hard.

3.2.2 Main Results

The main result of this paper is an optimal cohomological k-consistency level lower bound for solving approximate graph coloring, which follows from proving a level lower bound on random graphs matching the bounds in Theorem 3.1.1.

Theorem 3.2.1. *Asymptotically almost surely as $n \to \infty$, if $6d^3 \leq \log n$ then a random d-regular n-vertex graph has chromatic number at least $d/4 \log d$ but is accepted by the cohomological k-consistency algorithm for 3-colorability for all $k \leq n \cdot d^{-O(d)}$. Consequently, the cohomological k-consistency algorithm does not solve 3 vs. $d/4 \log d$-coloring on n-vertex graphs for any $k \leq n \cdot d^{-O(d)}$.*

Setting d to be a constant in Theorem 3.2.1 yields $k \geq \Omega(n)$, which is asymptotically optimal since cohomological k-consistency (trivially) always correctly solves approximate graph coloring when $k = n$. On the other extreme, we obtain a superconstant lower bound on k for all $d \leq (\log(n)/6)^{1/3}$.

As our second result, we provide a simpler proof of the following lower bound for so-called *lax and null-constraining* CSPs due to Chan and Ng [CN25]. As these conditions are rather technical to define rigorously (although they are satisfied by many natural CSPs), we refer the reader to the paper for details.

Theorem 3.2.2 (Informal, cf. [CN25], Theorem 1.3)**.** *If an n-variable t-CSP $\mathcal{P}$ is non-trivial, lax, and null-constraining, then there exist $\Delta > 0$ and $\delta > 0$ such that, with probability $\delta - o_n(1)$, a randomly sampled instance of $\mathcal{P}$ with Δn constraints is unsatisfiable but is accepted by the cohomological k-consistency algorithm for all $k \leq \Omega(n)$. Consequently, the cohomological k-consistency algorithm does not solve $\mathcal{P}$ for any $k \leq \Omega(n)$.*

We prove Theorem 3.2.2 through similar techniques as sketched in Section 3.1.3, and hence the lower bound in Theorem 3.2.2 also carries over to polynomial calculus degree. While not surprising if one is familiar with the area, this degree lower bound is seemingly new and in particular does not follow from [CN25].

3.2.3 Proof Techniques

To describe our proof techniques, we must first introduce the cohomological k-consistency algorithm. It can be thought of as a blend of two classic techniques in the design of CSP algorithms: *local consistency checking*, which examines whether small subinstances of the CSP are satisfiable and the local solutions are mutually compatible; and *weak global consistency checking*, which relaxes the domain of the CSP to one with some (typically algebraic) structure that enables feasibility to be checked efficiently.

The local consistency checking part of the algorithm is performed using the well-known *k-consistency* heuristic, which checks whether a so-called *k-consistent* set of partial solutions exists for the CSP instance. More precisely, for an instance $\mathbf{A}$ of CSP($\mathbf{T}$), a k-consistent collection κ of partial homomorphisms $\mathbf{A} \to \mathbf{T}$ is one where (1) for each $\varphi \in \kappa$ with domain size smaller than k, there exists for all

$S \in \binom{A}{k}$ some homomorphism $\psi_S \in \kappa$ extending φ to S, and (2) all restrictions of all $\varphi \in \kappa$ to smaller domains are also in κ. Thus, a k-consistent collection of partial homomorphisms "looks like" restrictions of a global homomorphism $\mathbf{A} \to \mathbf{T}$ to small parts of $\mathbf{A}$. Note that unions of k-consistent collections are also k-consistent, so there is a well-defined maximal k-consistent subcollection of any collection of partial homomorphisms.

The weak global consistency part consists of solving the system $L_k(\mathbf{A}, \mathbf{T}, \kappa)$ of linear equations in (3.2a)–(3.2b) over the integers $\mathbb{Z}$. The system $L_k(\mathbf{A}, \mathbf{T}, \kappa)$ is defined with respect to a collection κ of partial homomorphisms $\mathbf{A} \to \mathbf{T}$ and has variables $x_{X,\varphi}$ for each $X \in \binom{A}{\leq k}$ and partial homomorphism $\varphi\colon \mathbf{A} \to \mathbf{T}$ whose domain is X.

$$\sum_{\varphi \in \kappa(X)} x_{X,\varphi} = 1 \qquad \text{for all } X \in \binom{A}{\leq k} \tag{3.2a}$$

$$\sum_{\varphi \in \kappa(X), \varphi\restriction_Y = \psi} x_{X,\varphi} = x_{Y,\psi} \qquad \text{for all } Y \subseteq X \in \binom{A}{\leq k} \text{ and } \psi \in \kappa(Y)\,. \tag{3.2b}$$

Note that if k is at least the maximal arity of a relation in $\mathbf{T}$, then each $\{0,1\}$-valued solution to $L_k(\mathbf{A}, \mathbf{T}, \kappa)$ corresponds to a global homomorphism $\mathbf{A} \to \mathbf{T}$ even when κ is the full set of partial homomorphisms of domain size at most k. However, such solutions, if they exist, cannot be found efficiently in general unless $\mathsf{P} = \mathsf{NP}$. Hence, solving $L_k(\mathbf{A}, \mathbf{T}, \kappa)$ over $\mathbb{Z}$ is indeed a relaxation of solving the instance $\mathbf{A}$ of CSP($\mathbf{T}$).

The cohomological k-consistency algorithm consists of two iterative steps, each of which refines a candidate collection of partial homomorphisms $\mathbf{A} \to \mathbf{T}$. In the first step, the algorithm finds a maximal k-consistent subcollection κ in the current candidate collection. The second step is a "stress test" of sorts for each homomorphism $\varphi \in \kappa$, which consists of solving $L_k(\mathbf{A}, \mathbf{T}, \kappa)$ while fixing φ to 1 and all other homomorphisms in κ with the same domain as φ to 0. Intuitively, this corresponds to committing to the true local solution φ while solving the rest of the problem in the relaxed form given by $L_k(\mathbf{A}, \mathbf{T}, \kappa)$.

More formally, the algorithm description is as follows.

Cohomological k-consistency algorithm for CSP($\mathbf{T}$) *on an instance* $\mathbf{A}$:

1. Maintain a collection $\mathcal{H} = \{\mathcal{H}(X) \mid X \in \binom{A}{\leq k}\}$ of partial homomorphisms $\mathbf{A} \to \mathbf{T}$. For each $X \in \binom{A}{\leq k}$, initialize $\mathcal{H}(X) = \mathrm{Hom}(\mathbf{A}[X], \mathbf{T})$.

2. Repeat until each set $\mathcal{H}(X)$ stabilizes:

   a) Find the maximal k-consistent collection κ in $\mathcal{H}$. Update $\mathcal{H} \leftarrow \kappa$.

   b) For each $X \in \binom{A}{\leq k}$ and $\varphi \in \mathcal{H}(X)$, check whether $L_k(\mathbf{A}, \mathbf{T}, \mathcal{H})$ has a solution satisfying $x_{X,\varphi} = 1$ and $x_{X,\psi} = 0$ for every $\psi \in \mathcal{H}(X) \setminus \{\varphi\}$. If not, remove φ from $\mathcal{H}(X)$ for the next iteration of the loop.

3. If $\mathcal{H}(X) = \emptyset$ for some $X \in \binom{A}{\leq k}$, then reject; otherwise accept.

The parameter k is called the *level*. Already the k-consistency heuristic always correctly decides whether $\mathbf{A} \to \mathbf{T}$ when $k = |A|$ (but then does not necessarily run in time polynomial in $\mathbf{A}$). Hence, the same is true for cohomological k-consistency, and a level lower bound of $\Omega(|A|)$ is therefore asymptotically optimal.

Since cohomological k-consistency is a relaxation, it always accepts satisfiable instances of the original CSP. To show a lower bound on the level required for cohomological k-consistency to solve CSP($\mathbf{T}$), we should therefore exhibit a *fooling instance* $\mathbf{A}$ of CSP($\mathbf{T}$), which is one such that $\mathbf{A} \not\to \mathbf{T}$ but the algorithm still accepts $\mathbf{A}$ for small k. More specifically, we want to find an instance $\mathbf{A} \not\to \mathbf{T}$ and a collection κ of partial homomorphisms $\mathbf{A} \to \mathbf{T}$ such that κ is k-consistent and each partial restriction of $L_k(\mathbf{A}, \mathbf{T}, \kappa)$ in Step 2b) of cohomological k-consistency has a solution.

Fooling Instances from Algebraic Proof Complexity: The main technical contribution of the paper is a rather general result that shows that such collections of partial homomorphisms are implicit in any pseudo-reduction operator for the polynomial encoding $\mathcal{P}_{\mathbf{A}\to\mathbf{T}}$ in (2.18a)–(2.18d), whenever this operator is constructed via the Alekhnovich–Razborov method of local reduction operators sketched in Section 3.1.3. In a bit more detail, recall that $\mathcal{P}_{\mathbf{A}\to\mathbf{T}}$ is over the set of Boolean variables $\mathbf{x}_{A,T} = \{x_{a,i} \mid a \in A, i \in T\}$. Any (partial) function $\varphi : A \to T$ can then be identified with a monomial $m_\varphi = \prod_{(a,i)\in\varphi} x_{a,i}$. Suppose that $\mathbf{A} \not\to \mathbf{T}$, that $\mathbb{F}$ has characteristic 0, and that $\mathcal{P}_{\mathbf{A}\to\mathbf{T}}$ admits a degree-D local pseudo-reduction operator R as in Section 3.1.3, where for each monomial m we associate a substructure $\mathbf{A}(m)$ of $\mathbf{A}$ from the vertices of A mentioned by m, and define $R(m) = R_{\langle \mathcal{P}_{\mathbf{A}(m)\to\mathbf{T}}\rangle}(m)$. We show that:

1. the functions $\varphi : A \to T$ such that $R(m_\varphi) \neq 0$ form a D-consistent collection κ_R of partial homomorphisms;

2. the assignment $x_{X,\varphi} \mapsto R(m_\varphi)$ is an $\mathbb{F}[\mathbf{x}_{A,T}]$-valued solution to $L_k(\mathbf{A}, \mathbf{T}, \kappa_R)$;

3. for each monomial m of degree at most D, the polynomial $R(m)$ has integer coefficients if each local reduction operator is defined with respect to a lexicographic variable order.

Consequently, composing the map $x_{X,\varphi} \mapsto R(m_\varphi)$ with any evaluation on a point in $\mathbb{Z}^{|A|\cdot|T|}$, which we know is then also in $\mathbb{F}^{|A|\cdot|T|}$ since $\mathbb{F}$ is of characteristic 0, yields a solution to $L_k(\mathbf{A}, \mathbf{T}, \kappa_R)$ over $\mathbb{Z}$.

