KTH ROYAL INSTITUTE
OF TECHNOLOGY

Doctoral Thesis in Computer Science

On Long Proofs of Simple Truths

KILIAN RISSE

Stockholm, Sweden 2022

On Long Proofs of Simple Truths

KILIAN RISSE

Academic Dissertation which, with due permission of the KTH Royal Institute of Technology, is submitted for public defence for the Degree of Doctor of Philosophy on Thursday the 27th of October 2022, at 2:00 p.m. in F3, Lindstedtsvägen 26, Stockholm

Doctoral Thesis in Computer Science
KTH Royal Institute of Technology
Stockholm, Sweden 2022

ISBN 978-91-8040-354-2
TRITA-EECS-AVL-2022:55

Printed by: Universitetsservice US-AB, Sweden 2022

Abstract

Propositional proof complexity is the study of certificates of infeasibility. In this thesis we consider several proof systems with limited deductive ability and unconditionally show that they require long refutations of the feasibility of certain Boolean formulas.

We show that the depth d Frege proof system, restricted to line size M, requires proofs of length at least $\exp\bigl(n/(\log M)^{O(d)}\bigr)$ to refute the Tseitin contradiction defined over the $n \times n$ grid graph, improving upon the recent result of Pitassi et al. [PRT22]. Along the way we also sharpen the lower bound of Håstad [Hås20] on the depth d Frege refutation size for the same formula from exponential in $\tilde{\Omega}(n^{1/59d})$ to exponential in $\tilde{\Omega}(n^{1/(2d-1)})$.

We also consider the perfect matching formula defined over a sparse random graph on an odd number of vertices n. We show that polynomial calculus over fields of characteristic $\neq 2$ and sum of squares require size exponential in $\Omega(n/\log^2 n)$ to refute said formula. For depth d Frege we show that there is a constant $\delta > 0$ such that refutations of these formulas require size $\exp\bigl(\Omega(n^{\delta/d})\bigr)$.

The perfect matching formula has a close sibling over bipartite graphs: the graph pigeonhole principle. There are two methods to prove resolution refutation size lower bounds for the pigeonhole principle. On the one hand there is the general width-size tradeoff by Ben-Sasson and Wigderson [BW01] which can be used to show resolution refutation size lower bounds in the setting where we have a sparse bipartite graph with n holes and $m \ll n^2$ pigeons. On the other hand there is the pseudo-width technique developed by Razborov [Raz04] that applies for any number of pigeons, but requires the graph to be somewhat dense. We extend the latter technique to also cover the previous setting and more: for example, it has been open whether the functional pigeonhole principle defined over a random bipartite graph of bounded degree and $\mathrm{poly}(n) \geq n^2$ pigeons requires super-polynomial size resolution refutations. We answer this and related questions.

Finally we also study the circuit tautology which claims that a Boolean function has a circuit of size s computing it. For $s = \mathrm{poly}(n)$ we prove an essentially optimal Sum of Squares degree lower bound of $\Omega(s^{1-\varepsilon})$ to refute this claim for any Boolean function. Further, we show that for any $0 < \alpha < 1$ there are Boolean functions on n bits with circuit complexity larger than 2^{n^α} but the Sum of Squares proof system requires size $2^{\left(2^{\Omega(n^\alpha)}\right)}$ to prove this. Lastly we show that these lower bounds can also be extended to the monotone setting.

Sammanfattning

Propositionell beviskomplexitet är studerar certifikat av icke-satisfierbarhet. Vi betraktar bevissystem med begränsad deduktiv förmåga och bevisar ovillkorliga undre gränser för längden på vederläggningar av formlers satisferbarhet. Denna avhandling bevisar flera nya sådana undre gränser för bevissystemen resolution, polynomialkalkyl, kvadratsummor, och Frege-system av begränsat djup.

Vi visar att Frege-systemet av djup d, begränsat till rader av storlek M, kräver minst bevis av längd minst $\exp\bigl(n/(\log M)^{O(d)}\bigr)$ för att motbevisa Tseitin-kontradiktionen definierad över $n \times n$-rutnätet, vilket förbättrar ett nyligen visat resultat av Pitassi et al. [PRT22]. Längs vägen skärper vi även Håstads undre gräns [Hås20] för längd för Frege av djup d för samma formel från exponentiell i $\tilde{\Omega}(n^{1/59d})$ till exponentiell i $\tilde{\Omega}(n^{1/(2d-1)})$.

Vi betraktar också formeln för perfekt matchning över en gles slumpgraf med ett udda antal hörn n. Vi visar att polynomkalkyl över kroppar med karaktäristik $\neq 2$ och kvadratsummor kräver längd exponentiell i $\Omega(n/\log^2 n)$ för att motbevisa denna formel. För Frege av djup d visar vi att det finns en konstant $\delta > 0$ så att vederläggningar av dessa formler kräver storlek $\exp\bigl(\Omega(n^{\delta/d})\bigr)$.

Formeln för perfekt matchning har ett nära syskon över bipartita grafer: duvslagsprincipen över grafer. Det finns två metoder för att visa undre gränser för refutations för duvslagsprincipen. Å ena sidan finns Ben-Sasson och Wigdersons [BW01] generella avvägning mellan bredd och storlek som kan användas för att visa undre gränser för resolution i fallet där vi har en gles bipartit graf med n hål och $m \ll n^2$ duvor. Å andra sidan finns pseudo-bredd-tekniken utvecklad av Razborov [Raz04] som kan appliceras för valfritt antal duvor, men kräver att grafen är någorlunda tät. Vi utökar den senare tekniken till att även täcka det förstnämnda fallet och mer: till exempel har det varit öppet om den funktionella duvslagssprincipen definierad över en slumpmässig bipartit graf med begränsade gradtal och $\mathrm{poly}(n) \geq n^2$ duvor kräver motbevis av superpolynomisk storlek. Vi besvarar detta och relaterade frågor.

Slutligen studerar vi också kretstautologin som hävdar att en Boolean funktion har en krets av storlek s som beräknar den. Vi bevisar en väsentligen optimal undre gräns för gradtal för kvadratsummor på $\Omega(s^{1-\varepsilon})$ för att motbevisa detta påstående för varje Boolesk funktion, för $s > \mathrm{poly}(n)$. Vidare visar vi att det för alla $0 < \alpha < 1$ finns Booleska funktioner på n bitar med kretskomplexitet större än $2^{n^{\alpha}}$ men där kvadratsummor kräver storlek $2^{\left(2^{\Omega(n^{\alpha})}\right)}$ för att bevisa detta.

Acknowledgments

Foremost I want to thank all my advisors, in alphabetical order, Per Austrin, Johan Håstad and Jakob Nordström. They always stood by my side, never doubted me and seemingly had infinite patience to repeatedly explain the same concept to me. Without them this thesis would not exist.

Jakob, thank you for introducing me to proof complexity and for sharing your enthusiasm for this subject with me. I have fond memories of the trips to Lund and all the research discussions with you. Johan, thank you for always having an open door, patiently listening to me and all the thoughts you have shared with me. Per, I have learned a lot from you – about research but also about movies. Thank you for your good spirit and all the long afternoons discussing research.

In addition to my advisors I also have the honor to call Susanna de Rezende and Dmitry Sokolov my co-authors. It was, and is, a lot of fun to solve problems with you and I am looking forward to our continued collaboration. During my stay at KTH I further had the pleasure to collaborate with Aaron, Sagnik, Jonah, Joseph and Aleksa. Thank you for the time in front of a whiteboard with you.

Naemi, Mattias, Chris and Chris, Martin, Tom, Linda, Irena and Solomon, you guys made my stay in Sweden worthwhile – thank you! Special thanks to Naemi and Solomon who dared to share an apartment with me. Back in Switzerland I want to thank Bernhard, Adriano, Matthias, Simon, Daniel, Raphael, Leonie, Linh-Thu and Franziska for staying in touch with me even though I was mostly absent the past years. Kay, Florine I am so grateful to call you my friends. Thank you for always having an open door in Copenhagen.

My family has supported me throughout the last years despite fully understanding what I do. Thank you Mom and Dad for your generosity and patience, Anna, Pascal and Jannis for always having an empty bed for me, and Leon, Chanel, Christine, Dominique and Urs for all the discussions and the time that you chose to spend with me.

Last but not least, Patrycja, thank you for making my dark days brighter, for enduring my absent-mindedness and simply for being you.

Contents

Part I

Thesis

Introduction

This chapter is a non-technical introduction to the thesis intended for readers with little or no mathematical background. All notions informally introduced in this chapter are revisited in Chapter 2.

1.1 Proofs

This thesis is about proofs and contains proofs proving properties of proofs. But before talking more about proofs (and proofs about proofs) it might be worthwhile to take a step back and think about what a proof really is – after all it does seem to be a central concept of this thesis.

Broadly speaking, a proof is an object that, hopefully, convinces others of some claim. Depending on the situation a proof may have very different form: sometimes a paper from an authority may be sufficient, while in other situations, like in court, a proof has to be convincing enough so that the judge believes it "beyond reasonable doubt". And even in mathematics itself there are several notions of a proof. For example a proof published in an article intended to convince other mathematicians of the validity of a theorem is usually not very formal. Well-known steps may be skipped, some statements may be left to prove by the reader and sometimes there are even oversights by the authors. As such it is not surprising that, once in a while, mathematicians make mistakes and therefore have to retract articles.

Mathematicians could prevent this from happening – since the late 19th century we know of proof systems that produce machine verifiable proofs. These proofs are great to verify statements and make sure that they are indeed correct. However, fully formalizing a proof is very time consuming, they are usually not human readable and cannot convey the intuition of a proof. As such these formal proofs are an interesting object to study but not very useful to communicate proofs to other human beings.

The first formal proof system has been put forward by Frege [Fre79] in 1879 and these systems have later been popularized in the 1920s by Hilbert when he proposed his program to base mathematics on a solid foundation. Hilbert's ultimate goal was to base mathematics on a logic basis: he wanted to reprove all mathematical statements from a finite number of assumptions (also known as *axioms*) such that (i) these assumptions do not contradict each other, and (ii) all true statements can be proven from these assumptions. This is a seemingly desirable goal if one believes that mathematics describes universal truth.

However, this turns out to be impossible: in 1931 Gödel [Göd31] proved that strong enough proof systems, as studied by Hilbert and others, are either self-contradicting or they cannot prove the statement that the system itself can prove all correct statements. In other words, we cannot hope for a single set of non-contradicting axioms $\mathcal{A}$ that can prove the statement "all true consequences can be derived from $\mathcal{A}$". This theorem, known as Gödel's incompleteness theorem, left Hilbert's original program in shambles.

On the upside, this led other mathematicians to study related problems and ask more refined questions about the existence of proofs. We just learned that we cannot prove everything. Can we at least determine what statements can be proven? That is, given some statement P, can we decide whether the claim P can be proven from our favorite set of axioms? Even this turns out to be too much to ask for. But before discussing this we should take a small detour explaining what we mean by "we can tell" that a statement has a proof – what kind of machine are we allowed to use to determine whether there is a proof of a statement? Mathematicians put forward different formalisms in the 30s of the last century, one of them being the Turing machine. This formal machine, proposed by Turing, intends to be able to perform any task that could also be done "by pen and paper", or, in more modern terms, by an algorithm. In 1936 Turing [Tur37], and independently Church [Chu36] using a different albeit equivalent formalism, showed that no algorithm running in finite time can determine whether a statement can be proven from a given set of axioms.

To summarize, there are correct statements with no proof and, even worse, we cannot tell whether a given statement has a proof. Now, I cannot blame you if your mind has started to wander and you find this a quite lofty and dry discussion. So let me give you a very concrete real-life consequence. We all use computers, smartphones or even smartwatches in our daily lives. Would it not be great if we could write an algorithm that ensures that all these devices never crash? No need to ever reboot your computer because — reasons? Well, you may have guessed, this is unfortunately one of these so-called undecidable statements. Bear this in mind the next time you call your computer person – they are trying hard to solve an undecidable problem. Have some patience, get a coffee, and, in the mean time, reboot

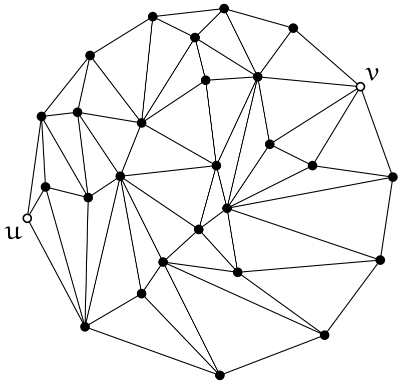


Figure 1.1: A graph with two labeled nodes u and v

your computer.

But joking aside, it really seems like we have to give up on these undecidable statements. Maybe there is more to discover in the set of decidable statements? For example, do such statements always have small proofs? Can small proofs be found efficiently? These, and related questions, are the foundations of theoretical computer science.

1.2 Efficient Proofs

From now on we only study claims that are decidable, i.e., have a proof, though it may be long. Let us start with a simple example of the kind of statements that we want to consider. Suppose we are given a graph with two labeled nodes u, v and the claim that these two nodes are connected by a path of length 5. An illustration of an example graph can be found in Figure 1.1.

Is this claim correct? Staring at the graph for a bit one can actually find a path of length 5, as illustrated in Figure 1.2 on the following page. This illustration is a pretty convincing proof – that was not so hard to prove. What about the claim that there is no path of length at most 4 connecting u to v? Can we convince ourselves that this holds? We could try to enumerate all paths between u and v and check that there is no shorter path. This would indeed give us a valid proof, but it seems cumbersome as there are so many paths connecting u to v.

Let us try to create a more concise proof, as follows. Find all nodes at distance at most one from u. This is not difficult: this is u and all neighbors $N(u)$ of u. Once we have found the nodes V_1 at distance at most 1 from u,

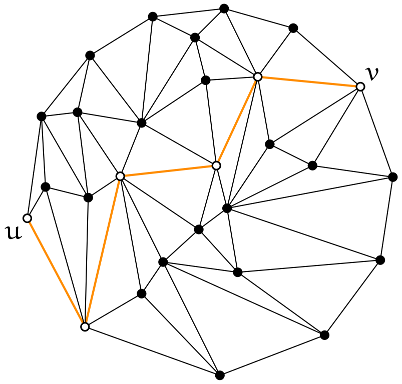


Figure 1.2: A path of length 5

we can now find all nodes at distance at most two from u: these are the nodes of V_1 as well as the neighborhood $N(V_1)$ of it. Iterating this idea, we see that v is not in the set of nodes V_4 at distance at most 4 from u. Thus we proved the claim. This proof is illustrated in Figure 1.3 on the following page.