To find solutions to the partially restricted versions of $L_k(\mathbf{A}, \mathbf{T}, \kappa_R)$ in Step 2b of the algorithm, we therefore have to find for each φ in κ_R a point z in $\mathbb{Z}^{|A|\cdot|T|}$ such that $R(m_\varphi)(z) = 1$ and $R(m_\psi)(z) = 0$ for all $\psi \in \kappa_R$ with the same domain as φ. For this, we use the fact that the substructures $\mathbf{A}(m_\varphi)$ are the same for all partial homomorphisms with the same domain. We have by definition that $R(m_\varphi)$

is the unique linear combination of irreducible terms modulo $\langle \mathcal{P}_{\mathbf{A}(m_\varphi)\to\mathbf{T}} \rangle$ such that

$$m_\varphi = R(m_\varphi) + q \tag{3.3}$$

for some $q \in \langle \mathcal{P}_{\mathbf{A}(m_\varphi)\to\mathbf{T}} \rangle$. Hence, for any (indicator vector of a) homomorphism $\gamma \colon \mathbf{A}(m_\varphi) \to \mathbf{T}$, we have that $R(m_\varphi)(\gamma) = m_\varphi(\gamma)$. Thus, if we take γ to be a partial homomorphism extending φ to the domain of $\mathbf{A}(m_\varphi)$, then $R(m_\varphi)(\gamma) = 1$ and $R(m_\psi)(\gamma) = 0$ for all other homomorphisms with the same domain as φ.

The first two of the three items above follow readily from the definition of $\mathcal{P}_{\mathbf{A}\to\mathbf{T}}$ and the properties in Definition 3.1.3. The proof of the third item uses the so-called *lex game* of Felszeghy, Ráth, and Rónyai [FRR06]. The lex game over a polynomial ring $\mathbb{F}[x_1, \ldots, x_n]$ with a lexicographic monomial order $\prec$ has two players named Lea and Stan, and is defined with respect to a (finite) set $V \subseteq \mathbb{F}^n$ and monomial m. The main result of [FRR06] is that Lea has a winning strategy in the lex game if and only if the vanishing ideal $I(V)$ of V contains a polynomial with leading term m with respect to $\prec$. The key to showing Item 3 using the lex game is that this winning strategy can be encoded as a polynomial p in $I(V)$, which has leading term m and turns out to have integer coefficients when $V \subseteq \{0,1\}^n$. Therefore, we have that $R_{I(V)}(m) = R_{I(V)}(m - p)$, and since each term in the polynomial $m - p$ is smaller than m and has integer coefficients, we can use induction on $\prec$ and linearity to show that $R_{I(V)}(m)$ also has integer coefficients. We can then use the finite-domain Nullstellensatz in Theorem 2.4.3 to show that the ideals $\langle \mathcal{P}_{\mathbf{A}(m_\varphi)\to\mathbf{T}} \rangle$ coincide with the vanishing ideals of their roots, and hence that this integrality result applies to such ideals.

Applying the Connection: Given this connection to polynomial calculus degree lower bounds, we can obtain our level lower bound for approximate graph coloring directly from the degree lower bound for random graph coloring in Theorem 3.1.1, which is proved via the Alekhnovich–Razborov method. For our lower bounds for random instances of lax and null-constraining CSPs, it suffices to prove a reducibility lemma as in Section 3.1.3 for such instances. For this we can define the set $S(m)$ for each monomial m using a construction from [CN25] called the *BW closure*. The results of [CN25, Sections 6-7] establish something very similar to a reducibility lemma but in different terminology. Hence, we mostly need to reinterpret their technical results to obtain our alternative proof of their lower bounds. This simplifies the argument considerably and bypasses the need for the (quite involved) machinery in [CN25, Section 8].

3.3 On the Power of Non-Boolean Polynomial Calculus

This paper is a systematic study of size lower bounds for polynomial calculus over non-Boolean variables. We introduce a unified perspective that extends existing techniques [Sok20; IMP23]. With this framework as a starting point, we prove multiple new size lower bounds for polynomial calculus over non-Boolean variables, as well as a simulation result showing that certain non-Boolean bases

are, surprisingly, much stronger than others for refuting, e.g., the pigeonhole principle.

3.3.1 Background

All polynomial calculus size lower bounds discussed so far hold only for problem encodings over Boolean variables. For proof systems operating with propositional formulas, such as resolution, this is essentially the only choice, but more expressive mathematical objects such as polynomials over a field might offer other alternatives. For graph k-colorability in particular, the roots-of-unity-valued polynomials in (2.16a)–(2.16b) form a natural non-Boolean problem encoding.

Bayer suggested applying the Gröbner basis algorithm to the roots-of-unity encoding to solve graph coloring, and it is also the encoding used in the NuLA algorithm [DLMO09; DLMM11; DMPR+15] mentioned in Section 3.1.1. To prove lower bounds on the running time of these algebraic algorithms, we therefore need to prove a polynomial calculus size lower bound for refuting the roots-of-unity encoding of graph coloring. This question, as well as proving size lower bounds over non-Boolean variable domains in general, is also motivated by the fact that many stronger proof systems than polynomial calculus can effectively simulate affine shifts between variable domains. Hence, to prove lower bounds for such proof systems, a necessary step is to prove polynomial calculus size lower bounds that hold regardless of the choice of variable domain.

Perhaps surprisingly, such results do not follow in any straightforward way from the known lower bounds for the Boolean case; size lower bounds in proof complexity are in general notoriously encoding-dependent. For some problems, exponential savings in proof size can come from simple adjustments such as introducing dual variables [dRLNS21], or, as is the case for the Tseitin contradictions in Section 2.5.2, switching from Boolean to $\{\pm 1\}$-valued variables [Urq87; Gri98]. These differences can also materialize in practical automated theorem proving. For instance, the key to the computer-assisted proof in [SH23] of a long-standing conjecture in mathematics was to find a particularly suitable problem encoding.

Further illustrating this point, the usual methods for proving size lower bounds completely break down over non-Boolean variables. The first size lower bounds in this setting were therefore obtained only relatively recently by Sokolov [Sok20]; his results are for randomly sampled k-CNF formulas and for the (graph) pigeonhole principle, and hold for polynomial calculus with $\{\pm 1\}$-valued variables. However, his techniques break down for the functional pigeonhole principle.

Impagliazzo, Mouli, and Pitassi [IMP23] extended Sokolov's techniques and established lower bounds for polynomial calculus over finite fields, even allowing a bounded number of extension variables, each of which depends on a limited number of original variables. Dahiya, Mahajan, and Mouli [DMM24] later extended the results of [Sok20] and improved the parameters in [IMP23], albeit

for a different formula.[6]

Mouli [Mou24] exhibited a formula that requires exponential size to refute over {±1}-valued variables but has refutations of polynomial size over Boolean variables. Since a separation in the other direction was already known for the previously mentioned Tseitin contradictions, his results show that polynomial calculus over {±1}-valued variables is incomparable to polynomial calculus over Boolean variables with respect to size.

Bonacina, Galesi, and Lauria [BGL23] introduced and studied sum-of-squares over the complex numbers $\mathbb{C}$ and proved size lower bounds for a formula whose variables take values equal to a root of unity. By a simulation in the same paper, their result implies a size lower bound for polynomial calculus over $\mathbb{C}$ for the same formula.

There are also many recent results in a related but somewhat different direction. One such result, by Alekseev [Ale21], establishes lower bounds on the bit complexity of polynomial calculus refutations with unconstrained extension variables. Another line of work, e.g., [FSTW21; AF22; GHT22; HLT24; BLRS25; CGMS25; EGLT25], shows size lower bounds for fragments of the *ideal proof system (IPS)* of Grochow and Pitassi [GP18]. While all these results hold for much stronger proof systems than are considered here, they are shown for problems that cannot be expressed as unsatisfiable CNF formulas and hence fall outside the scope of propositional proof systems as originally defined by Cook and Reckhow [CR79].

Before this paper, no strong polynomial calculus size lower bounds were known for the roots-of-unity encoding of graph coloring, nor for the functional pigeonhole principle over {±1}-valued variables. Already Bayer [Bay82, Chapter 3, Sections 2.3-2.6] was interested in the complexity of the Gröbner basis algorithm for k-colorability in the roots-of-unity encoding; proving lower bounds on running time in this setting reduces to establishing a polynomial calculus size lower bound for this problem. In terms of proof complexity, this problem was highlighted by Lauria [Lau18, p. 260], and explicitly stated as an open problem by Lauria and Nordström [LN17, Open Problem 5.1] and Bonacina, Galesi, and Lauria [BGL23, Section 6]. Proving size lower bounds for the functional pigeonhole principle over {±1}-valued variables was in turn posed as an open problem by Sokolov [Sok20, Section 7].

3.3.2 Main Results

Our first results are size lower bounds for the roots-of-unity encodings of graph coloring and the functional pigeonhole principle, resolving the open problems in Section 3.3.1.

Theorem 3.3.1 (Informal)**.** *For fixed integers k, d satisfying $3 \leq k < d/4 \log d$, and a graph G on n vertices that is either a random d-regular graph or an Erdős-Rényi random graph with edge probability d/n, polynomial calculus over any field of characteristic either*

[6] Currently, there is a gap in both [IMP23] and [DMM24] that affects how these papers relate to our results; see Paper C for a discussion.

0 *or not dividing* k *asymptotically almost surely as* $n \to \infty$ *requires size* $\exp(n \cdot d^{O(1)})$ *to refute that* G *is* k*-colorable when expressed in the roots-of-unity encoding.*

For constant d, the lower bound in Theorem 3.3.1 is optimal up to constant factors in the exponent.

Theorem 3.3.2 (Informal)**.** *If a bipartite graph* $G = (U \sqcup V, E)$ *is a good boundary expander with* $|V| = n$, $|U| = O(n)$*, and constant left degree, then every polynomial calculus refutation of the graph functional pigeonhole principle on* G *over* $\{\pm 1\}$*-valued variables requires size* $\exp(\Omega(n))$*.*

Just as for Theorem 3.3.1, the lower bound in Theorem 3.3.2 is optimal up to constants in the exponent.

Our framework also allows us to prove that it is NP-hard to *automate* [BPR00] polynomial calculus augmented with limited extension variables. More formally, let $\mathrm{PCR}_{\{\pm 1\}\cup\{0,1\}}$ denote polynomial calculus resolution over a field $\mathbb{F}$ of characteristic distinct from 2, where each variable $x_i, \overline{x}_i$ also has a $\{\pm 1\}$-valued counterpart $y_i, \overline{y}_i$. Our result shows that efficiently searching for $\mathrm{PCR}_{\{\pm 1\}\cup\{0,1\}}$-refutations is impossible if $\mathsf{P} \neq \mathsf{NP}$.

Theorem 3.3.3 (Informal)**.** *If there is an algorithm that, given any unsatisfiable CNF formula* F*, finds a* $\mathrm{PCR}_{\{\pm 1\}\cup\{0,1\}}$*-refutation of* F *in time polynomial in the minimum monomial size among all* $\mathrm{PCR}_{\{\pm 1\}\cup\{0,1\}}$*-refutations of* F*, then* $\mathsf{P} = \mathsf{NP}$*.*

We prove Theorem 3.3.3 through an approach pioneered in the breakthrough work of Atserias and Müller [AM20] and further developed in [GKMP20; Bel20; dRGNP+21; dRez21; IR22; Gar24; Pap24; CdR25], where NP-hardness of automating a proof system reduces to proving a size lower bound for refuting a certain meta-complexity-theoretic formula in that proof system; see Section 3.3.4 for more details.

Finally, as a complement to our lower bounds for polynomial calculus over roots of unity, we show that polynomial calculus over *non*-roots of unity efficiently simulates the *cutting planes* proof system [CCT87] with coefficients bounded in magnitude by a polynomial in the number of variables. This fragment of cutting planes is usually denoted CP^* and was first considered in [BPR97; BC96].

Theorem 3.3.4 (Informal)**.** *Let* $c \neq 0$ *be a non-root of unity in a field* $\mathbb{F}$ *of characteristic* 0*. If cutting planes refutes a set* $\mathcal{L}$ *of linear inequalities in length* s *and coefficient magnitude* t*, then polynomial calculus over* $\mathbb{F}$ *with* $\{1, c\}$*-valued variables can refute a natural polynomial encoding of* $\mathcal{L}$ *in size* $s \cdot t^{O(1)}$*.*

By rescaling, Theorem 3.3.4 holds also when replacing $\{1, c\}$ by $\{a, b\}$ where $ab \neq 0$ and $a^{-1}b$ is not a root of unity. Since CP^* efficiently refutes the pigeonhole principle, combining Theorem 3.3.4 with Theorem 3.3.2 (or [Sok20, Theorem 6.3]) yields a separation between polynomial calculus over roots of unity and its counterpart over non-roots of unity. This result complements the separations in [Urq87; Gri98; Mou24] between Boolean and $\{\pm 1\}$-valued variables and provides

another example of the shifts in reasoning power that can arise from simple encoding adjustments. Since our framework can be used to prove lower bounds for the pigeonhole principle, the simulation in Theorem 3.3.4 is a barrier against extending our techniques to also include non-roots of unity.

Conversely, Theorem 3.3.4 also points to a new direction toward size lower bounds for CP*. All existing techniques are based on special kinds of reductions to lower bounds for other computational models: either for monotone circuits through *feasible interpolation* [Kra97; BPR97; Pud97; HC99; BEGJ00; FHL16; HP17; FPPR22], for other proof systems via *lifting* [GGKS20; GKRS19], or for communication protocols [Sok24]. The types of formulas amenable to such reductions are quite limited; for example, these techniques seem inapplicable for the long-standing open problem of proving cutting planes (and even CP*) size lower bounds for random k-CNF formulas when k is constant. Sokolov [Sok20] established polynomial calculus size lower bounds for this problem over $\{\pm 1\}$-valued variables. While some other simulations of CP* by algebraic proof systems are known [RT08; IMP20] (some of which use similar ideas as in our simulation; see Section 3.3.5 for a discussion), proving size lower bounds for polynomial calculus over non-roots of unity seems more approachable than for the proof systems considered in [RT08; IMP20].

3.3.3 Framework for Lower Bounds

To explain the challenges of proving the lower bounds in this paper, it is instructive to contrast our size lower bounds for k-colorability in the roots-of-unity encoding with the proof of Corollary 3.1.2. As is typical, the proof of Corollary 3.1.2 proceeds by proving a polynomial calculus degree lower bound and then arguing that it implies a size lower bound. This is how the proof of our result works as well, but the setting of non-Boolean variables presents multiple challenges that do not arise in the Boolean case.

By an observation of Lauria and Nordström [LN17, Proposition 2.2][7] it turns out that a degree lower bound in the Boolean encoding (2.15a)–(2.15d) implies the same degree lower bound for the roots-of-unity encoding (2.16a)–(2.16b). The proof consists of a linear substitution of the variables in $Col_{\xi}(G, k)$ by those in $Col(G, k)$ together with a line-by-line simulation argument. The substitution does not preserve size, however, so we cannot deduce Theorem 3.3.1 from Corollary 3.1.2 in this way.

For low-degree formulas over Boolean variables, and in particular for the Boolean encoding $Col(G, k)$ of k-colorability, a polynomial calculus degree lower bound implies a size lower bound in a black-box fashion through the size-degree relation (Theorem 2.4.4). Unfortunately, the size-degree relation is false for formulas over non-Boolean variables, as witnessed by the $\{\pm 1\}$-valued encoding (2.13a)–(2.13b) of Tseitin formulas on expander graphs. These formulas require linear polynomial calculus degree over fields of characteristic differ-

[7] For a proof, see instead [LN23, Proposition 2.2].

ent from 2 [BGIP01, Theorem 10], but can be refuted in polynomial size [Gri98, Lemma 1b]. Therefore, any proof that converts a degree lower bound for $Col_\xi(G,k)$ to a size lower bound must use properties specific to $Col_\xi(G,k)$.

Since there is no general size-degree relation over the roots of unity, we must use other methods. Both the size-degree relation and the related size-width relation for resolution (see Theorem 2.4.2) can be seen as abstractions of a paradigm for proving size lower bounds through degree or width lower bounds, which is implicit in early works on resolution such as [Gal77; Hak85; Urq87; CS88] and was more explicitly spelled out by Beame and Pitassi [BP96] and later by Ben-Sasson and Wigderson [BW01]. We present this approach in the next paragraph for resolution (for simplicity), but it applies also to polynomial calculus after some minor technical adjustments, with degree replacing size and monomials replacing clauses. We use an adaptation of it for our lower bounds over non-Boolean variables.

The proof of the size-degree relation in [IPS99] follows the same strategy as that of [CEI96, Theorem 10], which shows that a small *resolution* refutation can be turned into a polynomial calculus refutation of low degree. Inspired by [CEI96], Beame and Pitassi [BP96] contrapositively observe that many earlier resolution size lower bounds can be obtained from width lower bounds by turning a small refutation π of a formula F over the variables $x_1,\dots,x_n$ into a low-width refutation of $F\restriction_\rho$, where ρ is a small restriction. If the restricted formula cannot be refuted in low width, this contradicts that π is small.

In more detail, consider a resolution refutation π of F that is of length S, and let $\mathrm{High}_w(\pi)$ denote the set of clauses in π of width at least w. Clearly $|\mathrm{High}_w(\pi)| \le S$, and by an averaging argument some literal ℓ_i appears in at least a w/n fraction of the clauses in $\mathrm{High}_w(\pi)$. Therefore, by applying a restriction ρ to π setting the variable x_i of ℓ_i to either 0 or 1, we can satisfy a $w/2n$ fraction of the clauses in $\mathrm{High}_w(\pi)$, and the result is a resolution refutation $\pi\restriction_\rho$ of $F\restriction_\rho$ with at most $(1-w/2n)\cdot S$ clauses of width at least w. Repeating this argument m times yields a refutation $\pi\restriction_{\rho_m}$ of $F\restriction_{\rho_m}$ with at most $(1-w/2n)^m \cdot S$ clauses of width at least w. If $\pi\restriction_{\rho_m}$ cannot be refuted in width w, it follows that $S > (1-w/2n)^{-m} \ge \exp(wm/2n)$.

It is clear where this argument fails if the variables are not Boolean: assigning a variable can no longer satisfy a large portion of high-width clauses (or remove high-degree monomials). Therefore, there is no hope of directly relating size to degree in this way over non-Boolean variables. Instead of relating size to degree or width as above, we consider an auxiliary complexity measure we call *quadratic degree*, following Sokolov [Sok20] and Impagliazzo, Mouli, and Pitassi [IMP23]. Quadratic degree is measured on *monomial pairs* in a given polynomial and, very roughly speaking, quantifies the amount of "interaction" between the two monomials from the perspective of the input set of polynomials. We then relate quadratic degree to both size and degree separately. An overview can be found in Figure 3.1.

The first part of the proof, which is rather simple and generic, is to show that a refutation of low quadratic degree can be turned into one of low degree. The second and more involved part is to show that a small refutation of $Col_\xi(G,k)$ can

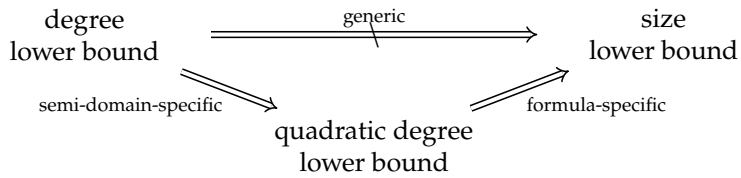

Figure 3.1: *An overview of the proof of Theorem 3.3.1. The link between degree and quadratic degree only depends on the domain axioms of the formula under consideration, while the link between quadratic degree and size uses properties specific to* $Col_\xi(G, k)$.

be turned into one of low quadratic degree by replicating the argument above with quadratic degree in place of width or degree.

The starting point of relating size to quadratic degree is to find a variable x_v that contributes to many pairs of high quadratic degree in π, just as in the argument above. However, instead of restricting x_v we apply a more complicated operation called *splitting on* x_v to π. Very roughly speaking, splitting on x_v involves dividing π into cases based on the value of x_v and then combining these into a single refutation so that many monomial pairs of high quadratic degree are cancelled out. The idea behind splitting is due to Sokolov [Sok20], but the version we use is closer to that of Impagliazzo, Mouli, and Pitassi [IMP23].

To be able to put the pieces back into a formula that retains the original formula's structure and hence remains hard to refute, the variable x_v that was split on must be "irrelevant" to the original refutation, which for us means that it only appears in the refuted formula in polynomials contained in an ideal generated by a univariate polynomial with multiple roots. A given variable will however not be irrelevant to π, so before splitting on x_v we apply a small restriction ρ to $Col_\xi(G, k)$ that isolates x_v from the rest of the polynomials in $Col_\xi(G, k)$. It is here that the structure of $Col_\xi(G, k)$ is crucial, since the restriction ρ must simultaneously isolate the variable x_v and preserve the hardness of $Col_\xi(G, k)\restriction_\rho$.

Constructing the restriction ρ is one of the main technical challenges in our size lower bound for graph coloring. Our restriction builds on machinery from the paper [CdRNP+25] in Section 3.1. Inspired by Romero and Tunçel [RT24], it was shown in [CdRNP+25] that in a random graph G with vertex set V, every small set $U \subseteq V$ can be covered by a slightly larger set, called the *closure of* U and denoted by $\mathrm{cl}(U)$, such that all vertices in $V \setminus \mathrm{cl}(U)$ have at most one neighbor inside $\mathrm{cl}(U)$ and the subgraph consisting of vertices close to $\mathrm{cl}(U)$ in $G[V \setminus \mathrm{cl}(U)]$ is a forest. Therefore, partially coloring $\mathrm{cl}(U)$ has only a small impact on the set of proper colorings of $G[V \setminus \mathrm{cl}(U)]$. Once we have identified a variable $x_v \in \mathit{Vars}(Col_\xi(G, k))$ to split on, we can isolate it from the rest of $Col_\xi(G, k)$ by applying a partial coloring χ to $\mathrm{cl}(N(v) \cup \{v\})$ but leaving x_v unassigned. Since partial colorings of closures do not affect the rest of the graph too much, we are able to prove that $Col_\xi(G, k)\restriction_\chi$ remains hard to refute.

The overall proof structure for Theorem 3.3.2 is the same as in this overview, but there the innovation is mostly in the definition of quadratic degree and splitting

that can accommodate the functionality axioms; those in [Sok20] immediately turn the functional pigeonhole principle into a formula that is refutable in constant size. The proof of Theorem 3.3.3 also follows a similar outline, but with some additional modifications that we discuss in Section 3.3.4. A key technical contribution of this paper is to provide general, abstract treatments of quadratic degree and splitting, which specialize to the ad-hoc definitions in prior work in each specific case but also instantiate into useful notions for formulas where the definitions in prior work do not suffice. This abstract treatment also allows us to appeal to general results instead of directly establishing the straightforward but tedious parts of proofs with the same outline as above, such as showing that refutations of low quadratic degree can be turned into refutations of low degree.