Observe that this argument gives us an algorithm to determine the length of the shortest path between u and v: simply run the described procedure until v is, for the first time, in the set V_i and report that the shortest path between u and v is of distance i. Claims that have an efficient algorithm, as the one just described, make up the class P. That is, P consists of all claims that can be efficiently proven *and*, furthermore, this short proof can also be found efficiently.

Let us consider a claim that we suspect is of a different nature: the claim that a graph contains a cycle visiting every node exactly once (this is known as a Hamiltonian cycle). If you have the stamina, you can stare at the graph in Figure 1.1 for a while and you will notice that there is indeed a Hamiltonian cycle – one example is highlighted in Figure 1.4 on the following page. Again, this illustration is a short, efficient proof similar to the proof of the existence of a path of length 5 in Figure 1.2. So far there is no evidence that would justify calling this claim of a different nature.[1] It gets more interesting if we consider the negation of the previous claim, namely the claim that "there is no Hamiltonian cycle". Suppose our graph does not contain a Hamiltonian cycle. How would we prove this? Of course there is the brute-force proof: enumerate each ordering of nodes

[1] Here we ignore the question whether such proofs can be found efficiently. We discuss this question in detail later on.

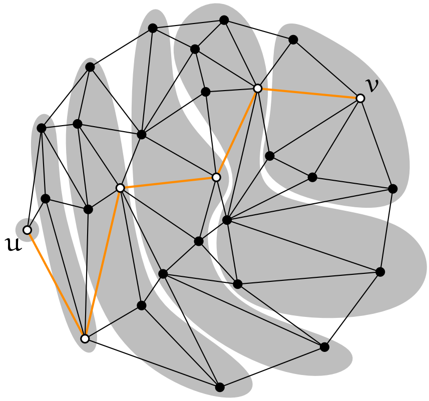


Figure 1.3: The nodes at distance $0, 1, 2, 3, 4$ and 5 from u

and rule out that this ordering gives rise to a Hamiltonian cycle. If we consider a graph on 28 nodes, as in Figure 1.1, then this brute-force proof would consist of about $28! \approx 10^{29}$ many orderings. To get a feeling for this number, suppose that we are really quick at checking orderings, say, we can check a million orderings a second. In this case it would take us a mere 10^{15} years to check all orderings – as the universe is only about 10^{10} years old this is not an awfully useful proof.

If the naïve proof of such a small graph is already this long, we have to investigate whether there are more efficient proofs of this claim. So far we have not succeeded in this endeavor and it is generally believed that there are no proofs significantly shorter than our naïve proof.[2] And this is by far the only statement for which we do not have short proofs – there is an entire cluster of statements that seem impossible to prove efficiently. In order to discuss this in the language of theoretical computer science we need to introduce two classes of statements.

The first class is called NP and consists of all statements that have an efficient proof, e.g., "there is a Hamiltonian cycle" or "there is no path of length 4". The second class is coNP which consists of all statements whose negation has an efficient proof: for example, "there is *no* Hamiltonian cycle" or "there is no path of length 4". Already from the given examples we see that NP and coNP have certain statements in common. The big open question is whether *all* claims of coNP are also in NP, or in words, whether the negation of efficiently verifiable statements also have short proofs. It is widely believed that this is not the case: it is believed that there are claims in

[2] It should be mentioned that there are *slightly* more efficient proofs. If a graph has n nodes, then proofs can be made of length roughly 2^n instead of the $n! \approx 2^{n \log n}$ length our naïve proof has.

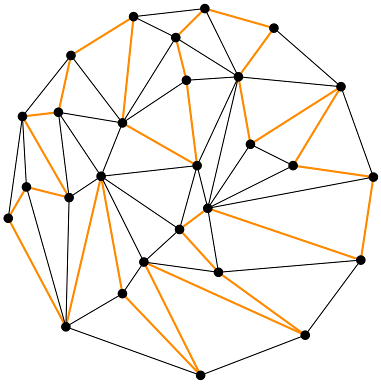


Figure 1.4: A Hamiltonian cycle

coNP, for example, "there is no Hamiltonian cycle", that inherently require long proofs.

This thesis continues a line of work that eventually hopes to separate NP from coNP. The program, put forward by Cook and Reckhow [CR79], suggests to study increasingly more powerful proof systems and prove that these systems require long proofs for some claims in coNP. Starting with "weak" proof systems, i.e., proof systems of limited deductive ability, we hope to develop a toolbox of lower bound techniques that can be applied to increasingly stronger proof systems. Apart from its intrinsic interest, separating NP from coNP would have some interesting consequences. Let us explain.

By definition we have that P is contained in NP, as all claims in P have short proofs. Also, it is not so hard to convince yourself that a claim C is in P if and only if the claim "not C" is also in P: as in the path example, we can run the efficient algorithm that finds a proof of C. By definition, this algorithm is complete, meaning that it always outputs a proof if one exists. Thus running the efficient algorithm and obtaining no proof from it is itself a short proof of the claim "not C". As such we conclude that P lies in the intersection of coNP and NP. Equivalently we can write this in symbols as $\mathsf{P} \subseteq \mathsf{NP} \cap \mathsf{coNP}$.

Now suppose that we can separate NP from coNP. We claim that this implies that P is a strict subset of NP: because P is closed under complementation but NP is not, there has to be some statement that is in NP but not in P. To determine whether $\mathsf{P} \overset{?}{=} \mathsf{NP}$, or equivalently whether a claim with a short proof also has an efficient algorithm that recovers the short proof, is one of the seven millennium problems put forward in 2000 by the Clay Mathematics Institute [CJW06].

As previously mentioned, this thesis continues the program of Cook and Reckhow with the eventual goal to separate NP from coNP. We prove several new proof size lower bounds for different formulas and proof systems. We achieve this by adapting and extending known lower bound techniques, thereby expanding the current toolbox of lower bound techniques for proof complexity. In Chapter 2 we cover the necessary background to then discuss our contributions in Chapter 3. No formal proofs are covered in the first part of the thesis. The proofs of the mentioned theorems can be found in Part II which contains all the discussed papers.

Background

In the first section of this chapter we revisit some well-known notions from complexity theory as well as some basic graph theory. In Section 2.2 we introduce the relevant proof systems and in Section 2.3 the propositional formulas that we intend to study.

2.1 Preliminaries

For an integer $n \in \mathbb{N}$ we let $[n] = \{1, \dots, n\}$ denote the set of integers from 1 through n, and we let $\log n$ denote the logarithm of n to the base 2.

Let $f, g : \mathbb{R} \to \mathbb{R}$ be two functions. We write $f(n) \in O\big(g(n)\big)$, respectively $f(n) \in \Omega\big(g(n)\big)$, if there is a constant c and an n_0 such that for all $n \geq n_0$ it holds that $f(n) \leq c \cdot g(n)$, respectively $f(n) \geq c \cdot g(n)$. We say that f is *poly-bounded* if there is a constant c such that $f \in O(n^c)$. We further introduce the notation $f(n) \in o\big(g(n)\big)$ to mean that for all constants $c > 0$ there is an n_0 such that $f(n) \leq c \cdot g(n)$, for all $n \geq n_0$.

Similarly we sometimes want to supress dependencies on constants and write $f(n, \varepsilon) \in O_\varepsilon\big(g(n, \varepsilon)\big)$, respectively $f(n, \varepsilon) \in \Omega_\varepsilon\big(g(n, \varepsilon)\big)$, to mean that there exists a function $c(\varepsilon) > 0$ and a constant n_0 such that for all $n \geq n_0$ it holds that $f(n, \varepsilon) \leq c(\varepsilon) \cdot g(n, \varepsilon)$, respectively $f(n, \varepsilon) \geq c(\varepsilon) \cdot g(n, \varepsilon)$. In a similar vein we sometimes even want to suppress logarithmic dependencies: we write $f(n) \in \tilde{O}\big(g(n)\big)$ to mean that there are constants c and n_0 such that for all $n \geq n_0$ it holds that $f(n) \leq \log^c(n) \cdot g(n)$, and $f(n) \in \tilde{\Omega}\big(g(n)\big)$ if there are constants c and n_0 such that $f(n) \geq \log^c(n) \cdot g(n)$, for all $n \geq n_0$.

2.1.1 Graph Theory

Let us recall some standard graph terminology. A *graph* $G = (V, E)$ is a 2-tuple that consists of a set of vertices $V = V(G)$ and a set of edges $E = E(G) \subseteq V \times V$, and a *bipartite graph* $G = (U, V, E)$ is a 3-tuple that consists of two disjoint vertex sets U and V, with $V(G) = U \cup V$, and an edge set $E = E(G) \subseteq (U \times V) \cup (V \times U)$.

We only consider simple and undirected graphs; all graphs contain no self-loops, i.e., edges (u, u), and if $(u, v) \in E$, then also $(v, u) \in E$ and we may thus think of the edge set E as containing sets of size 2.

The neighborhood of a vertex $u \in V(G)$ is $N(u) = \{v \mid \{u, v\} \in E\}$, the neighborhood of a set of vertices $U \subseteq V(G)$ is $N(U) = \bigcup_{u \in U} N(u)$ and for sets $U, W \subseteq V(G)$ the neighborhood of U in W is $N(U, W) = N(U) \cap W$. We denote by $\deg(v) = |N(v)|$ the degree of a vertex $v \in V$, by $\Delta(G)$ the maximum degree, $\delta(G)$ the minimum degree and by $d(G)$ the average degree of G. For a set of vertices $U \subseteq V(G)$ we let $G[U]$ denote the graph induced by U, i.e., $G[U] = (U, E_U)$, where $E_u = \{e \in E(G) \mid e \subseteq U\}$.

A graph $G = (V, E)$ on n vertices is an *α-expander* (has *vertex expansion* α) if for all sets $U \subseteq V$ of size $|U| \leq n/2$ it holds that $|N(U, V \setminus U)| \geq \alpha|U|$.

The *grid graph* (more commonly *torus*) $G = (V, E)$ of dimension n consists of vertices $V = \{(i, j) \mid 0 \leq i, j < n\}$ and the vertex (i, j) is connected by edges to the four neighbors at distance 1, i.e., where one coordinate is identical and the other changes by ± 1 modulo n.

We denote the *uniform distribution over* d*-regular graphs on* n *vertices* by $\mathcal{G}(n, d)$ and tacitly assume that nd is even. A graph G contains H as a *topological minor* if there is an injective map $\sigma : V(H) \to V(G)$ and for every $\{u, v\} \in E(H)$ there is a path $p_{uv} \subseteq G$ from $\sigma(u)$ to $\sigma(v)$ that is pairwise vertex-disjoint from all other paths except in the endpoints. The paths p_{uv} are the *edge embeddings* of the minor.

2.1.2 Languages, Formulas and Circuits

An alphabet Σ is a finite set of symbols and we let Σ^* be the set of finite length strings that can be formed over Σ. For a string $w \in \Sigma^*$ we let $|w|$ denote the length of w. A *language* $L \subseteq \Sigma^*$ is a set of finite length strings over the alphabet Σ. We may usually assume that the alphabet is $\{0, 1\}$ but it is often convenient to work with larger alphabets.

Let us say that the *depth* of a string is the number of changes of symbols in a string: the depth is 0 if the string s is empty, and otherwise, if $s = (s_1, \ldots, s_k)$, then the depth is defined to be $1 + \mathrm{Depth}(s_i, \ldots, s_k)$, where i is the largest integer such that $s_{i-1} = s_1$.

Formulas The language of *DeMorgan formulas* consists of all strings defined recursively over the alphabet consisting of variables $x_1, \ldots, x_n$, the connectives $\vee, \wedge, \neg$, the symbols $\top, \bot$, and the brackets $(,)$ as follows.

1. The symbols $\top$ and $\bot$ denote formulas,

2. any variable x_i is a formula,

3. if F is a formula, then so is $\neg F$,

4. if F_1 and F_2 are formulas, then so are $(F_1 \wedge F_2)$ and $(F_1 \vee F_2)$.

We use $\bar{x}_i$ as a shorthand for the formula $\neg x_i$ and call the formulas x_i and $\bar{x}_i$ *literals*. A *subformula* of a formula F is a substring of F that is also a formula and the *size* of a formula is the length of the string representing it.

We can think of a formula F as a binary tree T_F where each internal node is either a node with 2 children and labeled with an $\vee$ or an $\wedge$ or a node with 1 child labeled with $\neg$, and each leaf node is either labeled with a variable or one of the symbols $\top$ or $\bot$. For a branch b in T_F, let $L(b) = (\ell_1, \ldots, \ell_k)$ denote the string of internal labels encountered on b if traversed from root to leaf. We define the *depth* of the branch b as $\text{Depth}\big(L(b)\big)$, and the depth of a formula F as the maximum depth of any branch in T_F.

An *assignment* is a mapping $\alpha : \{x_1, \ldots, x_n\} \longrightarrow \{\textsc{True}, \textsc{False}\}$ that sets each variable to either $\textsc{True}$ or $\textsc{False}$. We often identify $\textsc{True}$ with 1 and $\textsc{False}$ with 0 and call a variable that may be $\textsc{True}$ or $\textsc{False}$ a *bit*. A formula F evaluates to $\textsc{True}$ under an assignment α if

1. F is of the form $\top$, or

2. F is of the form x_i and $\alpha(x_i) = \textsc{True}$, or

3. F is of the form $\neg G$ and G does *not* evaluate to $\textsc{True}$ under α, or

4. F is of the form $(G_1 \wedge G_2)$ and both G_1 and G_2 evaluate to $\textsc{True}$ under α, or

5. F is of the form $(G_1 \vee G_2)$ and at least one G_i evaluates to $\textsc{True}$ under α.

If F does not evaluate to $\textsc{True}$ under α, then we say that it evaluates to $\textsc{False}$ under α and write $F(\alpha)$, or $\alpha(F)$, to denote the (unique) value F evaluates to under α. When writing formulas we usually ignore most brackets. In particular we write $x_1 \vee x_2 \vee \cdots \vee x_n$ as a shorthand for $(x_1 \vee (x_2 \vee (\cdots (x_{n-1} \vee x_n) \cdots)$ and similarly for $\wedge$. A formula G is *logically implied* by the formulas $F_1, \ldots, F_k$, written as $F_1, \ldots, F_k \models G$, if for all assignments α such that $\alpha(F_1 \wedge \cdots \wedge F_k) = \textsc{True}$ it also holds that $G(\alpha) = \textsc{True}$.