3.3.4 Non-Automatability of Polynomial Calculus with Extension Variables

A proof system P is *automatable* [BPR00] if there exists an algorithm that, given an unsatisfiable CNF formula F, finds a P-refutation of F in time polynomial in the size of the smallest such refutation. Here we consider the proof system $\mathrm{PCR}_{\{\pm1\}\cup\{0,1\}}$, which is polynomial calculus resolution augmented with $\{\pm1\}$-valued versions of all variables. To show that $\mathrm{PCR}_{\{\pm1\}\cup\{0,1\}}$ is not automatable, the idea behind the approach of [AM20], which we follow, is to construct an efficiently computable function A that maps a given k-CNF formula F to another CNF formula (or in our case, a set of polynomials) $A(F)$, such that:

1. If F is satisfiable, then $A(F)$ has a $\mathrm{PCR}_{\{\pm1\}\cup\{0,1\}}$-refutation of size $\mathrm{poly}(|F|, n)$.

2. If F is unsatisfiable, then any $\mathrm{PCR}_{\{\pm1\}\cup\{0,1\}}$-refutation of $A(F)$ requires size at least $\exp\left(|F|^{\Omega(1)}\right)$.

If such a function A exists and $\mathrm{PCR}_{\{\pm1\}\cup\{0,1\}}$ is automatable, we can decide in polynomial time whether F is satisfiable: simply run the algorithm automating $\mathrm{PCR}_{\{\pm1\}\cup\{0,1\}}$ on $A(F)$ for a (large enough) polynomial number of steps. If the automating algorithm finds a refutation of $A(F)$, then F is satisfiable by Item 1; and if the automating algorithm does not halt, then by Item 2, F is unsatisfiable. Hence, if such a function A exists, then $\mathrm{PCR}_{\{\pm1\}\cup\{0,1\}}$ is not automatable unless $\mathsf{P} = \mathsf{NP}$.

The idea of [AM20] is to define A by mapping a given formula F to the meta-complexity-theoretic formula $REF(s, F)$ encoding the statement "F has a resolution refutation of size s."[8] If F is satisfiable, then known results [Pud03; dRGNP+21] show that $REF(s, F)$ has a polynomial calculus refutation of size $\mathrm{poly}(s, |F|)$, which establishes Item 1 when setting $s = \mathrm{poly}(|F|)$.

The technical crux is to prove the size lower bound in Item 2 when F is unsatisfiable. As in [dRGNP+21], we prove this lower bound for a *lifted* version of

[8] We actually use a variant of this formula, but we ignore this difference in this informal overview.

$REF(s, F)$, which replaces each variable by a certain *gadget* function depending on a small number of new variables. As shown in [dRGNP+21], the formula $REF(s, F)$ remains easy to refute after this transformation if F is satisfiable.

The formula $REF(s, F)$ comes with a natural partition of its variables into *blocks*, reflecting the structure of the purported refutation of F. Building on [AM20], it was shown in [dRGNP+21] that any polynomial calculus refutation of $REF(s, F)$ for unsatisfiable F contains a monomial that mentions many blocks; in other words, refuting the formula $REF(s, F)$ requires high *block degree*. They then prove a generic block-degree to size *lifting theorem*, which shows that if refuting a set $\mathcal{P}$ requires high block degree, then for an appropriate gadget function, the lifted set of polynomials $\text{Lift}(\mathcal{P})$ requires large monomial size to refute.

We show that the lifting theorem in [dRGNP+21] holds even when size is measured in the proof system $\text{PCR}_{\{\pm1\}\cup\{0,1\}}$ instead of polynomial calculus. The proof in [dRGNP+21] is by a random restriction argument, showing that for any small polynomial calculus refutation π of $\text{Lift}(\mathcal{P})$ there exists a variable assignment to $\text{Lift}(\mathcal{P})$ that simplifies the formula back to $\mathcal{P}$ while killing all monomials in π of large block degree. Since the proof system $\text{PCR}_{\{\pm1\}\cup\{0,1\}}$ has both $\{0, 1\}$-valued and $\{\pm1\}$-valued variables, we cannot hope to kill monomials by only restricting variables. Instead, we again define an appropriate notion of quadratic degree that we call *block quadratic degree*, and then apply a random *restriction-or-splitting* which either restricts $\{0, 1\}$-valued variables or splits on $\{\pm1\}$-valued variables. We show that in a small $\text{PCR}_{\{\pm1\}\cup\{0,1\}}$-refutation of $\mathcal{P}$, such an operation is likely to substantially decrease the block quadratic degree of each line while simplifying $\text{Lift}(\mathcal{P})$ back to $\mathcal{P}$. Finally, we appeal to our abstract framework to show that a refutation of low block quadratic degree can be turned into one of low block degree, completing the proof of the lifting theorem.

3.3.5 Simulating Cutting Planes with Bounded Coefficients

The cutting planes proof system operates with linear inequalities. In more detail, a linear inequality L is a predicate $\ell(x) \geq b$ for a linear form $\ell(x) = \sum_i a_i x_i$ with coefficients $a_i \in \mathbb{Z}$ and n Boolean variables $x_i \in \{0, 1\}$. Given a non-root of unity c in a field $\mathbb{F}$, we can define $\{1, c\}$-valued variables $y_i = c^{x_i}$. The key to Theorem 3.3.4 is then to define, for each inequality L, a polynomial P_L in the variables y_i, whose monomial size is polynomial in n and which evaluates to 0 on $\{1, c\}$-valued assignments to the y_i's precisely when the corresponding $\{0, 1\}$-valued assignment to the x_i's satisfies L.

By introducing dual variables $\overline{x}_i$ satisfying $\overline{x}_i + x_1 = 1$, we can rewrite ℓ to have nonnegative coefficients. If the coefficients of ℓ are polynomially bounded in n, then the evaluations of ℓ that satisfy L are contained in the polynomially bounded interval $[b, \|L\|]$, where $\|L\| = \sum_i |a_i|$. Since $y_i = c^{x_i}$, writing $m_\ell = \prod_{i\in[n]} y_i^{a_i}$ we can then define P_L as the polynomial

$$\prod_{d\in[b,\|L\|]} (m_\ell - c^d) = \prod_{d\in[b,\|L\|]} (c^{\ell(x)} - c^d) \tag{3.4}$$

which contains only polynomially many distinct terms; in particular, all terms in P_L are powers of m_ℓ with exponent at most $\|L\|$. Since c is not a root of unity, the map $z \mapsto c^z$ is injective and hence the linear inequality L is satisfied if and only if $P_L = 0$. Using this encoding, the proof of Theorem 3.3.4 proceeds by simulating each line of a CP* proof, replacing each linear inequality L by P_L.

We learned of this encoding transformation from [AP15]. The same encoding seems to have been independently discovered in [IMP20], and is used in [IMP20, Theorem 5] to prove that polynomial calculus over finite fields of large enough characteristic with extension variables represented by low-depth algebraic circuits simulates CP*. The same paper also shows an analogous simulation result over characteristic 0, but there a different encoding is used [IMP20, Theorem 4].

3.4 Conclusions and Future Work

In this thesis we prove new lower bounds, non-automatability results, and simulations for algebraic proof systems. We also show that polynomial calculus degree lower bounds established through the *Alekhnovich–Razborov method* transfer to the cohomological k-consistency algorithm, and hence also to weaker CSP hierarchies. Many natural related questions in both directions remain unanswered. We highlight a few of these below, and refer to the concluding remarks of each included paper for more.

Lower Bounds for (Random) CSPs: Random graph k-colorability is an instance of a *random constraint satisfaction problem* (see Section 2.5.3). As mentioned in Section 3.1.1, Molloy and Salvatipour [MS07] and Chan and Molloy [CM13] achieve fine-grained characterizations of the resolution hardness of rather general families of (sparse) random constraint satisfaction problems; in particular, k-colorability on sparse random graphs belongs to the family studied in [CM13].

There are clear similarities between the properties of random constraint satisfaction problems used in [MS07; CM13] and the properties of random (hyper)graphs needed for our results in Papers A, B, and C, but hardness characterizations as general as in [MS07; CM13] are not known for polynomial calculus. Moreover, the characterization would necessarily need to be different from that in [CM13], since some problems satisfying the hardness criterion in that paper (for instance, random k-XOR equations) are easy for polynomial calculus, at least over some fields. Could our techniques, perhaps in conjunction with the field-dependent hardness criterion in [AR03], help prove such a characterization?

More concretely, it is currently open to prove optimal polynomial calculus size and degree lower bounds for many graph problems over random graphs, such as independent set and perfect matching. Can our techniques be useful for these problems? By the connection to CSP hierarchies established in Paper B, such results for (P)CSPs on random graphs would be of interest both in proof complexity and in CSP theory.

A specific PCSP for which such results would be very interesting is *approximate graph homomorphism*, which is defined with respect to a pair of graphs (G, H) such that $G \to H$. The task is to decide whether an input graph is either homomorphic to G or not even to H. This problem generalizes c vs. ℓ-coloring, where G and H are complete graphs on c and ℓ vertices, respectively. As mentioned in Section 3.2.1, a classical result in CSP theory (later subsumed by the Dichotomy Theorem [Bul17; Zhu20]) is the *Hell–Nešetřil Dichotomy* [HN90], which states that the graph homomorphism problem is NP-hard if the template graph is not bipartite and in P otherwise. The *Brakensiek–Guruswami Conjecture* [BG21, Conjecture 1.2] aims to extend this result to the PCSP setting and states that approximate graph homomorphism with template (G, H) is NP-hard whenever G is not bipartite (in which case H also is not bipartite).

It is not hard to show that the Brakensiek–Guruswami Conjecture is true if and only if it is true whenever G is an odd cycle and H is a complete graph on at least 3 vertices. For similar reasons as in the discussion of Theorem 3.2.1, it would in turn suffice to show that random instances of graph homomorphism with an odd-cycle template are hard to refute for polynomial calculus to establish the Brakensiek–Guruswami Conjecture unconditionally for this proof system, and hence also for algorithms captured by it.[9] This formulation of the problem appears in Alexander Razborov's well-regarded list of open problems [Raz26, Open Problem 16]. It seems likely that such a result would be established using the Alekhnovich–Razborov method, in which case the lower bounds would carry over to cohomological k-consistency and weaker hierarchies by the results of Paper B.

More Lower Bounds over Non-Boolean Bases: Size lower bounds for polynomial calculus over non-Boolean variables is still a nascent area. Over roots of unity and finite fields (where each non-zero element is a root of unity), the work in this thesis and the papers mentioned in Section 3.3.1 together provide a rich variety of hard instances. Over infinite fields, much less is known for other domains than the roots of unity. A concrete open problem, which the results of Paper C strongly suggest is worthwhile and fundamental, is to prove any size lower bound at all for polynomial calculus over non-roots of unity, say over the reals and with $\{1, 2\}$-valued variables. By the simulation result in Paper C it seems, however, that new ideas are needed to tackle this problem.

It would deepen our understanding of polynomial calculus to also develop a systematic framework for proving size lower bounds over non-Boolean variables. Paper C makes progress in this direction, but a key part of the proof structure in Section 3.3.3 remains problem-specific. Over Boolean variables, the frameworks for polynomial calculus degree lower bounds due to Alekhnovich and

[9] Note, however, that for odd cycles with at least 5 vertices this cannot hold with high probability: with uniformly positive probability, a sparse random graph contains a triangle, which is an unsatisfiable instance of length-(≥ 5) odd-cycle graph homomorphism even by itself. On such graphs, one can refute the problem on the triangle subgraph by brute force in constant size.