A formula F is *satisfiable* if there is an assignment α such that $F(\alpha) = \textsc{True}$, a formula F is *unsatisfiable*, or a *contradiction*, if $F(\alpha) = \textsc{False}$ for all assignments α and similarly a formula F is a *tautology* if $F(\alpha) = \textsc{True}$ under all assignments. We denote the language that consists of all satisfiable formulas by SAT, the language of unsatisfiable formulas by UNSAT and the language of tautologies by TAUT.

A disjunction of literals, e.g., $C = \left(x_3 \vee \bar{x}_{42} \vee \cdots \vee \bar{x}_7\right)$, is called a *clause*, the *width* of C, denoted by $\mathrm{Width}(C)$, is equal to the number of literals in C, and a k-*clause* is any clause of width at most k. A formula F is in *conjunctive normal form (CNF)* if it is a conjunction of clauses, i.e., of the form $\bigwedge_{i=1}^{m} C_i$, where each C_i is a clause. The width of F, denoted by $\mathrm{Width}(F)$ is the maximum width of any clause occurring, the formula F is a k-*CNF* if each clause in F is a k-clause and F is of *bounded width* if there is a constant k such that F is a k-CNF.

A Boolean function $f : \{0,1\}^n \rightarrow \{0,1\}$ on n bits is represented by a formula F over n variables if for all assignments α it holds that $f(\alpha) = F(\alpha)$.

Proposition 2.1.1. *Every Boolean function* $f : \{0,1\}^n \rightarrow \{0,1\}$ *on* n *bits can be represented as a CNF of size* $O(n \cdot 2^n)$*.*

Proof. We construct the CNF F as follows. For each assignment α such that $f(\alpha) = \textsc{False}$, we add the negation, written as a clause, of the unique n variable conjunction that is satisfied by α to F. It is readily verified that $f(\alpha) = F(\alpha)$ for all assignements α. □

The language that consists of all satisfiable CNFs is denoted by CNF-SAT, and we let the set of unsatisfiable CNFs be denoted by CNF-UNSAT.

Circuits Note that when defining formulas, we may be forced to write down the exact same subformula several times. Circuits allow the re-use of such subformulas, thus decreasing the size of the representation.

We define the language of Boolean *circuits* over the DeMorgan basis as follows. A string C is a circuit if it is a sequence $(F_1, \ldots, F_s)$ of DeMorgan formulas over the variables $x_1, \ldots, x_n, y_1, \ldots, y_s$, where each F_i is of the following form:

1. one of the symbols $\top$ or $\bot$, or
2. x_j, for $j \in [n]$, or
3. $\neg y_j$, for $j < i$, or
4. $(y_j \circ y_k)$, for $j, k < i$ and $\circ \in \{\wedge, \vee\}$.

We determine whether a circuit $C = (F_1, \ldots, F_s)$ evaluates to $\textsc{True}$ under an assignment α (to the variables $x_1, \ldots, x_n$) as follows. Starting with $i = 1$ and $\alpha_0 = \alpha$, we sequentially evaluate the formula F_i under the assignment α_{i-1}, and extend α_{i-1} to the variable y_i to obtain α_i defined as

$$\alpha_i(z) = \begin{cases} \alpha_{i-1}(z), & \text{if } z \neq y_i, \\ \alpha_{i-1}(F_i), & \text{otherwise.} \end{cases}$$

We say that C evaluates to TRUE under α if $\alpha_s(y_s) = \text{TRUE}$ and C evaluates to FALSE otherwise. As for formulas we write $C(\alpha)$ to denote the unique value that C evaluates to under α. Finally, we say that a Boolean function f is *represented* by the circuit C, or *computed* by the circuit C, if $C(\alpha) = f(\alpha)$ for all assignments α.

Note that we can view a circuit C as a directed acyclic graph: for each $i \in [s]$ we have a node that is connected by an incoming edge to all nodes j such that y_j occurs in the formula F_i. We label each internal node by the function it computes, i.e. $\land$, $\lor$, or $\neg$, and add an additional outgoing edge from the node s. The *depth* of a circuit is defined analogously to formulas: the depth of a source to sink path is $\text{Depth}(L(p))$, where the function L is defined as for formulas, and the depth of the circuit is the maximum depth of any source to sink path.

Let us recall that most Boolean functions require large circuits.

Theorem 2.1.2 ([Sha49]). *Asymptotically almost surely as* $n \to \infty$ *a random Boolean function* $f : \{0,1\}^n \to \{0,1\}$ *needs a circuit of size at least* $\Omega(2^n/\log n)$.

2.1.3 The Complexity Classes P, P/poly, NP and coNP

A language $L \subseteq \Sigma^*$ is computable in polynomial time if there is an algorithm A that takes a poly-bounded number of steps in the input length, such that $A(\ell) = 1$ if and only if $\ell \in L$. We denote by P all languages that are computable in polynomial time.

Similarly, we say that a language $L \subseteq \{0,1\}^*$ is in P/poly if there is a polynomial p and a family of circuits $(C_n)_{n\in\mathbb{N}}$, each circuit C_n of size at most $p(n)$, such that $C_n(\alpha) = 1$ if and only if $\alpha \in L \cap \{0,1\}^n$.

The class NP consists of all languages $L \subseteq \Sigma_1^*$ for which there is an algorithm A, taking a poly-bounded number of steps in the input length, such that for all $\ell \in L$ there is an $x \in \Sigma_2^*$ such that $|x| = \text{poly}(|\ell|)$ and $A(\ell, x) = 1$. The class coNP consists of all languages whose complement is in NP, i.e., all languages $L \subseteq \Sigma^*$ such that the language $\bar{L} = \Sigma^* \setminus L$ is in NP.

A language L is complete for a class C if L is in C and for all other problems $L' \in C$ there is an algorithm A, taking a poly-bounded number of steps in the input length, such that $A(\ell') \in L$ if and only if $\ell' \in L'$. It is well-known that the languages SAT and CNF-SAT are NP-complete while the languages TAUT, UNSAT and CNF-UNSAT are coNP-complete.

Note that $P \subseteq NP \cap coNP$. Literally the $1 million question is whether $P \neq NP$ [CJW06]. It is widely believed that P is a proper subset of NP but this statement seems to be out of reach for current techniques. A question of similar flavor is whether $NP \neq coNP$. This is also open but widely believed to be incomparable. Note that NP $\neq$ coNP implies that P $\neq$ NP, as P is closed under the complement.

2.2 Proof Systems

The following definition is due to Cook and Reckhow [CR79]. A *proof system* P for a language $L \subseteq \Sigma_1^*$ is a language in $\Sigma_1^* \times \Sigma_2^*$ such that

1. P is poly-time computable, i.e., there is an algorithm A such that $A(\ell, \pi) = 1$ if and only if $(\ell, \pi) \in P$, and A runs in time $\mathrm{poly}(|\ell|, |\pi|)$,
2. for each $\ell \in L$ there is a $\pi \in \Sigma_2^*$ such that $(\ell, \pi) \in P$, and
3. for all $\ell \in \Sigma_1^* \setminus L$ and any $\pi \in \Sigma_2^*$ it holds that $(\ell, \pi) \notin P$.

A proof system is said to be *complete* if it satisfies Item 2, and *sound* if it satisfies Item 3. Throughout the thesis we only consider complete and sound proof systems. We say that $\pi \in \Sigma_2^*$ is a P proof of $\ell \in L$ if $(\ell, \pi) \in P$. The *size* of a proof π is $|\pi|$, also denoted by $\mathrm{Size}(\pi)$, and the size of refuting ℓ in P is $\mathrm{Size}_P(\ell) = \min_\pi \mathrm{Size}(\pi)$, where the minimum ranges over all P proofs of ℓ. The proof system P for L is *poly-bounded* if there is a polynomial p such that for all $\ell \in L$ it holds that $\mathrm{Size}_P(\ell) \leq p(|\ell|)$. A *propositional proof system* is a proof system for the language CNF-UNSAT. Note that we could also consider proof systems for TAUT but it is customary to work with proof systems for CNF-UNSAT and we follow the crowd. A proof in a propositional proof system is also called a *refutation* and we use these terms interchangeably.

Proposition 2.2.1 ([CR79]). *There is a poly-bounded propositional proof system if and only if* NP = coNP.

This proposition suggest the following program to separate NP from coNP: prove for stronger and stronger proof systems that they are not poly-bounded. In the past 30 years there has been a lot of work on proving exponential lower bounds for “weak” propositional proof systems such as resolution [Tse68; Hak85; BW01; Raz04a; ABRW04; IOSS16; ABdR+18; AM19], bounded depth Frege [Ajt94; PBI93; KPW95; BP96; Ben02; PRST16; Hås20], polynomial calculus [Raz98; BGIP01; BI10; AR03; MN15; LN17], sum of squares [Gri01; MPW15; BHK+16; KMOW17; Pot17; AH19], or cutting planes [Pud97; BPR97; FPPR17; HP17]. Ultimately the hope is that we understand lower bound strategies well enough so that we can start tackling “strong” propositional proof systems like Frege [Fre79], extended Frege [CR79] or the ideal proof system [GP18].

For now we settle to prove lower bounds for "weak" proof systems in order to extend the current toolbox of lower bound techniques. This thesis fits in there and proves several new lower bounds by extending and adapting lower bounds techniques to our needs.

Before defining the proof systems that we use throughout this thesis we need a way to compare the strength of two proof systems. Let $P \subseteq \Sigma_1^* \times \Sigma_2^*$ and $Q \subseteq \Sigma_1^* \times \Sigma_3^*$ be proof systems for a language L. We say that P *poly-simulates* Q if there is a function $f : \Sigma_3^* \to \Sigma_2^*$, computable in polynomial time in the input length, such that for all $\ell \in L$ and $\pi \in \Sigma_3^*$ it holds that if $(\ell, \pi) \in Q$, then $(\ell, f(\pi)) \in P$. Put in words, the function f translates Q-proofs of ℓ to P-proofs of ℓ with at most a polynomial increase in size, for all $\ell \in L$. It is readily seen that if Q is poly-bounded and P poly-simulates Q, then P is poly-bounded as well.

Furthermore, some of the propositional proof systems are based on (semi-)algebraic reasoning. As such we need to discuss how to translate a CNF $F = \wedge_{i=1}^m C_i$ over n variables $x_1, \dots, x_n$ into a system of polynomials $\mathcal{P}_F$. We define $\mathcal{P}_F$ over the $2n$ variables $x_1, \dots, x_n, \bar{x}_1, \dots, \bar{x}_n$ and say that the system is satisfied under an assignment α if and only if all polynomials evaluate to zero under α. In order to enforce that each variable only takes Boolean values, we add the Boolean axioms

$$y(1-y)$$

to $\mathcal{P}_F$ for each variable y. We also want to ensure that $\bar{x}_i$ is the negation of x_i and thus also add the negation axioms

$$x_i + \bar{x}_i - 1$$

to $\mathcal{P}_F$. Finally we add for each clause C_i the following polynomial to $\mathcal{P}_F$. Assuming that the clause C_i can be written as $\vee_{j=1}^{w_i} v_{ij}$, for the appropriate literals v_{ij}, we translate C_i into the monomial

$$\prod_{i=1}^{w_i} \bar{v}_{ij} \quad ,$$

where we use the convention that $\bar{\bar{x}}_k = x_k$. Assuming that the Boolean axioms and the negation axioms are honored it should be evident that above polynomial is equal to 0 if and only if one literal v_{ij} is set to 1 and thus the clause C_i is satisfied. For a CNF F, let us denote by $\mathrm{Deg}(F)$ the maximum degree of any polynomial occurring in $\mathcal{P}_F$.

2.2.1 Resolution

Resolution is arguably the most studied proof system and has been first defined by Blake [Bla37] in the 30ies of the last century. Resolution is a propositional proof system with refutations of the following form. A resolution refutation π of an unsatisfiable CNF F is a sequence of clauses $(C_1, \ldots, C_L)$ such that $C_L = \bot$ is the empty clause and each clause C_i either occurs in F or is derived from come clauses C_j and C_k, for $j, k < i$, by the *resolution rule*

$$\frac{B \lor x \qquad C \lor \overline{x}}{B \lor C} \ .$$

The *length* of π, denoted $\mathrm{Length}(\pi)$, is L and the *width* of π is the maximum width of any clause occurring in π. We denote by $\mathrm{Width}_R(F)$ the minimum width of any resolution refutation of F and, similarly, let $\mathrm{Length}_R(F) = \min_\pi \mathrm{Length}(\pi)$ where π ranges over all resolution refutations of F.

The most common way to prove resolution size lower bounds is by proving a lower bound on the width required to refute a formula and then applying the following width-length trade off due to Ben-Sasson and Wigderson.

Theorem 2.2.2 ([BW01]). *For any CNF F over n variables it holds that*

$$\mathrm{Length}_R(F) = \exp\left(\Omega\left(\frac{(\mathrm{Width}_R(F) - \mathrm{Width}(F))^2}{n}\right)\right) \ .$$

Thus, assuming that the CNF F is of constant width, if we manage to show a width lower bound linear in the number of variables, then we obtain a $2^{\Omega(n)}$ length lower bound on any resolution refutation of F. It is worth noting that this is essentially optimal as there is always a resolution refutation of length 2^n.

2.2.2 Bounded Depth Frege

Let us first define the more general Frege proof system that can derive any consequence from a set of axioms. We then specialize the system to a propositional proof system. A k-ary Frege rule is any rule of the form

$$\frac{A_1 \qquad A_2 \qquad \ldots \qquad A_k}{B}$$

such that $A_1, \ldots, A_k \models B$. Let $\mathcal{R}$ be a set of Frege rules. A Frege proof of the formula G from the formulas $F_1, \ldots, F_m$ over $\mathcal{R}$ is a sequence of formulas $(H_1, \ldots, H_L)$ such that $H_L = G$ and each line H_i is either equal to an axiom F_j or is derived from $H_{j_1}, \ldots, H_{j_k}$, for $j_1, \ldots, j_k < i$, by a k-ary Frege rule in $\mathcal{R}$.

The width of a proof is the size of the largest formula occurring, the length is the number of formulas in the proof and the depth of a proof is the maximum depth of any formula appearing.