Razborov [AR03] and Mikša and Nordström [MN24] both provide general, easily applicable methods for proving polynomial calculus size lower bounds when combined with the size-degree relation (see Theorem 2.4.4). Can we establish a similar framework for formulas over non-Boolean variables? A potential approach to this question is as follows.

There is no size-degree relation for polynomial calculus over non-Boolean variables (see Section 3.3.3) and we only know how to prove size lower bounds via degree lower bounds. Therefore, any such framework would need to identify a general property of formulas for which a degree lower bound can be transferred to a size lower bound over non-Boolean variables. A natural candidate for such a property is *self-reducibility*. Roughly speaking, a self-reducible formula is one where any small subset of variables can be restricted (possibly by setting a few more variables) in such a way that the restricted formula is a smaller version of the original formula.

Self-reducibility is ubiquitous in proof complexity; for instance, the key to the proof of Theorems 3.3.1 and 3.3.2 is to show that the roots-of-unity encoding of k-colorability on random graphs and the functional pigeonhole on boundary expanders are both self-reducible in a weak sense. Other size lower bounds over non-Boolean variables, such as those in [Sok20] and [IMP23], can also be interpreted this way. A suitable formalization of self-reducibility along with existing frameworks for degree lower bounds could therefore lead to a reasonably general framework for polynomial calculus size lower bounds over non-Boolean variables.

References

[Aar16] Scott Aaronson, $\mathsf{P} \stackrel{?}{=} \mathsf{NP}$, in *Open Problems in Mathematics*, Springer, Cham, 2016, pages 1–122. [9]

[AB09] Sanjeev Arora and Boaz Barak, *Computational Complexity: A Modern Approach*. Cambridge University Press, Cambridge, 2009, pages xxiv+579. [8, 9]

[ABM04] Dimitris Achlioptas, Paul Beame, and Michael Molloy, A sharp threshold in proof complexity yields lower bounds for satisfiability search, *Journal of Computer and System Sciences*, vol. 68, no. 2, pages 238–268, 2004, Preliminary version in STOC '01. [28]

[ABRW02] Michael Alekhnovich, Eli Ben-Sasson, Alexander A. Razborov, and Avi Wigderson, Space complexity in propositional calculus, *SIAM Journal on Computing*, vol. 31, no. 4, pages 1184–1211, Apr. 2002, Preliminary version in *STOC '00*. [18]

[AD22] Albert Atserias and Víctor Dalmau, Promise constraint satisfaction and width, in *Proceedings of the 33rd Annual ACM-SIAM Symposium on Discrete Algorithms (SODA '22)*, Jan. 2022, pages 1129–1153. [34]

[AF22] Robert Andrews and Michael A. Forbes, Ideals, determinants, and straightening: Proving and using lower bounds for polynomial ideals, in *Proceedings of the 54th Annual ACM SIGACT Symposium on Theory of Computing (STOC '22)*, Jun. 2022, pages 389–402. [40]

[AGH17] Per Austrin, Venkatesan Guruswami, and Johan Håstad, $(2 + \varepsilon)$-SAT is NP-hard, *SIAM Journal on Computing*, vol. 46, no. 5, pages 1554–1573, 2017. [34]

[AH89] Kenneth Appel and Wolfgang Haken, *Every Planar Map Is Four Colorable* (Contemporary Mathematics). American Mathematical Society, Providence, RI, 1989, vol. 98, pages xvi+741, With the collaboration of J. Koch. [27]

[Ale21] Yaroslav Alekseev, A lower bound for polynomial calculus with extension rule, in *Proceedings of the 36th Computational Complexity Conference (CCC '21)*, ser. Leibniz International Proceedings in Informatics (LIPIcs), vol. 200, 2021, pages 21:1–21:18. [40]

[Alo00] Noga Alon, Algebraic and probabilistic methods in discrete mathematics, in *Visions in Mathematics*, Noga Alon, Jean Bourgain, Alain Connes, Mikhail Gromov, and Vitali Milman, Eds., Birkhäuser Basel, 2000, pages 455–470. [27]

[Alo99] Noga Alon, Combinatorial Nullstellensatz, *Combinatorics, Probability and Computing*, vol. 8, no. 1-2, pages 7–29, 1999. [15, 27]

[AM20] Albert Atserias and Moritz Müller, Automating resolution is NP-hard, *Journal of the ACM*, vol. 67, no. 5, pages 31:1–31:17, Oct. 2020, Preliminary version in *FOCS '19*. [41, 45, 46]

[AO19] Albert Atserias and Joanna Ochremiak, Proof complexity meets algebra, *ACM Transactions on Computational Logic*, vol. 20, pages 1:1–1:46, Feb. 2019, Preliminary version in *ICALP '17*. [28, 32]

[AP15] Kristoffer Arnsfelt Hansen and Vladimir V. Podolskii, Polynomial threshold functions and Boolean threshold circuits, *Information and Computation*, vol. 240, pages 56–73, Feb. 2015, Preliminary version in *MFCS '13*. [47]

[AR03] Michael Alekhnovich and Alexander A. Razborov, Lower bounds for polynomial calculus: Non-binomial case, *Proceedings of the Steklov Institute of Mathematics*, vol. 242, pages 18–35, 2003, Available at `http://people.cs.uchicago.edu/~razborov/files/misha.pdf`. Preliminary version in *FOCS '01*. [21, 28–32, 47, 49]

[AS16] Noga Alon and Joel H. Spencer, *The Probabilistic Method*, Fourth edition. John Wiley & Sons, Inc., Hoboken, NJ, 2016, pages xiv+375. [32]

[Bay82] David Allen Bayer, The division algorithm and the Hilbert scheme, Available at `https://www.math.columbia.edu/~bayer/papers/Bayer-thesis.pdf`, Ph.D. dissertation, Harvard University, Jun. 1982. [23, 40]

[BBKO21] Libor Barto, Jakub Bulín, Andrei Krokhin, and Jakub Opršal, Algebraic approach to promise constraint satisfaction, *Journal of the ACM*, vol. 68, no. 4, pages 28:1–28:66, Aug. 2021. [34]

[BC96] Samuel R. Buss and Peter Clote, Cutting planes, connectivity and threshold logic, *Archive for Mathematical Logic*, vol. 35, pages 33–63, 1996. [41]

[BCMM05] Paul Beame, Joseph C. Culberson, David G. Mitchell, and Cristopher Moore, The resolution complexity of random graph k-colorability, *Discrete Applied Mathematics*, vol. 153, no. 1-3, pages 25–47, Dec. 2005. [28, 29, 32]

[BEGJ00] María Luisa Bonet, Juan Luis Esteban, Nicola Galesi, and Jan Johannsen, On the relative complexity of resolution refinements and cutting planes proof systems, *SIAM Journal on Computing*, vol. 30, no. 5, pages 1462–1484, 2000, Preliminary version in *FOCS '98*. [42]

[Bel20] Zoë Bell, "Automating regular or ordered resolution is NP-hard," Electronic Colloquium on Computational Complexity (ECCC), Technical Report TR20-105, Jul. 2020. [41]

[BG17] Christoph Berkholz and Martin Grohe, Linear diophantine equations, group CSPs, and graph isomorphism, in *Proceedings of the Twenty-Eighth Annual ACM-SIAM Symposium on Discrete Algorithms (SODA '17)*, 2017, pages 327–339. [34]

[BG19] Joshua Brakensiek and Venkatesan Guruswami, An algorithmic blend of LPs and ring equations for promise CSPs, in *Proceedings of the 2019 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA '19)*, 2019, pages 436–455. [34]

[BG21] Joshua Brakensiek and Venkatesan Guruswami, Promise constraint satisfaction: Algebraic structure and a symmetric boolean dichotomy, *SIAM Journal on Computing*, vol. 50, no. 6, pages 1663–1700, 2021, Preliminary version in *SODA '18*. [34, 48]

[BGIP01] Samuel R. Buss, Dima Grigoriev, Russell Impagliazzo, and Toniann Pitassi, Linear gaps between degrees for the polynomial calculus modulo distinct primes, *Journal of Computer and System Sciences*, vol. 62, no. 2, pages 267–289, Mar. 2001, Preliminary version in *CCC '99*. [22, 43]

[BGL23] Ilario Bonacina, Nicola Galesi, and Massimo Lauria, On vanishing sums of roots of unity in polynomial calculus and sum-of-squares, *Computational Complexity*, vol. 32, no. 12, 2023. [40]

[BGWŽ20] Joshua Brakensiek, Venkatesan Guruswami, Marcin Wrochna, and Stanislav Živný, The power of the combined basic linear programming and affine relaxation for promise constraint satisfaction problems, *SIAM Journal on Computing*, vol. 49, no. 6, pages 1232–1248, 2020, Preliminary version in *SODA '20*. [34]

[BI10] Eli Ben-Sasson and Russell Impagliazzo, Random CNF's are hard for the polynomial calculus, *Computational Complexity*, vol. 19, no. 4, pages 501–519, 2010, Preliminary version in *FOCS '99*. [22, 28]

[BIKP+94] Paul Beame, Russell Impagliazzo, Jan Krajíček, Toniann Pitassi, and Pavel Pudlák, Lower bounds on Hilbert's Nullstellensatz and propositional proofs, in *Proceedings of the 35th Annual IEEE Symposium on Foundations of Computer Science (FOCS '94)*, Nov. 1994, pages 794–806. [15]

[BIKP+97] Samuel R. Buss, Russell Impagliazzo, Jan Krajíček, Pavel Pudlák, Alexander A. Razborov, and Jiří Sgall, Proof complexity in algebraic systems and bounded depth Frege systems with modular counting, *Computational Complexity*, vol. 6, no. 3, pages 256–298, 1997. [17]

[BIS07] Paul Beame, Russell Impagliazzo, and Ashish Sabharwal, The resolution complexity of independent sets and vertex covers in random graphs, *Computational Complexity*, vol. 16, no. 3, pages 245–297, Oct. 2007, Preliminary version in *CCC '01*. [28]

[BJK05] Andrei Bulatov, Peter Jeavons, and Andrei Krokhin, Classifying the complexity of constraints using finite algebras, *SIAM Journal on Computing*, vol. 34, no. 3, pages 720–742, 2005. [24]

[BK14] Libor Barto and Marcin Kozik, Constraint satisfaction problems solvable by local consistency methods, *Journal of the ACM*, vol. 61, no. 1, Jan. 2014, Preliminary version in *FOCS '09*. [33]

[BKPS02] Paul Beame, Richard Karp, Toniann Pitassi, and Michael Saks, The efficiency of resolution and Davis-Putnam procedures, *SIAM Journal on Computing*, vol. 31, no. 4, pages 1048–1075, 2002, Preliminary versions of these results appeared in *FOCS '96* and *STOC '98*. [28]

[BKW17] Libor Barto, Andrei Krokhin, and Ross Willard, Polymorphisms, and how to use them, in *The Constraint Satisfaction Problem: Complexity and Approximability*, ser. Dagstuhl Follow-Ups, vol. 7, Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2017, pages 1–44. [33]

[Bla37] Archie Blake, Canonical expressions in Boolean algebra, Ph.D. dissertation, University of Chicago, 1937. [14]