A Frege proof system $\mathcal{F}^p$ is a finite set of Frege rules $\mathcal{R}$ that are implicationally complete, i.e., if the formulas $F_1, \ldots, F_m$ logically imply G, then there is a Frege proof of G from the fromulas $F_1, \ldots, F_m$ over $\mathcal{R}$. The following theorem due to Reckhow states that any two Frege systems poly-simulate each other.

Theorem 2.2.3 ([Rec75]). *Any two Frege propositional proof systems poly-simulate each other. Furthermore, the simulation preserves the depth of the proof up to an additive constant, while the size of the proof increases by at most a multiplicative constant.*

Reckhow in fact proved a more general statement, which also allows a certain freedom in the choice of the basis over which the formulas are defined. We stick to the DeMorgan basis and this extension is thus not central to the following discussion. The interested reader is encouraged to consult [Kra19] for a more complete treatment.

As different Frege systems poly-simulate each other, we may consider any such system. Fix an arbitrary Frege proof system $\mathcal{F}^p$ as defined above. We define the Frege propositional proof system $\mathcal{F}$ to be the language that contains all tuples (F, π), where $F = \bigwedge_{i=1}^{m} C_i$ is an unsatisfiable CNF and π is an $\mathcal{F}^p$ proof of $\bot$ from the formulas $C_1, \ldots, C_m$.

Similarly we define the depth d Frege refutational proof system $\mathcal{F}_d$: it is the language that contains all tuples (F, π) from $\mathcal{F}$ where π is of depth at most d.

Note that by Theorem 2.2.3 there is a constant d_0, such that for all $d \geq d_0$, the proof systems $\mathcal{F}_d$ is complete. We only consider such d in this thesis. Note that when the depth is reduced, then the size of a proof may blow up substantially.

2.2.3 Polynomial Calculus

The polynomial calculus propositional proof system, introduced in [CEI96] though we use the slightly stronger version introduced in [ABRW02], is based on algebraic reasoning. A polynomial calculus refutation π of the CNF F over a field $\mathbb{F}$ consists of an ordered sequence of polynomials $(p_1, \ldots, p_L)$ such that p_L is the constant 1 polynomial and each polynomial p_i occurs either in $\mathcal{P}_F$ or is derived by one of the following two polynomial calculus rules from p_j and p_k, where $j, k < i$,

$$\frac{q_1 \qquad q_2}{\alpha q_1 + \beta q_2} \qquad\qquad\qquad \frac{q}{xq}\ ,$$

for some variable x and constants $\alpha, \beta \in \mathbb{F}$. We denote the polynomial calculus propositional proof system over $\mathbb{F}$ by $\mathrm{PC}_{\mathbb{F}}$.

The *degree* of π is the maximum degree of any polynomial that occurs in π and the length of π is L. The degree of refuting F in $\mathrm{PC}_{\mathbb{F}}$ is denoted by $\mathrm{Deg}_{\mathrm{PC}_{\mathbb{F}}}(F)$ and defined as the minimum degree of any polynomial calculus refutation over $\mathbb{F}$ of the CNF F. Finally, let us also define the *monomial size* of a refutation, which simply counts the number of monomials occurring in the proof and the monomial size of refuting F in $\mathrm{PC}_{\mathbb{F}}$, denoted by $\mathrm{MSize}_{\mathrm{PC}_{\mathbb{F}}}(F)$, is the minimum monomial size of any $\mathrm{PC}_{\mathbb{F}}$ refutation.

While for resolution we had a width-length trade off, for polynomial calculus we have a degree-monomial-size trade off.This trade off in fact predates the celebrated resolution trade off. Recall that $\mathrm{Deg}(F)$ denotes the maximum degree of any polynomial occurring in $\mathcal{P}_F$.

Theorem 2.2.4 ([CEI96; IPS99]). *For any unsatisfying CNF F over n variables and any field $\mathbb{F}$ it holds that*

$$\mathrm{MSize}_{\mathrm{PC}_{\mathbb{F}}}(F) = \exp\left(\Omega\left(\frac{(\mathrm{Deg}_{\mathrm{PC}_{\mathbb{F}}}(F) - \mathrm{Deg}(F))^2}{n}\right)\right) .$$

Let us remark that polynomial calculus over any field $\mathbb{F}$ simulates resolution with respect to size as well as degree [ABRW02], i.e., a resolution proof of size s and width w can be translated into a $\mathrm{PC}_{\mathbb{F}}$ proof of size $O(s)$ and degree $O(w)$.

2.2.4 Sum of Squares

Sum of Squares (SoS) is a propositional proof system based on semi-algebraic reasoning. In contrast to the other proof systems SoS is static meaning that a proof is a single line. In fact, it is a single polynomial identity.

A Sum of Squares (SoS) refutation of a CNF F consists of a sequence of polynomials $(r_1, \dots, r_m, s_1, \dots, s_t)$ such that

$$\sum_{i \in [m]} r_i p_i + \sum_{i \in [t]} s_i^2 = -1 \ ,$$

for $\mathcal{P}_F = \{p_1, \dots, p_m\}$. As for polynomial calculus one can define the degree and monomial size of a refutation. For SoS there is also a degree-monomial-size trade off.

Theorem 2.2.5 ([AH19]). *For any unsatisfying CNF F over n variables it holds that*

$$\mathrm{MSize}_{\mathrm{SoS}}(F) = \exp\left(\Omega\left(\frac{(\mathrm{Deg}_{\mathrm{SoS}}(F) - \mathrm{Deg}(F))^2}{n}\right)\right) .$$

It is not so hard to show that SoS poly-simulates resolution. Quite surprisingly, Berkholz [Ber18] showed that SoS also poly-simulates $PC_{\mathbb{R}}$. It should be mentioned that there is a separation for finite fields and this is thus the best one could hope for.

2.3 Propositional Formulas

We are interested in several simple principles which, as we show in the following, turn out to be hard for different proof systems.

2.3.1 Tseitin

The Tseitin formula is defined over a graph $G = (V, E)$. At the heart of the formula is the following idea: count the number of edges by a double sum over the vertices and a sum over the edges incident to a single vertex

$$\frac{1}{2}\sum_{v \in V}\sum_{e : v \in e} 1 = m \ ,$$

where we divide by 2 as each edge is summed over twice. This implies in particular that the sum

$$\sum_{v \in V}\sum_{e : v \in e} 1 = 2m$$

is equal to an even number. Moreover, if we associate each edge e with a Boolean variable x_e, no matter the assignment $\alpha : \{x_e \mid e \in E\} \to \{0, 1\}$ this sum is still even:

$$\sum_{v \in V}\sum_{e : v \in e} \alpha(x_e) = 2|\alpha^{-1}(1)| \ . \tag{2.1}$$

Suppose we are given a charge function $\tau : V \to \{0, 1\}$ that assigns each vertex either a charge of 0 or 1 and, furthermore, that we enforce that at each vertex v the incident edge variables sum to the charge $\tau(v)$ modulo 2. That is, for each vertex $v \in V$ we have the constraint

$$\sum_{e : v \in e} x_e = \tau(v) \mod 2 \ . \tag{2.2}$$

All these constraints together imply that the double sum from Equation (2.1) is equal to

$$\sum_{v \in V}\sum_{e : v \in e} x_e = \sum_{v \in V} \tau(v) \mod 2 \ .$$

Thus if $|\tau^{-1}(1)|$ is odd, then there is no satisfying assignment. This defines the Tseitin contradiction: fix a charge function that assigns an odd charge to the graph and claim that there is an assignment as described above.

The CNF associated with above system of equations modulo 2 is defined as follows. For each vertex $v \in V$ and a fixed τ note that the vertex axiom of v (Axiom 2.2) can be used to define a Boolean function, mapping from $\deg(v)$ many inputs to TRUE, FALSE, depending on whether the constraint is satisfied by a given assignment, identifying TRUE with 1 and FALSE with 0. By Proposition 2.1.1 we may obtain a CNF $F_{v,\tau}$ of size $O(\deg(v) \cdot 2^{\deg(v)})$ that is satisfied if and only if the corresponding vertex axiom is satisfied. Then, the CNF associated with the Tseitin contradiction is defined as

$$\mathrm{Tseitin}(G, \tau) = \bigwedge_{v \in V} F_{v,\tau} \quad .$$

Observe that this formula is of exponential size in the maximum degree of the graph. As such this formula is usually only considered in combination with graphs of constant degree.

2.3.2 Perfect Matching

The perfect matching formula is also defined over a graph $G = (V, E)$. The formula claims that there is a subset of edges $M \subseteq E$ such that every edge is in precisely one edge of M. In other words, we want a set $M \subseteq E$ such that for every vertex $v \in V$ it holds that

$$|\{e \in M \mid v \in e\}| = 1 \quad .$$

A moment of reflection reveals that a matching always contains an even number of vertices: every edge in M consists of two vertices, each of which is in precisely one edge and hence there is an even number of vertices matched. Thus if we start with a graph G defined over an odd number of vertices, there cannot be a perfect matching in this graph. This defines the perfect matching contradiction.

We have a variable per edge $\{x_e \mid e \in E\}$ and for each vertex $v \in V$ we add the following clauses to the CNF formula $\mathrm{PM}(G)$:

$$\bigvee_{e : v \in e} x_e \quad , \qquad \text{and} \qquad \bar{x}_e \vee \bar{x}_{e'} \quad ,$$

for any $e, e' \in E$ such that $e \cap e' = \{v\}$.

2.3.3 Pigeonhole Principle

One encoding of the pigeonhole principle (PHP) is the perfect matching formula $\mathrm{PM}(G)$ defined over a bipartite graph $G = (U, V, E)$ with $|U| > |V|$.

This is the strongest encoding of the principle: there are subformulas of PM(G) that are contradictions. Let us define two subformulas of interest to us.

Recall that we have a variable for each edge in the graph $\{x_e \mid e \in E\}$. The ordinary pigeonhole principle PHP(G), for $G = (U, V, E)$, has a clause per vertex $u \in U$

$$\bigvee_{e : u \in e} x_e \quad , \tag{2.3}$$

ensuring that the vertex u is matched to some vertex in the neighborhood, and for any two distinct edges $\{u, v\}, \{u', v\} \in E$, where $v \in V$ and $u, u' \in U$, we add the axiom

$$\bar{x}_e \vee \bar{x}_{e'} \quad . \tag{2.4}$$

These axioms ensure that each vertex in V is matched to at most one vertex in U. This is a contradiction if $|U| > |V|$.

We can add further axioms, making the formula more and more constrained and thus easier to refute. The functional pigeonhole principle FPHP(G) is defined by also adding Axiom 2.4 for distinct edges $\{u, v\}, \{u, v'\} \in E$, where $u \in U$ and $v, v' \in V$. Finally, we can recover the perfect matching principle PM(G) by adding Axiom 2.3 for every vertex V.

2.3.4 The Truthtable Formula

The truthtable formula is a bit more involved to define. We want to encode the claim that a Boolean function $f : \{0,1\}^n \to \{0,1\}$, given as a binary string of length 2^n, has a circuit of size at most s. By Theorem 2.1.2 most Boolean functions require circuits of size $\Omega(2^n/\log n)$ and thus for $s \ll 2^n/\log n$, this formula is a contradiction for most Boolean functions f.

The formula $\mathrm{Circuit}_s(f)$ consists of two parts. The first part of the formula defines a circuit, and the second part of the formula ensures that the circuit encoded by the first part indeed computes f. In the following we describe a polynomial encoding of the truthtable formula. This encoding conveys enough of an idea for the following discussion and is already cumbersome to explain – the CNF translation consists of even more axioms and variables. For the formal encoding used in Paper C we recommend the interested reader to consult Section C.2.3 as well as Section C.7.

Let us first describe the *structure variables* which are used in the first part of the formula to describe the circuit.

Each of the s gates is indexed from 1 to s, with the output gate being gate s. This labeling will be topological, in a sense that each gate $v \in [s]$

has no input from a gate $u > v$. Each gate $v \in [s]$ has three variables $\mathrm{isNeg}(v)$, $\mathrm{isOr}(v)$ and $\mathrm{isAnd}(v)$ associated with it, indicating the operation computed at v. Similarly, for a gate $v \in [s]$ and a wire $a \in \{1,2\}$ we have variables $\mathrm{isFromConst}(v, a)$, $\mathrm{isFromInput}(v, a)$ and $\mathrm{isFromGate}(v, a)$ indicating whether the input wire a of v is connected to a constant, a variable or a gate.

Finally we have the variables indicating to what constant, variable or gate an input wire is connected to: We have variables $\mathrm{constantValue}(v, a)$, $\mathrm{isInput}(v, a, i)$ and $\mathrm{isGate}(v, a, u)$, for $a \in \{1,2\}$, $i \in [n]$ and $u < v$. The corresponding variables indicate the constant value, the input x_i or the gate u that the input wire a of v is connected to – assuming a is indeed connected to the corresponding kind.

The structure variables come along with a set of axioms that we refer to as the *structure axioms*. The first axioms ensure that every wire is connected to a single kind

$$\mathrm{isFromConst}(v, a) + \mathrm{isFromInput}(v, a) + \mathrm{isFromGate}(v, a) = 1 \quad \forall\, v \in [s] \ ,$$

and the second group of axioms makes sure that each gate is of precisely one kind

$$\mathrm{isNeg}(v) + \mathrm{isOr}(v) + \mathrm{isAnd}(v) = 1 \quad \forall\, v \in [s] \ .$$

The final group of structure axioms ensures that the variables that indicate to what input or gate a fixed wire is connected to always sum to one, except at gate 1 as it cannot have any inputs from other gates

$$\sum_{i=1}^{n} \mathrm{isInput}(v, a, i) = 1 \quad \forall\, v \in [s] \ , \text{ and}$$

$$\sum_{u=1}^{v-1} \mathrm{isGate}(v, a, u) = 1 \quad \forall v \in [s] \setminus \{1\} \ .$$

This completes the description of the first part of the formula.

The second part of the formula is defined over the so-called *evaluation variables* which describe what value is computed at each gate v on input $\alpha = \alpha_1, \ldots, \alpha_n$.

Each gate has $3 \cdot 2^n$ evaluation variables associated with it. These are $\mathrm{out}_\alpha(v)$, indicating the Boolean value computed at gate $v \in [s]$ on input $\alpha \in \{0,1\}^n$, and the variables $\mathrm{in}_\alpha(v, a)$ which indicate the value brought to vertex $v \in [s]$ on wire $a \in \{1,2\}$ on input $\alpha \in \{0,1\}^n$.