[BLRS25] Amik Raj Behera, Nutan Limaye, Varun Ramanathan, and Srikanth Srinivasan, New bounds for the ideal proof system in positive characteristic, in *Proceedings of the 52nd International Colloquium on Automata, Languages, and Programming (ICALP '25)*, ser. Leibniz International Proceedings in Informatics (LIPIcs), vol. 334, Jul. 2025, pages 22:1–22:20. [40]

[BN21] Samuel R. Buss and Jakob Nordström, Proof complexity and SAT solving, in *Handbook of Satisfiability*, ser. Frontiers in Artificial Intelligence and Applications, Armin Biere, Marijn J. H. Heule, Hans van Maaren, and Toby Walsh, Eds., 2nd, vol. 336, IOS Press, Feb. 2021, ch. 7, pages 233–350. [13, 14, 17, 29]

[BP96] Paul Beame and Toniann Pitassi, Simplified and improved resolution lower bounds, in *Proceedings of the 37th Annual IEEE Symposium on Foundations of Computer Science (FOCS '96)*, Oct. 1996, pages 274–282. [28, 43]

[BPR00] María Luisa Bonet, Toniann Pitassi, and Ran Raz, On interpolation and automatization for Frege systems, *SIAM Journal on Computing*, vol. 29, no. 6, pages 1939–1967, 2000. [31, 41, 45]

[BPR97] María Bonet, Toniann Pitassi, and Ran Raz, Lower bounds for cutting planes proofs with small coefficients, *Journal of Symbolic Logic*, vol. 62, no. 3, pages 708–728, Sep. 1997, Preliminary version in *STOC '95*. [41, 42]

[BR98] Paul Beame and Søren Riis, More on the relative strength of counting principles, in *Proof Complexity and Feasible Arithmetics (Rutgers, NJ, 1996)*, ser. DIMACS Series in Discrete Mathematics and Theoretical Computer Science, vol. 39, American Mathematical Society, Providence, RI, 1998, pages 13–35. [21]

[Bra22] Zarathustra Brady, *Notes on CSPs and polymorphisms*, `arXiv:2210.07383`, Oct. 2022. [33]

[Bul06] Andrei A. Bulatov, A dichotomy theorem for constraint satisfaction problems on a 3-element set, *Journal of the ACM*, vol. 53, no. 1, pages 66–120, Jan. 2006, Preliminary version in *FOCS '02*. [33]

[Bul11] Andrei A. Bulatov, Complexity of conservative constraint satisfaction problems, *ACM Transactions on Computational Logic*, vol. 12, no. 4, Jul. 2011, Preliminary version in *LICS '03*. [33]

[Bul17] Andrei A. Bulatov, A dichotomy theorem for nonuniform CSPs, in *Proceedings of the 58th Annual IEEE Symposium on Foundations of Computer Science (FOCS '17)*, 2017, pages 319–330. [33, 48]

[BW01] Eli Ben-Sasson and Avi Wigderson, Short proofs are narrow—resolution made simple, *Journal of the ACM*, vol. 48, no. 2, pages 149–169, Mar. 2001, Preliminary version in *STOC '99*. [14, 18, 28, 43]

[BW15] Lowell W. Beineke and Robert J. Wilson, Eds., *Topics in Chromatic Graph Theory* (Encyclopedia of Mathematics and its Applications). Cambridge University Press, 2015. [28]

[CCT87] William Cook, Collette Rene Coullard, and György Turán, On the complexity of cutting-plane proofs, *Discrete Applied Mathematics*, vol. 18, no. 1, pages 25–38, Nov. 1987. [41]

[CdR25] Gaia Carenini and Susanna F. de Rezende, On the automatability of tree-like k-DNF resolution, in *Proceedings of the 40th Computational Complexity Conference (CCC '25)*, ser. Leibniz International Proceedings in Informatics (LIPIcs), vol. 339, 2025, pages 14:1–14:21. [41]

[CdRNP+23] Jonas Conneryd, Susanna F. de Rezende, Jakob Nordström, Shuo Pang, and Kilian Risse, Graph colouring is hard on average for polynomial calculus and Nullstellensatz, in *Proceedings of the 64th Annual IEEE Symposium on Foundations of Computer Science (FOCS '23)*, Nov. 2023, pages 1–11. [27]

[CdRNP+25] Jonas Conneryd, Susanna F. de Rezende, Jakob Nordström, Shuo Pang, and Kilian Risse, *Graph colouring is hard on average for polynomial calculus and Nullstellensatz*, `arXiv:2503.17022`. Preliminary version in *FOCS '23*., 2025. [29, 44]

[CEI96] Matthew Clegg, Jeffery Edmonds, and Russell Impagliazzo, Using the Groebner basis algorithm to find proofs of unsatisfiability, in *Proceedings*

of the 28th Annual ACM Symposium on Theory of Computing (STOC '96), May 1996, pages 174–183. [17, 43]

[CGMS25] Prerona Chatterjee, Utsab Ghosal, Partha Mukhopadhyay, and Amit Sinhababu, IPS lower bounds for formulas and sum of ROABPs, in *Proceedings of the 45th IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science (FSTTCS '25)*, ser. Leibniz International Proceedings in Informatics (LIPIcs), vol. 360, Dec. 2025, pages 22:1–22:21. [40]

[CGP26] Jonas Conneryd, Yassine Ghannane, and Shuo Pang, Lower bounds for CSP hierarchies through ideal reduction, in *Proceedings of the 37th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA '26)*, Jan. 2026, pages 6436–6463. [33]

[CJW06] James Carlson, Arthur Jaffe, and Andrew Wiles, Eds., *The Millennium Prize Problems*. Clay Mathematics Institute, Cambridge, MA; American Mathematical Society, Providence, RI, 2006, pages xviii+165. [9]

[CKT91] Peter Cheeseman, Bob Kanefsky, and William M. Taylor, Where the really hard problems are, in *Proceedings of the 12th International Joint Conference on Artificial Intelligence (IJCAI '91) - Volume 1*, Morgan Kaufmann Publishers Inc., 1991, pages 331–337. [28]

[CM13] Siu On Chan and Michael Molloy, A dichotomy theorem for the resolution complexity of random constraint satisfaction problems, *SIAM Journal on Computing*, vol. 42, no. 1, pages 27–60, 2013, Preliminary version in *FOCS '08*. [28, 47]

[CN10] Stephen Cook and Phuong Nguyen, *Logical Foundations of Proof Complexity* (Perspectives in Logic). Cambridge University Press, Cambridge; Association for Symbolic Logic, La Jolla, CA, 2010, pages xvi+479. [13]

[CN25] Siu On Chan and Hiu Tsun Ng, "How random CSPs fool hierarchies: II," Electronic Colloquium on Computational Complexity (ECCC), Technical Report TR25-026, 2025, Preliminary version in *STOC '25*. [34, 35, 38]

[CNP24] Siu On Chan, Hiu Tsun Ng, and Sijin Peng, How random CSPs fool hierarchies, in *Proceedings of the 56th Annual ACM Symposium on Theory of Computing (STOC '24)*, Jun. 2024, pages 1944–1955. [34]

[Coo71] Stephen A. Cook, The complexity of theorem-proving procedures, in *Proceedings of the Third Annual ACM Symposium on Theory of Computing (STOC '71)*, Association for Computing Machinery, 1971, pages 151–158. [10]

[Coo75] Stephen A. Cook, Feasibly constructive proofs and the propositional calculus (preliminary version), in *Proceedings of the 7th Annual ACM Symposium on Theory of Computing (STOC '75)*, May 1975, pages 83–97. [12]

[CR74] Stephen A. Cook and Robert A. Reckhow, On the lengths of proofs in the propositional calculus (preliminary version), in *Proceedings of the 6th Annual ACM Symposium on Theory of Computing (STOC '74)*, Apr. 1974, pages 135–148. [11]

[CR79] Stephen A. Cook and Robert A. Reckhow, The relative efficiency of propositional proof systems, *Journal of Symbolic Logic*, vol. 44, no. 1, pages 36–50, Mar. 1979, Preliminary version in *STOC '74*. [11, 19, 22, 28, 40]

[CS88] Vašek Chvátal and Endre Szemerédi, Many hard examples for resolution, *Journal of the ACM*, vol. 35, no. 4, pages 759–768, Oct. 1988. [28, 43]

[CŽ23] Lorenzo Ciardo and Stanislav Živný, CLAP: A new algorithm for promise CSPs, *SIAM Journal on Computing*, vol. 52, no. 1, pages 1–37, 2023, Preliminary version in *SODA '22*. [34]

[CŽ24] Lorenzo Ciardo and Stanislav Živný, *The periodic structure of local consistency*, `arXiv:2406.19685.`, 2024. [34]

[CŽ25a] Lorenzo Ciardo and Stanislav Živný, Approximate graph coloring and the crystal with a hollow shadow, *SIAM Journal on Computing*, vol. 54, no. 4, pages 1138–1192, 2025, Combined full version of two papers published in *STOC '23* and *SODA '23*, respectively. [34]

[CŽ25b] Lorenzo Ciardo and Stanislav Živný, Semidefinite programming and linear equations vs. homomorphism problems, *SIAM Journal on Computing*, vol. 54, no. 3, pages 545–584, 2025, Preliminary version in *STOC '24*. [34]

[DLL62] Martin Davis, George Logemann, and Donald Loveland, A machine program for theorem proving, *Communications of the ACM*, vol. 5, no. 7, pages 394–397, Jul. 1962. [14]

[DLMM11] Jesús A. De Loera, Jon Lee, Peter N. Malkin, and Susan Margulies, Computing infeasibility certificates for combinatorial problems through Hilbert's Nullstellensatz, *Journal of Symbolic Computation*, vol. 46, no. 11, pages 1260–1283, Nov. 2011. [28, 39]

[DLMO09] Jesús A. De Loera, Jon Lee, Susan Margulies, and Shmuel Onn, Expressing combinatorial problems by systems of polynomial equations and Hilbert's Nullstellensatz, *Combinatorics, Probability and Computing*, vol. 18, no. 4, pages 551–582, Jul. 2009. [28, 39]

[DMM24] Yogesh Dahiya, Meena Mahajan, and Sasank Mouli, New lower bounds for polynomial calculus over non-Boolean bases, in *Proceedings of the 27th International Conference on Theory and Applications of Satisfiability Testing (SAT '24)*, ser. Leibniz International Proceedings in Informatics (LIPIcs), vol. 305, 2024, pages 10:1–10:20. [39, 40]

[DMPR+15] Jesús A. De Loera, Susan Margulies, Michael Pernpeintner, Eric Riedl, David Rolnick, Gwen Spencer, Despina Stasi, and Jon Swenson, Graph-coloring ideals: Nullstellensatz certificates, Gröbner bases for chordal graphs, and hardness of Gröbner bases, in *Proceedings of the 40th International Symposium on Symbolic and Algebraic Computation (ISSAC '15)*, Jul. 2015, pages 133–140. [28, 39]

[DO24] Víctor Dalmau and Jakub Opršal, Local consistency as a reduction between constraint satisfaction problems, in *Proceedings of the 39th Annual ACM/IEEE Symposium on Logic in Computer Science (LICS '24)*, Association for Computing Machinery, 2024. [34]

[DP60] Martin Davis and Hilary Putnam, A computing procedure for quantification theory, *Journal of the ACM*, vol. 7, no. 3, pages 201–215, 1960. [14]

[dRez21] Susanna F. de Rezende, Automating tree-like resolution in time $n^{o(\log n)}$ is ETH-hard, *Procedia Computer Science*, vol. 195, pages 152–162, 2021, Special issue for the Proceedings of the XI Latin and American Algorithms, Graphs and Optimization Symposium (LAGOS '21). [41]

[dRGNP+21] Susanna F. de Rezende, Mika Göös, Jakob Nordström, Toniann Pitassi, Robert Robere, and Dmitry Sokolov, Automating algebraic proof systems is NP-hard, in *Proceedings of the 53rd Annual ACM Symposium on Theory of Computing (STOC '21)*, Jun. 2021, pages 209–222. [41, 45, 46]

[dRLNS21] Susanna F. de Rezende, Massimo Lauria, Jakob Nordström, and Dmitry Sokolov, The power of negative reasoning, in *Proceedings of the 36th Annual Computational Complexity Conference (CCC '21)*, ser. Leibniz International Proceedings in Informatics (LIPIcs), vol. 200, Jul. 2021, pages 40:1–40:24. [18, 39]

[dRNRS25] Susanna F. de Rezende, Jakob Nordström, Kilian Risse, and Dmitry Sokolov, Exponential resolution lower bounds for weak pigeonhole principle and perfect matching formulas over sparse graphs, *TheoretiCS*, vol. 4, 2025, Preliminary version in *CCC '20*. [20]

[EGLT25] Tal Elbaz, Nashlen Govindasamy, Jiaqi Lu, and Iddo Tzameret, *Lower bounds against the ideal proof system in finite fields*, `arXiv:2506.17210v1`, 2025. [40]

[FHL16] Yuval Filmus, Pavel Hrubeš, and Massimo Lauria, Semantic versus syntactic cutting planes, in *Proceedings of the 33rd Symposium on Theoretical Aspects of Computer Science (STACS 16)*, ser. Leibniz International Proceedings in Informatics (LIPIcs), vol. 47, 2016, pages 35:1–35:13. [42]

[FKP19] Noah Fleming, Pravesh Kothari, and Toniann Pitassi, Semialgebraic proofs and efficient algorithm design, *Foundations and Trends in Theoretical Computer Science*, vol. 14, no. 1–2, pages 1–221, Dec. 2019. [13]

[FPPR22] Noah Fleming, Denis Pankratov, Toniann Pitassi, and Robert Robere, Random $\Theta(\log n)$-CNFs are hard for cutting planes, *Journal of the ACM*, vol. 69, no. 3, pages 19:1–19:32, Jun. 2022, Preliminary version in *FOCS '17*. [42]

[FRR06] Bálint Felszeghy, Balázs Ráth, and Lajos Rónyai, The lex game and some applications, *Journal of Symbolic Computation*, vol. 41, no. 6, pages 663–681, 2006. [38]

[FSTW21] Michael A. Forbes, Amir Shpilka, Iddo Tzameret, and Avi Wigderson, Proof complexity lower bounds from algebraic circuit complexity, *Theory of Computing*, vol. 17, no. 10, pages 1–88, Nov. 2021, Preliminary version in *CCC '16*. [40]

[Fu96] Xudong Fu, On the complexity of proof systems, Ph.D. dissertation, University of Toronto, 1996. [28]

[FV98] Tomás Feder and Moshe Y. Vardi, The computational structure of monotone monadic SNP and constraint satisfaction: A study through Datalog and group theory, *SIAM Journal on Computing*, vol. 28, no. 1, pages 57–104, 1998, Preliminary version in *STOC '93*. [24, 33]

[Gal77] Zvi Galil, On resolution with clauses of bounded size, *SIAM Journal on Computing*, vol. 6, no. 3, pages 444–459, 1977, Preliminary version in *STOC '75*. [22, 43]

[Gar24] Michal Garlík, Failure of feasible disjunction property for k-DNF resolution and NP-hardness of automating it, in *Proceedings of the 39th Computational Complexity Conference (CCC '24)*, ser. Leibniz International Proceedings in Informatics (LIPIcs), vol. 300, Jul. 2024, pages 33:1–33:23. [41]

[GGKS20] Ankit Garg, Mika Göös, Pritish Kamath, and Dmitry Sokolov, Monotone circuit lower bounds from resolution, *Theory of Computing*, vol. 16, no. 13, pages 1–30, 2020, Preliminary version in *STOC '18*. [42]

[GHT22] Nashlen Govindasamy, Tuomas Hakoniemi, and Iddo Tzameret, Simple hard instances for low-depth algebraic proofs, in *Proceedings of the 63rd Annual IEEE Symposium on Foundations of Computer Science (FOCS '22)*, Nov. 2022, pages 188–199. [40]

[GJ76] Michael Garey and David S. Johnson, The complexity of near-optimal graph coloring, *Journal of the ACM*, vol. 23, no. 1, pages 43–49, Jan. 1976. [24, 34]

[GJ79] Michael R. Garey and David S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman and Company, 1979. [10]

[GKMP20] Mika Göös, Sajin Koroth, Ian Mertz, and Toniann Pitassi, Automating cutting planes is NP-hard, in *Proceedings of the 52nd Annual ACM Symposium on Theory of Computing (STOC '20)*, Jun. 2020, pages 68–77. [41]

[GKRS19] Mika Göös, Pritish Kamath, Robert Robere, and Dmitry Sokolov, Adventures in monotone complexity and TFNP, in *Proceedings of the 10th Innovations in Theoretical Computer Science Conference (ITCS '19)*, ser. Leibniz International Proceedings in Informatics (LIPIcs), vol. 124, Jan. 2019, pages 38:1–38:19. [42]

[GL10a] Nicola Galesi and Massimo Lauria, On the automatizability of polynomial calculus, *Theory of Computing Systems*, vol. 47, no. 2, pages 491–506, Aug. 2010. [31]

[GL10b] Nicola Galesi and Massimo Lauria, Optimality of size-degree trade-offs for polynomial calculus, *ACM Transactions on Computational Logic*, vol. 12, no. 1, pages 4:1–4:22, Nov. 2010. [31]

[GP18] Joshua A. Grochow and Toniann Pitassi, Circuit complexity, proof complexity, and polynomial identity testing: The ideal proof system, *Journal of the ACM*, vol. 65, no. 6, pages 37:1–37:59, Nov. 2018, Preliminary version in *FOCS '14*. [40]

[Gri98] Dima Grigoriev, Tseitin's tautologies and lower bounds for Nullstellensatz proofs, in *Proceedings of the 39th Annual IEEE Symposium on Foundations of Computer Science (FOCS '98)*, Nov. 1998, pages 648–652. [22, 39, 41, 43]

[GV01] Dima Grigoriev and Nicolai Vorobjov, Complexity of Null- and Positivstellensatz proofs, *Annals of Pure and Applied Logic*, vol. 113, no. 1–3, pages 153–160, Dec. 2001. [28]

[Hak85] Armin Haken, The intractability of resolution, *Theoretical Computer Science*, vol. 39, no. 2-3, pages 297–308, Aug. 1985. [20, 43]

[HC99] Armin Haken and Stephen A. Cook, An exponential lower bound for the size of monotone real circuits, *Journal of Computer and System Sciences*, vol. 58, no. 2, pages 326–335, Apr. 1999. [42]

[Hil93] David Hilbert, Ueber die vollen Invariantensysteme, *Mathematische Annalen*, vol. 42, no. 3, pages 313–373, 1893. [15]

[HLT24] Tuomas Hakoniemi, Nutan Limaye, and Iddo Tzameret, Functional lower bounds in algebraic proofs: Symmetry, lifting, and barriers, in *Proceedings of the 56th Annual ACM Symposium on Theory of Computing (STOC '24)*, Jun. 2024, pages 1396–1404. [40]

[HLW06] Shlomo Hoory, Nathan Linial, and Avi Wigderson, Expander graphs and their applications, *Bulletin of the American Mathematical Society*, vol. 43, no. 4, pages 439–561, Oct. 2006. [7]

[HN90] Pavol Hell and Jaroslav Nešetřil, On the complexity of H-coloring, *Journal of Combinatorial Theory, Series B*, vol. 48, no. 1, pages 92–110, 1990. [33, 48]

[HP17] Pavel Hrubeš and Pavel Pudlák, Random formulas, monotone circuits, and interpolation, in *Proceedings of the 58th Annual IEEE Symposium on Foundations of Computer Science (FOCS '17)*, Oct. 2017, pages 121–131. [42]

[IMP20] Russell Impagliazzo, Sasank Mouli, and Toniann Pitassi, The surprising power of constant depth algebraic proofs, in *Proceedings of the 35th Annual ACM/IEEE Symposium on Logic in Computer Science (LICS '20)*, Association for Computing Machinery, 2020, pages 591–603. [42, 47]

[IMP23] Russell Impagliazzo, Sasank Mouli, and Toniann Pitassi, Lower bounds for polynomial calculus with extension variables over finite fields, in *38th Computational Complexity Conference (CCC '23)*, ser. Leibniz International Proceedings in Informatics (LIPIcs), vol. 264, 2023, pages 7:1–7:24. [38–40, 43, 44, 49]

[IPS99] Russell Impagliazzo, Pavel Pudlák, and Jiří Sgall, Lower bounds for the polynomial calculus and the Gröbner basis algorithm, *Computational Complexity*, vol. 8, no. 2, pages 127–144, 1999. [18, 31, 43]

[IR22] Dmitry Itsykson and Artur Riazanov, Automating OBDD proofs is NP-hard, in *Proceedings of the 47th International Symposium on Mathematical Foundations of Computer Science (MFCS '22)*, ser. Leibniz International Proceedings in Informatics (LIPIcs), vol. 241, 2022, pages 59:1–59:15. [41]

[Jea98] Peter Jeavons, On the algebraic structure of combinatorial problems, *Theoretical Computer Science*, vol. 200, no. 1, pages 185–204, 1998. [24]

[Kar72] Richard M. Karp, Reducibility among combinatorial problems, in *Complexity of Computer Computations*, ser. The IBM Research Symposia Series, Springer, 1972, pages 85–103. [10, 22, 27]

[KOWŽ23] Andrei Krokhin, Jakub Opršal, Marcin Wrochna, and Stanislav Živný, Topology and adjunction in promise constraint satisfaction, *SIAM Journal on Computing*, vol. 52, no. 1, pages 38–79, 2023. [34]

[Kra19] Jan Krajíček, *Proof Complexity* (Encyclopedia of Mathematics and Its Applications). Cambridge University Press, Mar. 2019, vol. 170. [11, 13, 19]

[Kra95] Jan Krajíček, *Bounded Arithmetic, Propositional Logic and Complexity Theory* (Encyclopedia of Mathematics and Its Applications). Cambridge University Press, Nov. 1995, vol. 60. [13]

[Kra97] Jan Krajíček, Interpolation theorems, lower bounds for proof systems, and independence results for bounded arithmetic, *Journal of Symbolic Logic*, vol. 62, no. 2, pages 457–486, Jun. 1997. [42]

[Kub04] Marek Kubale, Ed., *Graph Colorings* (Contemporary Mathematics). American Mathematical Society, Providence, RI, 2004, vol. 352, pages xii+208. [27]

[Lad75] Richard E. Ladner, On the structure of polynomial time reducibility, *Journal of the ACM*, vol. 22, no. 1, pages 155–171, Jan. 1975. [33]