Note that we have $\Theta(s^2 + sn)$ structure variables and $3s2^n$ evaluation variables, for a total of $\Theta(s^2 + s2^n)$ variables in $\mathrm{Circuit}_s(f)$.

The evaluation variables are accompanied by the *evaluation axioms* ensuring that the evaluation variables indeed compute the intended values. The first set of axioms ensures that the wires carry the value intended by the structure axioms. If a wire is connected to a constant, then the evaluation variable associated with that wire should always be equal to the constant

$$\mathrm{isFromConst}(v, a) \cdot \big(\mathrm{in}_\alpha(v, a) - \mathrm{constantValue}(v, a)\big) = 0 \ ,$$

and similarly if a wire is connected to an input or a gate

$$\begin{aligned} \mathrm{isFromInput}(v, a) \cdot \mathrm{isInput}(v, a, i) \cdot \big(\mathrm{in}_\alpha(v, a) - \alpha_i\big) &= 0 \ , \\ \mathrm{isFromGate}(v, a) \cdot \mathrm{isGate}(v, a, u) \cdot \big(\mathrm{in}_\alpha(v, a) - \mathrm{out}_\alpha(u)\big) &= 0 \ . \end{aligned}$$

The final set of evaluation axioms makes sure that the output evaluation variable of a gate is correctly related to the input evaluation variables:

$$\begin{aligned} \mathrm{isNeg}(v) \cdot \mathrm{out}_\alpha(v) &= \mathrm{isNeg}(v) \cdot \overline{\mathrm{in}_\alpha(v, 1)} \ , \\ \mathrm{isOr}(v) \cdot \mathrm{out}_\alpha(v) &= \mathrm{isOr}(v) \cdot \big(1 - \overline{\mathrm{in}_\alpha(v, 1)} \cdot \overline{\mathrm{in}_\alpha(v, 2)}\big) \ , \\ \mathrm{isAnd}(v) \cdot \mathrm{out}_\alpha(v) &= \mathrm{isAnd}(v) \cdot \mathrm{in}_\alpha(v, 1) \cdot \mathrm{in}_\alpha(v, 2) \ . \end{aligned}$$

Last but not least we have the axioms that ensure that the circuit outputs the function specified by the truthtable

$$\mathrm{out}_\alpha(s) = f(\alpha) \ .$$

Contributions

In this chapter we highlight the main results of the included papers. Each paper is discussed in a separate section. The sections first give some context how our results fit into the literature, followed by the main theorem and a brief discussion about the employed proof techniques.

3.1 On Bounded Depth Frege Refutations of the Tseitin Formula

Johan Håstad and Kilian Risse, *"On Bounded Depth Proofs for Tseitin Formulas on the Grid; Revisited"*, accepted to FOCS'22 [HR22]

Paper A concerns bounded depth Frege refutations of the Tseitin contradiction defined over grid graphs.

The study of bounded depth Frege refutations was initiated by Ajtai [Ajt94] who proved that the PHP cannot be refuted in polynomial size for any constant depth Frege system. This pioneering result was followed up by several papers in the 1990s, first improving Ajtai's result to hold up to depth $O(\log \log n)$ [PBI93; KPW95], and then extending it to the Tseitin contradiction defined over complete [UF96], as well as expander graphs [Ben02].

These developments followed previous work where the computational power of the class of bounded depth circuits[1] was studied [Sip83; FSS84; Yao85; Hås86; Raz88; Smo87]. It should not be surprising that it is simpler to argue about the computational power of a single circuit rather than a sequence of formulas forming a proof. This is exemplified by the lower bounds achieved by the end of the 1990s: while the bounded depth Frege lower bounds remained stuck at depth $O(\log \log n)$, the results for circuit size extended to almost logarithmic depth.

This gap was recently closed in two steps. First Pitassi et al. [PRST16] obtained super-polynomial bounded depth Frege lower bounds up to depth

[1] As in the bounded depth setting there is no major difference between circuits and formulas we gloss over the difference between these.

$o(\sqrt{\log n})$ and then Håstad [Hås20] managed to extend these results up to depth $\Theta(\frac{\log n}{\log \log n})$, which matches the result for circuits up to constants.

All these previous bounded depth Frege lower bounds considered the total size of a proof. The total size is composed of the length (number of steps) of a refutation and the size of each line. For some proof systems, such as resolution, each line is bounded in size and hence any super-polynomial lower bound on proof size also implies a lower bound on the number of proof steps. This is not necessarily true for bounded depth Frege – the line size may grow and it is thus an interesting question to study the number of lines required, given that each line is of bounded size.

This line of investigation was recently initiated by Pitassi et al. [PRT22]. They consider the Tseitin contradiction defined over the grid of size $n \times n$ and showed that if each line of the refutation is limited to size M and depth d, then a Frege refutation must consist of at least $\exp\left(n/2^{O(d\sqrt{\log M})}\right)$ many lines. For most interesting values of M this greatly improves the bounds implied by the results for total proof size of Håstad [Hås20]. In particular, if M is a polynomial, then the lower bounds are of the form $\exp(n^{1-o(1)})$, as long as $d = o(\sqrt{\log n})$, in contrast to the total size lower bounds of the form $\exp(n^{\Omega(1/d)})$. Pitassi, Ramakrishnan, and Tan [PRT22] rely on the restrictions introduced by Håstad [Hås20] but analyze them using the methods of Pitassi et al. [PRST16].

We study the same Tseitin contradiction on the grid but analyze the restrictions using the machinery set up by Håstad [Hås20]. This allows us to improve the result of Pitassi et al. [PRT22] to obtain the lower bound conjectured by them.

Theorem 3.1.1. *For any Frege refutation of the Tseitin principle defined over the* $n \times n$ *grid graph the following holds. If each line of the refutation is of size* M *and depth* d, *then the length of the refutation is*

$$\exp\left(\frac{n}{\left((\log n)^{O(1)} \log M\right)^{2d}}\right) .$$

Along the way we also improve the parameters of the refutation size lower bound due to Håstad [Hås20] from exponential in $\tilde{\Omega}(n^{1/59d})$ to exponential in $\tilde{\Omega}(n^{1/(2d-1)})$.

Theorem 3.1.2. *For* $d \leq O\left(\frac{\log n}{\log \log n}\right)$ *the following holds. Any depth-*d *Frege refutation of the Tseitin contradiction defined on the* $n \times n$ *grid requires size*

$$\exp\left(\Omega(n^{1/(2d-1)}(\log n)^{O(1)})\right) .$$

We achieve the improvements on total size by revisiting Håstad's original proof and carefully eliminating some undesired dependencies on the depth in the switching lemma. This forces us to use slightly more general restrictions for book-keeping but the over all proof remains unchanged.

We then use this improved proof of the switching lemma to obtain a multi-switching lemma with which we are able to prove Theorem 3.1.1. In order to prove the multi-switching lemma we need to analyze a new combinatorial game played on the grid graph. Already Håstad [Hås20] needed to analyze such a combinatorial game. This new game is quite a bit more complicated and requires an entirely new amortized analysis.

3.2 Average-Case Perfect Matching Lower Bounds

Per Austrin and Kilian Risse, *"Perfect Matching in Random Graphs is as Hard as Tseitin"*, SODA'22, to appear in TheoretiCS [AR22a]

This paper studies the power (or lack thereof) of the SoS, PC and bounded depth Frege proof systems when it comes to refuting the perfect matching formula $\mathrm{PM}(G)$ defined over sparse random graphs G on an odd number of vertices. Apart from being a natural and well-studied problem on its own, the perfect matching formula is interesting because of its close relation to two other widely studied families of formulas, namely the pigeonhole principle (PHP) and the Tseitin formula.

While most variants of the PHP are hard for PC [Raz98; MN15], the perfect matching variant is in fact easy to refute over any field [Rii93] and in SoS all variants of the PHP are easy to refute [GHP02]. On the other hand the Tseitin formula is (almost) always hard: for $\mathrm{PC}_{\mathbb{F}}$ over fields $\mathbb{F}$ of characteristic distinct from 2 [BGIP01; AR03] and SoS [Gri01] these formulas require linear degree if the underlying graph G is a good expander.[2]

Hence the perfect matching formula lies somewhere in between the easy PHP formula and the hard Tseitin formula and it is natural to wonder whether SoS or PC requires large degree to refute the perfect matching formula over non-bipartite graphs.

This is well understood if the perfect matching principle is defined over a *complete graph* on an odd nuber of vertices (also know as "MOD 2 principle"): the proof systems SoS and PC require degree $\Omega(n)$, except for PC defined over fields of characteristic 2 [BGIP01; Gri01]. Less is known for sparse graphs: Buss et al. [BGIP01] obtained worst-case lower bounds for PC showing that there exist bounded degree graphs on n vertices requiring $\Omega(n)$ degree refutations. This is obtained by a reduction from Tseitin formulas and while the work of Buss et al. predates the current interest in

[2] Observe that we cannot hope to prove degree lower bounds over fields of characteristic 2 as the constraints become linear and we can thus refute the Tseitin formula using Gaussian elimination. As the perfect matching formula $\mathrm{PM}(G)$ implies the Tseitin formula, PC over fields of characteristic 2 can easily refute $\mathrm{PM}(G)$, if G has an odd number of vertices.

the SoS system, it is not hard to see that the same reduction yields a similar $\Omega(n)$ degree lower bound for SoS.

However, for random graphs G little is known about the hardness of the perfect matching formula and, e.g., Razborov [Raz17] asked whether it is true that the Lovász-Schrijver hierarchy [LS91] (a proof system poly-simulated by SoS) requires n^ε rounds to refute the perfect matching principle on a random sparse regular graph with high probability. We answer this question by proving the following theorem.

Theorem 3.2.1. *There is a constant* d_0 *such that for all constants* $d \geq d_0$ *the following holds asymptotically almost surely for* $G \sim \mathcal{G}(n, d)$.

1. $\mathrm{Deg}_{\mathrm{PC}_{\mathbb{F}}}\big(\mathrm{PM}(G)\big) = \Omega(n/\log n)$ *for any fixed field* $\mathbb{F}$ *with* $\mathrm{char}(\mathbb{F}) \neq 2$.

2. $\mathrm{Deg}_{\mathrm{SoS}}\big(\mathrm{PM}(G)\big) = \Omega(n/\log n)$.

3. *There is a* $\delta > 0$ *such that* $\mathrm{Size}_{\mathcal{F}_d}\big(\mathrm{PM}(G)\big) = \exp\big(\Omega(n^{\delta/D})\big)$, *for all* $D \leq \frac{\delta \log n}{\log \log n}$.

Using the known degree-monomial-size tradeoffs for Polynomial Calculus [IPS99; CEI96] and Sum of Squares [AH19], the degree lower bounds from Theorem 3.2.1 imply near-optimal monomial size lower bounds of $\exp\big(\Omega(n/\log^2 n)\big)$.

We obtain these lower bounds by a worst-case to average-case reduction. We achieve this by using the embedding technique as introduced to proof complexity by Pitassi et al. [PRST16]: for, say, the SoS lower bound, our starting point is the $\Omega(n)$ *worst-case* degree lower bound in sparse graphs, and we then prove that these hard instances can be embedded in a random d-regular graph in such a way that the hardness of refuting the formula is preserved.

There are two main components to this argument. One of them is a new graph embedding theorem which may be of independent interest. Very loosely speaking, we show that any bounded-degree graph with $O(n/\log n)$ edges can be embedded as a *topological minor* into any bounded-degree α-expander on n vertices. But this does not quite suffice: for our application we also need to be able to control the parities of the path lengths used in the topological embedding. We show that as long as every large linear-sized subgraph contains an odd cycle of length $\Omega(1/\alpha)$, this is indeed possible. The following is a quite informal statement of our embedding theorem.

Theorem 3.2.2 (Informal)**.** *Let* G *be a constant degree* α*-expander on* n *vertices. If* H *is a graph with at most* $\frac{\varepsilon n}{\log n}$ *edges and* $\Delta(H) \ll \alpha^2 \cdot d(G)$, *then* G *contains* H *as a topological minor. Furthermore, if all large vertex induced subgraphs of*

G *contain an odd cycle of length* $\Omega(1/\alpha)$*, then one can choose the parities of the length of all the edge embeddings in the minor.*

This generalizes various classical results of a similar flavor (e.g. [KR96; KN19; CN19; Kri19]). See Paper B for a discussion comparing these (and other) existing embedding results to Theorem 3.2.2.

To motivate the second component of our worst-case to average-case, we need to look the reduction in a bit more detail. A quite naïve attempt to obtain average-case lower bounds from a sparse worst-case instance H on n vertices is to topologically embed the worst-case instance into a random regular graph G on $O(n \log n)$ vertices using Theorem 3.2.2. One would then like to argue that $\mathrm{PM}(G)$ is hard.

Suppose each path p_{uv} in the embedding of H in G corresponding to some edge $\{u, v\} \in E(H)$ is of odd length. Then it is straightforward to verify that the perfect matching formula defined over the embedding is at least as hard to refute as the worst-case instance $\mathrm{PM}(H)$: map each variable y_e, for $e \in p_{uv}$, alternatingly to x_{uv} or $\bar{x}_{uv}$ such that the first and last edges of p_{uv} are mapped to x_{uv} (using that p_{uv} is of odd length). This simple projection maps the perfect matching formula defined over the embedding of H to $\mathrm{PM}(H)$ and thus shows that the hardness of $\mathrm{PM}(H)$ should be inherited.

But having such a worst-case instance as a topological minor is *not* sufficient to conclude that $\mathrm{PM}(G)$ is hard. For instance G may contain an isolated vertex and it is then trivial to refute $\mathrm{PM}(G)$. On the other hand if we could guarantee that there is a perfect matching M in the subgraph of G induced by the vertices *not* used in the embedding of H, we can conclude that $\mathrm{PM}(G)$ is hard: hit the formula with the restriction corresponding to the matching M and by the argument from the previous paragraph we are basically left with the worst-case formula.

Thus if we can ensure that H is a topological minor of G with the two additional properties that (i) every path used in the embedding of H has odd length, and (ii) there exists a perfect matching in the subgraph of G induced by the vertices *not* used in the embedding of H, then we obtain average-case lower bounds for the perfect matching formula $\mathrm{PM}(G)$.

Let us elaborate a bit further on the properties required from the topological minor of H in G. As mentioned previously, our embedding theorem can ensure that all paths are of odd length. To ensure the second property we in fact do not embed H directly into G but rather into a suitably chosen vertex induced subgraph $G[T]$ with the crucial property that for any set of vertices $U \subseteq T$ of odd cardinality the induced subgraph $G[V \setminus U]$ has a perfect matching. As the embedding of H will consist of an odd number of vertices we then obtain property (ii) above. Since we now want to apply Theorem 3.2.2 not to G but to G[T], we have to ensure that G[T] satisfies all the conditions of that theorem. We prove what we refer to as the Partition Lemma, which asserts that an induced subgraph G[T] exists that satisfies both the perfect matching property described above, as well as all conditions of Theorem 3.2.2. The proof of the Partition Lemma relies primarily on the Lovász Local Lemma and spectral bounds to obtain the desired properties.

3.3 The Circuit Tautology is Hard for Sum of Squares

Per Austrin and Kilian Risse, *"The Minimum Circuit Size Problem is Hard for Sum of Squares"*, in submission [AR22b]

The minimum circuit size problem (MCSP), is central to complexity theory: given the truthtable of a Boolean function $f : \{0,1\}^n \to \{0,1\}$ and a parameter s, the MCSP problem asks whether there is a Boolean circuit of size at most s computing f. The MCSP is clearly in NP: given a circuit of size at most s we can easily check in time $O(s \cdot 2^n)$ whether this circuit computes f.

Despite decades of research it is not known whether the MCSP problem is NP-hard. In fact establishing this has been shown to be a difficult itself [KC00; MW15; Hir18]. As such it is an important goal to at least rule out that certain classes of efficient algorithms solve the MCSP problem.

Despite the intrinsic motivation to study MCSP, there are further good arguments from a proof complexity viewpoint to study this problem. For starters, the MCSP problem is believed to be a source of hard formulas even for strong proof systems. There are not many formulas that are conjectured hard for strong proof systems and as such it is important to at least establish this claim for weak proof systems. There are some lower bounds for resolution [Raz04a; Raz04b], polynomial calculus [Raz98] and resolution over low width CNFs [Raz15], but due to the meta complexity flavor of this problem it seems difficult to prove strong lower bounds. It is worth mentioning that it has been stated as an explicit open problem [Raz22] to prove SoS degree lower bounds for the $\text{Circuit}_s(f)$ formula.

Another proof complexity angle that motivates the study of this formula is that it tells us how hard it is to prove circuit lower bounds: consider the formula $\text{Circuit}_s(\text{SAT})$, for $s = n^{\omega(1)}$. Proving that a proof system cannot refute this formula is essentially showing that this proof system cannot efficiently refute that problems in NP possess polynomial size circuits, i.e., the proof system cannot efficiently prove $\mathsf{NP} \not\subseteq \mathsf{P/poly}$.

The main result of this paper is an essentially optimal[3] degree lower

[3] There is an SoS refutation of degree s: see Section C.6 for details.

bound for any Boolean function.

Theorem 3.3.1. *For all* $\varepsilon > 0$ *there is a* $d = d(\varepsilon)$ *such that the following holds. For* $n \in \mathbb{N}$*, all* $s \geq n^d$ *and any Boolean function* $f : \{0,1\}^n \to \{0,1\}$ *on* n *bits, SoS requires degree* $\Omega_\varepsilon(s^{1-\varepsilon})$ *to refute* $\mathrm{Circuit}_s(f)$*.*

From this lower bound one can also extract a monomial size lower bound for a restricted class of circuits. Furthermore, it can also be shown that SoS requires large degree to refute the claim that monotone slice functions have small monotone circuits. For details on these results we refer the interested reader to Paper C.

The idea of the proof is to restrict the structure part of the formula $\mathrm{Circuit}_s(f)$ in such a manner that the remaining satisfying assignments to the structure axioms (ignoring all other axioms) correspond to a well-behaved class of circuits $\mathcal{C}$. In a bit more detail we want that each circuit $C_\gamma \in \mathcal{C}$, where $\gamma \in \{0,1\}^m$, on input $\alpha \in \{0,1\}^n$ computes $C_\gamma(\alpha) = \oplus_{i \in N(\alpha)} \gamma_i$, for a bipartite graph $G = (\{0,1\}^n, [m], E)$.

Once we have this family of circuits $\mathcal{C}$, SoS simply has to show that none of the circuits in $\mathcal{C}$ compute the given truthtable, i.e., SoS has to show that there is no $\gamma \in \{0,1\}^m$ such that

$$\bigoplus_{i \in N(\alpha)} \gamma_i = f(\alpha) \ ,$$

for all $\alpha \in \{0,1\}^n$. But note that this is an xor constraint satisfaction problem and Grigoriev [Gri01] proved that if G is a good expander, then SoS requires linear degree in m to refute this. Thus by setting G to be a good expander we obtain the desired degree lower bound. There are some details to be filled in but this is the general gist of the argument.

3.4 The Sparse Weak Pigeonhole Principle is Hard for Resolution

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"*, CCC'20 [dRNRS20]

As previously mentioned, the main aim of proof complexity is to prove superpolynomial lower bounds for stronger and stronger proof systems to establish that $\mathsf{NP} \neq \mathsf{coNP}$. A slightly different strand of research has been to study different combinatorial principles and investigate what kinds of arguments are needed to efficiently establish these principles. This quantifies, in a way, the mathematical "depth" of these statements in terms of how strong a proof system is required to prove them.

In this work we consider the resolution proof system and the pigeonhole principle (PHP). This is one of the simplest, and yet most useful, combinatorial principles in mathematics and it has been widely studied in proof complexity. We consider the somewhat unorthodox setting when m is a super-polynomial function of n. This setting has been useful to establish that resolution refutations of the claim $\mathsf{NP} \not\subseteq \mathsf{P}/\mathsf{poly}$ are of doubly exponential size in $n^{O(1)}$ [Raz04a; Raz04b] by a reduction from the weak pigeonhole principle to the $\mathrm{Circuit}_s(f)$ formula.

These just mentioned lower bounds break with the general paradigm of proving resolution lower bounds. As mentioned in Section 2.2.1, the most common way to prove resolution refutation size lower bounds is to prove a width lower bound to then apply the width-length trade off to obtain a length (and thus size) lower bound. As the PHP can always be refuted by a resolution proof of width at most linear in number of holes n, independent of the number of pigeons $m > n$, we see that the width-length tradeoff stops giving useful lower bounds once $m \geq n^2$, as the number of variables increase as we increase the number of pigeons. However, there are resolution size lower bounds for the setting when there are $m > n^2$ pigeons: these can be shown by the seemingly ad-hoc arguments due to Raz [Raz04a] and subsequently Razborov [Raz04b].

A further peculiarity about the latter lower bounds is that they only apply to fairly dense graphs (recall that the PHP is defined over a bipartite graph $G = (U, V, E)$), while up to $m \ll n^2$ the lower bounds also hold for constant degree graphs. As such it is natural to wonder whether (i) the lower bounds for the setting $m \geq n^2$ can be strengthened to also holds for sparse graphs, and (ii) whether there is a single framework in which all these lower bounds can be proven.

We answer the latter in the affirmative and show how to generalize the pseudo-width method, devised by Razborov [Raz01; Raz03; Raz04b] in a series of 3 papers, to also apply in the sparse case.

Let us state three examples of the kind of lower bounds we obtain – the full, formal statements can be found in Paper D. The first theorem is an average-case lower bound for perfect matching formulas over a bipartite graph with a slightly superpolynomial number of pigeons.

Theorem 3.4.1 (Informal)**.** *Let G be a randomly sampled bipartite graph with n right vertices, $m = n^{o(\log n)}$ left vertices, and left degree $\Theta(\log^2 m)$. Then refuting the perfect matching formula over G in resolution requires length $\exp(\Omega(n^{1-o(1)}))$ asymptotically almost surely.*

Note that as the number of pigeons grow larger, it is clear that the left degree also has to grow – otherwise the birthday paradox will yield a small unsatisfiable subformula that can easily be refuted by brute force.

If m increases further to weakly exponential, then randomly sampled graphs no longer have good enough expansion for our techniques. However, there are explicit constructions of unbalanced expanders for which we can still get lower bounds.

Theorem 3.4.2 (Informal)**.** *There are explicitly constructible bipartite graphs* G *with* n *right vertices,* $m = \exp\bigl(O\bigl(n^{1/16}\bigr)\bigr)$ *left vertices, and left degree* $\Theta\bigl(\log^4 m\bigr)$ *such that refuting* $\mathrm{PM}(G)$ *requires length* $\exp\bigl(\Omega\bigl(n^{1/8-\varepsilon}\bigr)\bigr)$ *in resolution.*

Finally, for functional pigeonhole principle formulas we can also prove an exponential lower bound for *constant* left degree even if the number of pigeons is a large polynomial.

Theorem 3.4.3 (Informal)**.** *Let* G *be a randomly sampled bipartite graph with* n *right vertices,* $m = n^k$ *left vertices, and left degree* $\Theta\bigl((k/\varepsilon)^2\bigr)$*. Then refuting the functional pigeonhole principle formula over* G *in resolution requires length* $\exp\bigl(\Omega\bigl(n^{1-\varepsilon}\bigr)\bigr)$ *asymptotically almost surely.*

As already mentioned, we heavily build on the pseudo-width technique devised by Razborov. In order to handle sparse bipartite graphs, we join this technique with the idea of a "closure", as introduced to proof complexity by [AR03; ABRW04]. Consider a good bipartite expander $G = (U, V, E)$. Then, the closure of a set of vertices W is a set $\mathrm{cl}(W) \supseteq W$ of vertices such that if we remove this set from G, then the resulting graph is still a fairly good expander. The maybe at first somewhat surprising fact is that for the correct setting of parameters, it can be shown that the size of the closure is linearly related to the size of W.

This notion allows us to build a matching in an iterative fashion such that the remainder of the graph is always a good expander and thus, by Hall's condition, can be extended to any small set of vertices – as if we were on a complete bipartite graph. Combining this idea with the pseudo-width technique turns out to be fairly involved and we recommend the interested reader to consult the introduction of Paper D for a more detailed proof overview.

References

[Ajt94] M. Ajtai, ”The complexity of the pigeonhole principle”, *Combinatorica*, vol. 14, no. 4, pp. 417–433, 1994, Preliminary version in *FOCS ’88* (cit. on pp. 16, 27)

[ABRW02] M. Alekhnovich, E. Ben-Sasson, A. A. Razborov and A. Wigderson, ”Space complexity in propositional calculus”, *SIAM Journal on Computing*, vol. 31, no. 4, pp. 1184–1211, Apr. 2002, Preliminary version in *STOC ’00* (cit. on pp. 19, 20)

[ABRW04] ——, ”Pseudorandom generators in propositional proof complexity”, *SIAM Journal on Computing*, vol. 34, no. 1, pp. 67–88, 2004. DOI: 10.1137/S0097539701389944 (cit. on pp. 16, 35)

[AR03] M. Alekhnovich and A. A. Razborov, ”Lower bounds for polynomial calculus: Non-binomial case”, *Proceedings of the Steklov Institute of Mathematics*, vol. 242, pp. 18–35, 2003. [Online]. Available: http://people.cs.uchicago.edu/~razborov/files/misha.pdf (cit. on pp. 16, 29, 35)

[ABdR+18] A. Atserias, I. Bonacina, S. F. de Rezende, M. Lauria, J. Nordström and A. Razborov, ”Clique is hard on average for regular resolution”, in *Proceedings of the 50th Annual ACM Symposium on Theory of Computing (STOC ’18)*, Jun. 2018, pp. 866–877 (cit. on p. 16)

[AH19] A. Atserias and T. Hakoniemi, ”Size-Degree Trade-Offs for Sums-of-Squares and Positivstellensatz Proofs”, in *34th Computational Complexity Conference (CCC 2019)*, A. Shpilka, Ed., ser. Leibniz International Proceedings in Informatics (LIPIcs), vol. 137, Dagstuhl, Germany: Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 2019, 24:1–24:20, ISBN: 978-3-95977-116-0. DOI: 10.4230/LIPIcs.CCC.2019.24. [Online]. Available: http://drops.dagstuhl.de/opus/volltexte/2019/10846 (cit. on pp. 16, 20, 30)

[AM19] A. Atserias and M. Müller, ”Automating resolution is NP-hard”, in *Proceedings of the 60th Annual IEEE Symposium on Foundations of Computer Science (FOCS ’19)*, Nov. 2019, pp. 498–509 (cit. on p. 16)

[AR22a] P. Austrin and K. Risse, ”Perfect matching in random graphs is as hard as tseitin”, in *Proceedings of the 2022 ACM-SIAM Symposium on Discrete Algorithms, SODA 2022, Virtual Conference / Alexandria, VA, USA, January 9 - 12, 2022*, J. (Naor and N. Buchbinder, Eds., SIAM, 2022, pp. 979–1012. DOI:

10.1137/1.9781611977073.43. [Online]. Available: https://doi.org/10.1137/1.9781611977073.43 (cit. on p. 29)

[AR22b] ——, ”The minimum circuit size problem is hard for sum-of-squares”, Submitted Manuscript, 2022 (cit. on p. 32)

[BHK+16] B. Barak, S. B. Hopkins, J. Kelner, P. Kothari, A. Moitra and A. Potechin, ”A nearly tight sum-of-squares lower bound for the planted clique problem”, in *2016 IEEE 57th Annual Symposium on Foundations of Computer Science (FOCS)*, 2016, pp. 428–437 (cit. on p. 16)

[BP96] P. Beame and T. Pitassi, ”An exponential separation between the parity principle and the pigeonhole principle”, *Annals of Pure and Applied Logic*, vol. 80, no. 3, pp. 195–228, Aug. 1996, Preliminary version in *LICS ’93* (cit. on p. 16)

[Ben02] E. Ben-Sasson, ”Hard examples for the bounded depth Frege proof system”, *Computational Complexity*, vol. 11, no. 3-4, pp. 109–136, 2002 (cit. on pp. 16, 27)

[BI10] E. Ben-Sasson and R. Impagliazzo, ”Random CNF’s are hard for the polynomial calculus”, *Computational Complexity*, vol. 19, no. 4, pp. 501–519, 2010, Preliminary version in *FOCS ’99* (cit. on p. 16)

[BW01] E. Ben-Sasson and A. Wigderson, ”Short proofs are narrow—resolution made simple”, *Journal of the ACM*, vol. 48, no. 2, pp. 149–169, Mar. 2001, Preliminary version in *STOC ’99* (cit. on pp. 16, 18)