[Lar25] Alberto Larrauri, Ineffectiveness for search and undecidability of PCSP meta-problems, in *Proceedings of the 66th Annual IEEE Symposium on Foundations of Computer Science (FOCS '25)*, Dec. 2025, pages 1313–1350. [24]

[Las01] Jean B Lasserre, An explicit exact sdp relaxation for nonlinear 0-1 programs, in *International Conference on Integer Programming and Combinatorial Optimization*, Springer, 2001, pages 293–303. [28]

[Lau18] Massimo Lauria, Algorithm analysis through proof complexity, in *Proceedings of the 14th Conference on Computability in Europe (CiE '18)*, ser. Lecture Notes in Computer Science, vol. 10936, Jul. 2018, pages 254–263. [29, 40]

[Lev73] Leonid A. Levin, Universal sequential search problems, *Problemy peredachi informatsii*, vol. 9, no. 3, pages 115–116, 1973, In Russian. Available at `http://mi.mathnet.ru/ppi914`. [10]

[Lew21] Rhydian M. R. Lewis, *Guide to graph colouring—algorithms and applications* (Texts in Computer Science), Second. Springer, Cham, 2021, pages xiv+303. [28]

[LN17] Massimo Lauria and Jakob Nordström, Graph colouring is hard for algorithms based on Hilbert's Nullstellensatz and Gröbner bases, in *Proceedings of the 32nd Annual Computational Complexity Conference (CCC '17)*, ser. Leibniz International Proceedings in Informatics (LIPIcs), vol. 79, Jul. 2017, pages 2:1–2:20. [28, 29, 32, 40, 42]

[LN23] Massimo Lauria and Jakob Nordström, *Graph colouring is hard for algorithms based on Hilbert's Nullstellensatz and Gröbner bases*, arXiv:2306.00125, 2023. [42]

[Lov82] László Lovász, Bounding the independence number of a graph, in *North-Holland Mathematics Studies*, vol. 66, Elsevier, 1982, pages 213–223. [15]

[LP25] Moritz Lichter and Benedikt Pago, Limitations of affine integer relaxations for solving constraint satisfaction problems, in *Proceedings of the 52nd International Colloquium on Automata, Languages, and Programming (ICALP '25)*, ser. Leibniz International Proceedings in Informatics (LIPIcs), vol. 334, 2025, pages 166:1–166:17. [34]

[Mac77] Alan K. Mackworth, Consistency in networks of relations, *Artificial Intelligence*, vol. 8, no. 1, pages 99–118, 1977. [33]

[Mit02a] David G. Mitchell, Resolution complexity of random constraints, in *Principles and Practice of Constraint Programming (CP '02)*, Springer Berlin Heidelberg, 2002, pages 295–310. [28]

[Mit02b] David G. Mitchell, The resolution complexity of constraint satisfaction, Ph.D. dissertation, University of Toronto, 2002. [28]

[MN15] Mladen Mikša and Jakob Nordström, A generalized method for proving polynomial calculus degree lower bounds, in *Proceedings of the 30th Annual Computational Complexity Conference (CCC '15)*, ser. Leibniz International Proceedings in Informatics (LIPIcs), vol. 33, Jun. 2015, pages 467–487. [29]

[MN24] Mladen Mikša and Jakob Nordström, A generalized method for proving polynomial calculus degree lower bounds, *Journal of the ACM*, vol. 71, no. 6, Nov. 2024, Preliminary version in *CCC '15*. [5, 6, 18, 21, 28, 31, 32, 49]

[Mon74] Ugo Montanari, Networks of constraints: Fundamental properties and applications to picture processing, *Information Sciences*, vol. 7, pages 95–132, 1974. [33]

[Mou24] Sasank Mouli, Polynomial calculus sizes over the Boolean and Fourier bases are incomparable, in *Proceedings of the 65th Annual IEEE Symposium on Foundations of Computer Science (FOCS '24)*, Oct. 2024, pages 790–796. [40, 41]

[MS07] Michael Molloy and Mohammad R. Salavatipour, The resolution complexity of random constraint satisfaction problems, *SIAM Journal on Computing*, vol. 37, no. 3, pages 895–922, 2007, Preliminary version in *FOCS'03*. [28, 47]

[MSL92] David Mitchell, Bart Selman, and Hector Levesque, Hard and easy distributions of SAT problems, in *Proceedings of the Tenth National Conference on Artificial Intelligence (AAAI '92)*, AAAI Press, 1992, pages 459–465. [28]

[Nes00] Yurii Nesterov, Squared functional systems and optimization problems, in *High Performance Optimization*. Springer US, 2000, pages 405–440. [28]

[Ó C22] Adam Ó Conghaile, Cohomology in constraint satisfaction and structure isomorphism, in *47th International Symposium on Mathematical Foundations*

of Computer Science (MFCS '22), ser. Leibniz International Proceedings in Informatics (LIPIcs), vol. 241, 2022, pages 75:1–75:16. [33, 34]

[Pap24] Theodoros Papamakarios, Depth-d Frege systems are not automatable unless P = NP, in *Proceedings of the 39th Computational Complexity Conference (CCC '24)*, ser. Leibniz International Proceedings in Informatics (LIPIcs), vol. 300, Jul. 2024, pages 22:1–22:17. [41]

[Par00] Pablo A. Parrilo, Structured semidefinite programs and semialgebraic geometry methods in robustness and optimization, Available at `http://resolver.caltech.edu/CaltechETD:etd-05062004-055516`, Ph.D. dissertation, California Institute of Technology, May 2000. [28]

[Pud03] Pavel Pudlák, On reducibility and symmetry of disjoint NP pairs, *Theoretical Computer Science*, vol. 295, no. 1, pages 323–339, Feb. 2003, Special issue for *MFCS '01*. [45]

[Pud97] Pavel Pudlák, Lower bounds for resolution and cutting plane proofs and monotone computations, *Journal of Symbolic Logic*, vol. 62, no. 3, pages 981–998, Sep. 1997. [42]

[Raz03] Alexander A. Razborov, Resolution lower bounds for the weak functional pigeonhole principle, *Theoretical Computer Science*, vol. 1, no. 303, pages 233–243, Jun. 2003. [20]

[Raz04a] Ran Raz, Resolution lower bounds for the weak pigeonhole principle, *Journal of the ACM*, vol. 51, no. 2, pages 115–138, Mar. 2004, Preliminary version in *STOC '02*. [20]

[Raz04b] Alexander A. Razborov, Resolution lower bounds for perfect matching principles, *Journal of Computer and System Sciences*, vol. 69, no. 1, pages 3–27, Aug. 2004, Preliminary version in *CCC '02*. [20]

[Raz26] Alexander A. Razborov, *Open problems*, `https://people.cs.uchicago.edu/~razborov/teaching/index.html`. Accessed on April 8th, 2026, 2026. [21, 48]

[Raz98] Alexander A. Razborov, Lower bounds for the polynomial calculus, *Computational Complexity*, vol. 7, no. 4, pages 291–324, Dec. 1998. [20, 29–31]

[Rec75] Robert A. Reckhow, On the lengths of proofs in the propositional calculus, Available at `https://www.cs.toronto.edu/~sacook/homepage/reckhow_thesis.pdf`, Ph.D. dissertation, University of Toronto, 1975. [11, 28]

[Rii93] Søren Riis, Independence in bounded arithmetic, Ph.D. dissertation, University of Oxford, 1993. [21]

[Rob65] John Alan Robinson, A machine-oriented logic based on the resolution principle, *Journal of the ACM*, vol. 12, no. 1, pages 23–41, Jan. 1965. [14]

[RT08] Ran Raz and Iddo Tzameret, The strength of multilinear proofs, *Computational Complexity*, vol. 17, no. 3, pages 407–457, 2008. [42]

[RT24] Julian Romero and Levent Tunçel, Graphs with large girth and chromatic number are hard for Nullstellensatz, *SIAM Journal on Discrete Mathematics*, vol. 38, no. 3, pages 2108–2131, 2024. [29, 32, 44]

[SA90] Hanif D. Sherali and Warren P. Adams, A hierarchy of relaxations between the continuous and convex hull representations for zero-one programming problems, *SIAM Journal on Discrete Mathematics*, vol. 3, pages 411–430, 1990. [28]

[San21] Rahul Santhanam, *On the intractability of propositional proof complexity*, Presentation at the workshop *Reflections on Propositional Proofs in Algorithms*

and Complexity held online at *FOCS '21*. Video available at `http://youtu.be/d4d-yeJXtTQ` (retrieved May 26th, 2026), Feb. 2021. [13]

[Sch78] Thomas J. Schaefer, The complexity of satisfiability problems, in *Proceedings of the Tenth Annual ACM Symposium on Theory of Computing (STOC 78)*, 1978, pages 216–226. [33]

[SH23] Bernardo Subercaseaux and Marijn J. H. Heule, The packing chromatic number of the infinite square grid is 15, in *29th International Conference on Tools and Algorithms for the Construction and Analysis of Systems (TACAS '23)*, ser. Lecture Notes in Computer Science, vol. 13993, Cham: Springer Nature Switzerland, 2023, pages 389–406. [39]

[Sho87] Naum Z. Shor, An approach to obtaining global extremums in polynomial mathematical programming problems, *Cybernetics and Systems Analysis*, vol. 23, pages 695–700, Sep. 1987. [28]

[Sok20] Dmitry Sokolov, (Semi)Algebraic proofs over {±1} variables, in *Proceedings of the 52nd Annual ACM SIGACT Symposium on Theory of Computing (STOC '20)*, Association for Computing Machinery, 2020, pages 78–90. [21, 38–45, 49]

[Sok24] Dmitry Sokolov, Random ($\log n$)-CNF are hard for cutting planes (again), in *Proceedings of the 56th Annual ACM Symposium on Theory of Computing (STOC '24)*, Jun. 2024, pages 2008–2015. [42]

[Thu94] William P. Thurston, On proof and progress in mathematics, *Bulletin of the American Mathematical Society*, vol. 30, no. 2, pages 161–177, Apr. 1994, Available at `https://www.math.toronto.edu/mccann/199/thurston.pdf`. [1]

[Tse68] Grigori Tseitin, The complexity of a deduction in the propositional predicate calculus, *Zapiski Nauchnyh Seminarov Leningradskogo Otdelenija matematicheskogo Instituta im. V. A. Steklova akademii Nauk SSSR (LOMI)*, vol. 8, pages 234–259, 1968, In Russian. [22]

[Tur36] Alan M. Turing, On computable numbers, with an application to the Entscheidungsproblem, *Proceedings of the London Mathematical Society. Second Series*, vol. 42, no. 3, pages 230–265, 1936. [9]

[Urq87] Alasdair Urquhart, Hard examples for resolution, *Journal of the ACM*, vol. 34, no. 1, pages 209–219, Jan. 1987. [22, 39, 41, 43]

[Wal72] David L. Waltz, Generating semantic descriptions from drawings of scenes with shadows, Ph.D. dissertation, Massachusetts Institute of Technology, 1972. [33]

[Wig19] Avi Wigderson, *Mathematics and Computation: A Theory Revolutionizing Technology and Science*. Princeton University Press, Princeton, NJ, 2019, pages xiii+418. [1]

[Zhu20] Dmitriy Zhuk, A proof of the CSP dichotomy conjecture, *Journal of the ACM*, vol. 67, no. 5, Aug. 2020, Preliminary version in *FOCS '17*. [33, 34, 48]