[Ber18] C. Berkholz, ”The relation between polynomial calculus, Sherali-Adams, and sum-of-squares proofs”, in *Proceedings of the 35th Symposium on Theoretical Aspects of Computer Science (STACS ’18)*, ser. Leibniz International Proceedings in Informatics (LIPIcs), vol. 96, Feb. 2018, 11:1–11:14 (cit. on p. 21)

[Bla37] A. Blake, ”Canonical expressions in Boolean algebra”, Ph.D. dissertation, University of Chicago, 1937 (cit. on p. 17)

[BPR97] M. Bonet, T. Pitassi and R. Raz, ”Lower bounds for cutting planes proofs with small coefficients”, *Journal of Symbolic Logic*, vol. 62, no. 3, pp. 708–728, Sep. 1997, Preliminary version in *STOC ’95* (cit. on p. 16)

[BGIP01] S. R. Buss, D. Grigoriev, R. Impagliazzo and T. Pitassi, ”Linear gaps between degrees for the polynomial calculus modulo distinct primes”, *Journal of Computer and System Sciences*, vol. 62, no. 2, pp. 267–289, Mar. 2001, Preliminary version in *CCC ’99* (cit. on pp. 16, 29)

[CJW06] J. Carlson, A. Jaffe and A. Wiles, Eds., *The Millennium Prize Problems*. Providence, RI: American Mathematical Society, Jun. 2006 (cit. on pp. 9, 15)

[Chu36] A. Church, ”An unsolvable problem of elementary number theory”, *American Journal of Mathematics*, vol. 58, no. 2, pp. 345–363, 1936 (cit. on p. 4)

[CN19] J. Chuzhoy and R. Nimavat, *Large minors in expanders*, 2019. arXiv: `1901.09349 [cs.DS]` (cit. on p. 31)

[CEI96] M. Clegg, J. Edmonds and R. Impagliazzo, ”Using the Groebner basis algorithm to find proofs of unsatisfiability”, in *Proceedings of the 28th Annual ACM Symposium on Theory of Computing (STOC ’96)*, May 1996, pp. 174–183 (cit. on pp. 19, 20, 30)

[CR79] S. A. Cook and R. Reckhow, ”The relative efficiency of propositional proof systems”, *Journal of Symbolic Logic*, vol. 44, no. 1, pp. 36–50, Mar. 1979, Preliminary version in *STOC ’74* (cit. on pp. 8, 16)

[dRNRS20] S. F. de Rezende, J. Nordström, K. Risse and D. Sokolov, ”Exponential Resolution Lower Bounds for Weak Pigeonhole Principle and Perfect Matching Formulas over Sparse Graphs”, in *35th Computational Complexity Conference (CCC 2020)*, S. Saraf, Ed., ser. Leibniz International Proceedings in Informatics (LIPIcs), vol. 169, Dagstuhl, Germany: Schloss Dagstuhl–Leibniz-Zentrum für Informatik, 2020, 28:1–28:24, ISBN: 978-3-95977-156-6. DOI: `10.4230/LIPIcs.CCC.2020.28`. [Online]. Available: `https://drops.dagstuhl.de/opus/volltexte/2020/12580` (cit. on p. 33)

[FPPR17] N. Fleming, D. Pankratov, T. Pitassi and R. Robere, ”Random ˆ(log n)-CNFs are hard for cutting planes”, in *Proceedings of the 58th Annual IEEE Symposium on Foundations of Computer Science (FOCS ’17)*, Oct. 2017, pp. 109–120 (cit. on p. 16)

[Fre79] G. Frege, *Begriffsschrift, eine der arithmetischen nachgebildete Formelsprache des reinen Denkens*. Halle: Verlag von Louis Nebert, 1879. [Online]. Available: `http://resolver.sub.uni-goettingen.de/purl?PPN538957069` (cit. on pp. 3, 16)

[FSS84] M. Furst, J. Saxe and M. Sipser, ”Parity, circuits and the polynomial-time hierarchy”, *Mathematical Systems Theory*, vol. 17, pp. 13–27, 1984 (cit. on p. 27)

[Göd31] K. Gödel, ”Über formal unentscheidbare Sätze der Principia Mathematica und verwandter Systeme I”, *Monatshefte für Mathematik und Physik*, vol. 38, no. 1, pp. 173–198, 1931 (cit. on p. 4)

[Gri01] D. Grigoriev, ”Linear lower bound on degrees of Positivstellensatz calculus proofs for the parity”, *Theoretical Computer Science*, vol. 259, no. 1–2, pp. 613–622, May 2001 (cit. on pp. 16, 29, 33)

[GHP02] D. Grigoriev, E. A. Hirsch and D. V. Pasechnik, ”Complexity of semi-algebraic proofs”, in *STACS 2002*, Berlin, Heidelberg: Springer Berlin Heidelberg, 2002, pp. 419–430 (cit. on p. 29)

[GP18] J. A. Grochow and T. Pitassi, ”Circuit complexity, proof complexity, and polynomial identity testing: The ideal proof system”, *J. ACM*, vol. 65, no. 6, Nov. 2018, ISSN: 0004-5411. DOI: `10.1145/3230742`. [Online]. Available: `https://doi.org/10.1145/3230742` (cit. on p. 16)

[Hak85] A. Haken, ”The intractability of resolution”, *Theoretical Computer Science*, vol. 39, no. 2-3, pp. 297–308, Aug. 1985 (cit. on p. 16)

[Hås86] J. Håstad, ”Almost optimal lower bounds for small depth circuits”, in *Proceedings of the eighteenth annual ACM symposium on Theory of computing*, ser. STOC ’86, Berkeley, California, United States: ACM, 1986, pp. 6–20 (cit. on p. 27)

[Hås20] ——, ”On small-depth frege proofs for tseitin for grids”, *Journal of the ACM*, vol. 68, pp. 1–31, 2020 (cit. on pp. 16, 28, 29)

[HR22] J. Håstad and K. Risse, ”On bounded depth proofs for tseitin formulas on the grid; revisited”, Accepted to FOCS’22, 2022 (cit. on p. 27)

[Hir18] S. Hirahara, ”Non-black-box worst-case to average-case reductions within np”, in *2018 IEEE 59th Annual Symposium on Foundations of Computer Science (FOCS)*, 2018, pp. 247–258. DOI: `10.1109/FOCS.2018.00032` (cit. on p. 32)

[HP17] P. Hrubeš and P. Pudlák, ”Random formulas, monotone circuits, and interpolation”, in *Proceedings of the 58th Annual IEEE Symposium on Foundations of Computer Science (FOCS '17)*, Oct. 2017, pp. 121–131 (cit. on p. 16)

[IPS99] R. Impagliazzo, P. Pudlák and J. Sgall, ”Lower bounds for the polynomial calculus and the Gröbner basis algorithm”, *Computational Complexity*, vol. 8, no. 2, pp. 127–144, 1999 (cit. on pp. 20, 30)

[IOSS16] D. Itsykson, V. Oparin, M. Slabodkin and D. Sokolov, ”Tight lower bounds on the resolution complexity of perfect matching principles”, *Fundamenta Informaticae*, vol. 145, no. 3, pp. 229–242, Aug. 2016, Preliminary version in *CSR '15* (cit. on p. 16)

[KC00] V. Kabanets and J.-Y. Cai, ”Circuit minimization problem”, in *Proceedings of the Thirty-Second Annual ACM Symposium on Theory of Computing*, ser. STOC '00, Portland, Oregon, USA: Association for Computing Machinery, 2000, pp. 73–79, ISBN: 1581131844. DOI: 10.1145/335305.335314. [Online]. Available: https://doi.org/10.1145/335305.335314 (cit. on p. 32)

[KR96] J. Kleinberg and R. Rubinfeld, ”Short paths in expander graphs”, in *Proceedings of the 37th Annual Symposium on Foundations of Computer Science*, ser. FOCS '96, USA: IEEE Computer Society, 1996, p. 86 (cit. on p. 31)

[KMOW17] P. K. Kothari, R. Mori, R. O'Donnell and D. Witmer, ”Sum of squares lower bounds for refuting any csp”, in *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing*, ser. STOC 2017, Montreal, Canada: Association for Computing Machinery, 2017, pp. 132–145, ISBN: 9781450345286. DOI: 10.1145/3055399.3055485. [Online]. Available: https://doi.org/10.1145/3055399.3055485 (cit. on p. 16)

[Kra19] J. Krajíček, *Proof Complexity*, ser. Encyclopedia of Mathematics and Its Applications. Cambridge University Press, Mar. 2019, vol. 170 (cit. on p. 19)

[KPW95] J. Krajíček, P. Pudlák and A. R. Woods, ”An exponential lower bound to the size of bounded depth Frege proofs of the pigeonhole principle”, *Random Structures and Algorithms*, vol. 7, no. 1, pp. 15–40, 1995, Preliminary version in *STOC '92* (cit. on pp. 16, 27)

[Kri19] M. Krivelevich, ”Expanders – how to find them, and what to find in them”, in *Surveys in Combinatorics 2019*, ser. London Mathematical Society Lecture Note Series. Cambridge University Press, 2019, pp. 115–142. DOI: 10.1017/9781108649094.005 (cit. on p. 31)

[KN19] M. Krivelevich and R. Nenadov, ”Complete Minors in Graphs Without Sparse Cuts”, *International Mathematics Research Notices*, May 2019, rnz086, ISSN: 1073-7928. DOI: 10.1093/imrn/rnz086. eprint: https://academic.oup.com/imrn/article-pdf/doi/10.1093/imrn/rnz086/28672004/rnz086.pdf. [Online]. Available: https://doi.org/10.1093/imrn/rnz086 (cit. on p. 31)

[LN17] M. Lauria and J. 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, 2:1–2:20 (cit. on p. 16)

[LS91] L. Lovász and A. Schrijver, ”Cones of matrices and set-functions and 0-1 optimization”, *SIAM Journal on Optimization*, vol. 1, no. 2, pp. 166–190, 1991 (cit. on p. 30)

[MPW15] R. Meka, A. Potechin and A. Wigderson, ”Sum-of-squares lower bounds for planted clique”, in *Proceedings of the 47th Annual ACM Symposium on Theory of Computing (STOC ’15)*, Jun. 2015, pp. 87–96 (cit. on p. 16)

[MN15] M. Mikša and J. 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, pp. 467–487 (cit. on pp. 16, 29)

[MW15] C. D. Murrayand and R. R. Williams, ”On the (Non) NP-Hardness of Computing Circuit Complexity”, in *30th Conference on Computational Complexity (CCC 2015)*, D. Zuckerman, Ed., ser. Leibniz International Proceedings in Informatics (LIPIcs), vol. 33, Dagstuhl, Germany: Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 2015, pp. 365–380, ISBN: 978-3-939897-81-1. DOI: 10.4230/LIPIcs.CCC.2015.365. [Online]. Available: http://drops.dagstuhl.de/opus/volltexte/2015/5074 (cit. on p. 32)

[PRT22] T. Pitassi, P. Ramakrishnan and L. Tan, "Tradeoffs for small-depth frege proofs", in *2021 IEEE 62nd Annual Symposium on Foundations of Computer Science (FOCS)*, Los Alamitos, CA, USA: IEEE Computer Society, Feb. 2022, pp. 445–456. DOI: 10.1109/FOCS52979.2021.00052. [Online]. Available: https://doi.ieeecomputersociety.org/10.1109/FOCS52979.2021.00052 (cit. on p. 28)

[PBI93] T. Pitassi, P. Beame and R. Impagliazzo, "Exponential lower bounds for the pigeonhole principle", *Computational Complexity*, vol. 3, pp. 97–140, 1993, Preliminary version in *STOC '92* (cit. on pp. 16, 27)

[PRST16] T. Pitassi, B. Rossman, R. Servedio and L.-Y. Tan, "Polylogarithmic Frege depth lower bounds", in *Proceedings of the 48th Annual ACM Symposium on Theory of Computing (STOC '16)*, Jun. 2016, pp. 644–657 (cit. on pp. 16, 27, 28, 30)

[Pot17] A. Potechin, "Sum of squares lower bounds from symmetry and a good story", *CoRR*, vol. abs/1711.11469, 2017. arXiv: 1711.11469. [Online]. Available: http://arxiv.org/abs/1711.11469 (cit. on p. 16)

[Pud97] P. Pudlák, "Lower bounds for resolution and cutting plane proofs and monotone computations", *Journal of Symbolic Logic*, vol. 62, no. 3, pp. 981–998, Sep. 1997 (cit. on p. 16)

[Raz04a] R. Raz, "Resolution lower bounds for the weak pigeonhole principle", *Journal of the ACM*, vol. 51, no. 2, pp. 115–138, Mar. 2004, Preliminary version in *STOC '02* (cit. on pp. 16, 32, 34)

[Raz88] A. A. Razborov, "Bounded-depth formulae over the basis {and, xor} and some combintorial problems (in russian)", *Problems of Cybernetics. Complexity Theory and Applied Mathematical Logic*, pp. 149–166, 1988 (cit. on p. 27)

[Raz98] ——, "Lower bounds for the polynomial calculus", *Computational Complexity*, vol. 7, no. 4, pp. 291–324, Dec. 1998 (cit. on pp. 16, 29, 32)

[Raz01] ——, "Improved resolution lower bounds for the weak pigeonhole principle", Electronic Colloquium on Computational Complexity (ECCC), Technical Report TR01-055, Jul. 2001 (cit. on p. 34)

[Raz03] ——, ”Resolution lower bounds for the weak functional pigeonhole principle”, *Theoretical Computer Science*, vol. 1, no. 303, pp. 233–243, Jun. 2003 (cit. on p. 34)

[Raz04b] ——, ”Resolution lower bounds for perfect matching principles”, *Journal of Computer and System Sciences*, vol. 69, no. 1, pp. 3–27, Aug. 2004, Preliminary version in *CCC ’02* (cit. on pp. 32, 34)

[Raz15] ——, ”Pseudorandom generators hard for k-DNF resolution and polynomial calculus resolution”, *Annals of Mathematics*, vol. 181, no. 2, pp. 415–472, Mar. 2015 (cit. on p. 32)

[Raz17] ——, ”On the width of semialgebraic proofs and algorithms”, *Math. Oper. Res.*, vol. 42, no. 4, pp. 1106–1134, Nov. 2017, ISSN: 0364-765X. DOI: 10.1287/moor.2016.0840. [Online]. Available: https://doi.org/10.1287/moor.2016.0840 (cit. on p. 30)

[Raz22] ——, *Open problems*, 2022. [Online]. Available: https://people.cs.uchicago.edu/~razborov/teaching/index.html (visited on 05/04/2022) (cit. on p. 32)

[Rec75] R. A. Reckhow, ”On the lengths of proofs in the propositional calculus”, Ph.D. dissertation, University of Toronto, 1975. [Online]. Available: https://hdl.handle.net/1807/100390 (cit. on p. 19)

[Rii93] S. Riis, ”Independence in bounded arithmetic”, Ph.D. dissertation, University of Oxford, 1993 (cit. on p. 29)

[Sha49] C. E. Shannon, ”The synthesis of two-terminal switching circuits”, *The Bell System Technical Journal*, vol. 28, no. 1, pp. 59–98, 1949. DOI: 10.1002/j.1538-7305.1949.tb03624.x (cit. on p. 15)

[Sip83] M. Sipser, ”Borel sets and circuit complexity”, in *Proceedings of the fifteenth annual ACM symposium on Theory of computing*, ser. STOC ’83, New York, NY, USA: ACM, 1983, pp. 61–69, ISBN: 0-89791-099-0 (cit. on p. 27)

[Smo87] R. Smolensky, ”Algebraic methods in the theory of lower bounds for boolean circuit complexity”, in *Proceedings of the nineteenth annual ACM symposium on Theory of computing*, ser. STOC ’87, New York, New York, United States: ACM, 1987, pp. 77–82, ISBN: 0-89791-221-7 (cit. on p. 27)

[Tse68] G. Tseitin, ”On the complexity of derivation in propositional calculus”, in *Structures in Constructive Mathematics and Mathematical Logic, Part II*, A. O. Silenko, Ed., Consultants Bureau, New York-London, 1968, pp. 115–125 (cit. on p. 16)

[Tur37] A. M. Turing, ”On computable numbers, with an application to the Entscheidungsproblem”, *Proceedings of the London mathematical society*, vol. 2, no. 1, pp. 230–265, 1937 (cit. on p. 4)

[UF96] A. Urquhart and X. Fu, ”Simplified lower bounds for propositional proofs”, *Notre Dame Journal of Formal Logic*, vol. 37, no. 4, pp. 523–544, 1996 (cit. on p. 27)

[Yao85] A. C. Yao, ”Separating the polynomial-time hierarchy by oracles”, in *Foundations of Computer Science, 1985., 26th Annual Symposium on*, Oct. 1985, pp. 1–10. DOI: 10.1109/SFCS.1985.49 (cit. on p. 27)

Part II

Included Papers

Paper A

On Bounded Depth Proofs for Tseitin Formulas on the Grid; Revisited

Johan Håstad and Kilian Risse

Abstract

We study Frege proofs using depth-d Boolean formulas for the Tseitin contradiction on $n \times n$ grids. We prove that if each line in the proof is of size M then the number of lines is exponential in $n/(\log M)^{O(d)}$. This strengthens a recent result of Pitassi et al. [PRT22]. The key technical step is a multi-switching lemma extending the switching lemma of Håstad [Hås20] for a space of restrictions related to the Tseitin contradiction.

The strengthened lemma also allows us to improve the lower bound for standard proof size of bounded depth Frege refutations from exponential in $\tilde{\Omega}(n^{1/59d})$ to exponential in $\tilde{\Omega}(n^{1/(2d-1)})$.

A.1 Introduction

Mathematicians like proofs, formal statements where each line follows by simple reasoning rules from previously derived lines. Each line derived in this manner, assuming that the reasoning steps are sound, can give us some insight into the initial assumptions of the proof. A particularly interesting consequence is contradiction. Deriving an obviously false statement allows us to conclude that the initial assumptions, also called axioms, are contradictory. We continue the study of Frege proofs of contradiction where each line in the proof is a Boolean formula of depth d. This subject has a long tradition, so let us start with a very brief history.

A very basic proof system is resolution: each line of such a proof simply consists of a disjunction of literals. The derivation rules of resolution are also easy to understand and simple to implement, but the proof system nevertheless gives rise to reasonably short proofs for some formulas. It is far from easy to give lower bounds for the size of proofs in resolution but it has been studied for a long time and by now many strong bounds are known. An early paper by Tseitin [Tse68] defined an important class of contradictions based on graphs that is central to this and many previous papers. For each edge there is a variable and the requirement is that the parity of the variables incident to any given node sum to a particular bit which is called the charge of that vertex. If the sum of the charges is one modulo two this is a contradiction. For a subsystem of resolution, called regular resolution, Tseitin proved exponential lower bounds on refutations of these formulas. After this initial lower bound it took almost another two decades before the first strong lower bound for general resolution was obtained by Haken [Hak85], whose lower bound applied to the pigeonhole principle (PHP). Many other resolution lower bounds followed, but as we are not so interested in resolution and rather intend to study the more powerful proof system with formulas of larger, though still bounded, depth d on each line, let us turn to such proof systems.

The study of proofs with lines limited to depth d dates back several decades. A pioneering result was obtained by Ajtai [Ajt94] who showed that the PHP cannot be proved in polynomial size for any constant depth d. Developments continued in the 1990s and polynomial size proof were ruled out for values of d up to $O(\log \log n)$ for both the PHP [PBI93; KPW95] as well as the Tseitin contradiction defined over complete [UF96] and expander graphs [Ben02].

These developments followed previous work where the computational power of the class of circuits[1] of depth d was studied [Sip83; FSS84; Yao85;

[1] When the depth is small, there is no major difference between circuits and formulas so

Hås86; Raz88; Smo87]. It is not surprising that it is easier to understand the computational power of a single circuit rather than to reason about a sequence of formulas giving a proof. This manifested itself in that while the highest value of d for which strong bounds were known for size of proofs remained at $O(\log\log n)$, the results for circuit size extended to almost logarithmic depth.

This gap was (essentially) closed in two steps. First Pitassi et al. [PRST16] proved superpolynomial lower bounds for d up to $o(\sqrt{\log n})$ and then Håstad [Hås20] extended this to depth $\Theta(\frac{\log n}{\log\log n})$ which, up to constants, matches the result for circuits.

The key technique used in most of the described results is the use of restrictions. These set most of the variables to constants which simplifies the circuit or formulas studied. If done carefully one can at the same time preserve the contradiction refuted or the function computed. Of course one cannot exactly preserve the contradiction and to be more precise a contradiction with parameter n before the restriction turns into a contradiction of the same type but with a smaller parameter, n/T, after the restriction.

The simplification under a restriction usually takes place in the form of a switching lemma. This makes it possible to convert depth d formulas to formulas of depth $d-1$. A sequence of restrictions is applied to reduce the depth to (essentially) zero making the circuit or formula straightforward to analyze. The balance to be struck is to find a set of restrictions that leave a large resulting contradiction but at the same time allows a switching lemma to be proved with good parameters.

In proof complexity the most commonly studied measure is the total size of a proof. There are two components to this size, the number of reasoning steps needed and the size of each line of the proof. In some cases, such as resolution, each line is automatically bounded in size and hence any lower bound for proof size is closely related to the number of proof steps. In some other situation the line sizes may grow and an interesting question is whether this can be avoided.

This line of investigation for Frege proofs with bounded depth formulas was recently initiated by Pitassi et al. [PRT22]. They consider the Tseitin contradiction defined over the grid of size $n \times n$, a setting where strong total size lower bounds for Frege refutations of bounded depth had previously been given by Håstad [Hås20]. If each line of the refutation is limited to size M and depth d, then Pitassi et al. [PRT22] showed that the Frege proof must consist of at least $\exp(n/2^{O(d\sqrt{\log M})})$ many lines. For most interesting

the reader should feel free to ignore this difference.

values of M this greatly improves the bounds implied by the results for total proof size. In particular if M is a polynomial the lower bounds are of the form $\exp(n^{1-o(1)})$, as long as $d = o(\sqrt{\log n})$, in contrast to the total size lower bounds of the form $\exp(n^{\Omega(1/d)})$. Pitassi, Ramakrishnan, and Tan [PRT22] rely on the restrictions introduced by Håstad [Hås20] but analyze them using the methods of Pitassi et al. [PRST16].

We study the same Tseitin contradiction on the grid and improve the lower bounds to $\exp(n/(\log M)^{O(d)})$, a bound conjectured by Pitassi et al. [PRT22]. These bounds are the strongest bounds that can be proved by the present methods and even if we cannot match them by constructing actual proofs we can at least represent the intermediate results of a natural proof by such formulas. We discuss this in more detail below.

A.1.1 Overview of proof techniques

The structure of the proof of our main result follows the approach of [PRT22] but relies on proving much sharper variants of the switching lemma.

In a standard application of a switching lemma to proof complexity one picks a restriction and demands that switching happens to all depth two formulas in the entire proof. Each formula switches successfully with high probability and by an application of a union bound it is possible to find a restriction to get them all to switch simultaneously.

The key idea of [PRT22] is that one need not consider all formulas in the proof at the same time. Rather one can focus on the sub-formulas of a given line. It is sufficient to establish that these admit what is called an ℓ partial common decision tree of small depth. This is a decision tree with the property that at each leaf, each of the formulas can be described by a decision tree of depth ℓ. It turns out that this is enough to analyze the proof and establish that it cannot derive contradiction. The key property is that it is sufficient to only look at the constant number of formulas involved in each derivation step and analyze each such step separately.

The possibility to compute a set of formulas by an ℓ common partial decision tree after having been hit by a restriction is exactly what is analyzed by what has become known as a "multi-switching lemma" as introduced by [Hås14; IMP12]. This concept was introduced in order to analyze the correlation of a small circuits of bounded depth with parity but turns out to also be very useful in the current context.

Even though there is no general method, it seems like when it is possible to prove a standard switching lemma there is good hope to also prove a multi-switching lemma with similar parameters. This happens when going from [Hås86] to [Hås14] and when going from [PRST16] to [PRT22]. We follow the same approach here and this paper very much builds on [Hås20].

We need a slight modification of the space of restrictions and changes to some steps of the proof, but a large fraction of the proof remains untouched. Let us briefly touch on the necessary changes.

The switching lemma of Håstad [Hås20] has a failure probability to not switch to a decision tree of depth s of the form $(As)^{\Omega(s)}$ where A depends on other parameters. As a first step one needs to eliminate the factor s in the base of the exponent. This triggers the above mentioned change in the space of restrictions. This change enables us to prove a standard switching lemma with stronger parameters and, as a warm-up, we give this proof in the current paper. This results in an improvement of the lower bound for total proof size from $\exp(\tilde{\Omega}(n^{1/58d}))$ to $\exp(\tilde{\Omega}(n^{1/(2d-1)}))$. Even though the exponent's exponent is probably still off by a factor of 2, this is a substantial improvement in the parameters.

The high level idea of the proof of the multi-switching lemma is that for each of the formulas analyzed we try to construct a decision tree of depth ℓ. If this fails then we take the long branch in the resulting decision tree and instead query these variables in the common decision tree. A complication that arises is that the answers on the long path in the local decision tree and the answers on a potentially long branch in the common decision tree are different. This causes us to analyze a new combinatorial game on the grid, as defined in Section A.3.1.

A.1.2 Constructing small proofs

Let us finally comment on a possible upper bound; how to construct efficient refutations. If we are allowed to reason with linear equations modulo two then the Tseitin contradiction has efficient refutations. In particular on the grid we can sum all equations in a single column giving an equation containing $O(n)$ variables that must be satisfied. Adding the corresponding equation for the adjacent column maintains an equation of the same size and we can keep adding equations from adjacent columns until we have covered the entire grid. We derive a contradiction and we never use an equation containing more than $O(n)$ variables.

If we consider resolution then it is possible to represent a parity of size m as a set of clauses. Indeed, looking at the equation $\sum_{i=1}^{m} x_i = 0$ we can replace this by the 2^{m-1} clauses of full width where an odd number of variables appear in negative form. Now replace each parity in the above proof by its corresponding clauses. It is not difficult to check that Gaussian elimination can be simulated by resolution. Given linear equation $L_1 = b_1$ and $L_2 = b_2$ with m_1, and m_2 variables respectively, and both containing the variable x we want to derive all clauses representing $L_1 \oplus L_2 = b_1 \oplus b_2$. We have 2^{m_1-1} clauses representing the first linear equation and the 2^{m_2-1}

clauses representing the second linear equation. Now we can take each pair of clauses and resolve over x and this produces a good set of clauses. If L_1 and L_2 do not have any other common variables we are done. If they do contain more common variables then additional resolution steps are needed but these are not difficult to find and we leave it to the reader to figure out this detail. We conclude that Tseitin on the grid allows resolution proofs of length $2^{O(n)}$.

Let us consider proofs that contain formulas of depth d and let us see how to represent a parity. Given $\sum_{i=1}^{m} x_i = 0$ we can divide the variables in to groups of size $(\log M)^{d-1}$ and write down formulas of depth d and size M that represent the parity and the negation of the parity of each group. Assume that the output gate of each of these formulas is an or. We now use the above clause representation of the parity of the groups and get a set of $2^{m/(\log M)^{d-1}}$ formulas of size $mM/(\log M)^{d-1}$ that represent the linear equations This means that we can represent each line in the parity proof by about $2^{n/(\log M)^{d-1}}$ lines of size about M. We do not know how to syntactically translate a Gaussian elimination step to some proof steps in this representation and thus we do not actually get a proof, only a representation of the partial results.

A.1.3 Organization

Let us outline the contents of this paper. We start in Section A.2 and Section A.3 with some preliminaries. In Section A.4 we define the set of restrictions used in the current paper which are almost the same as in [Hås20]. We give some details how decision trees should be modified using local consistency in Section A.5. The important tool for turning switching lemmas to lower bounds for proofs is by something called t-evaluations and we recall this in Section A.6. Next we show how to construct these evaluations and derive our two main theorems assuming the new switching lemmas in Section A.7. The strengthened version of the standard switching lemma is given in Section A.8 and the extension to a multi-switching lemma is presented in Section A.9. Large portions of the proof for the standard switching lemma as well as many definitions are identical to the proof of [Hås20]. We end with some conclusions in Section A.10.

A.2 Preliminaries

We have a graph G which we call "the grid" but to avoid problems at the perimeter we in fact use the torus. In other words we have nodes indexed by (i, j), for $0 \leq i, j \leq n-1$ where n is an odd integer and a node (i, j)

is connected to the four nodes at distance 1, i.e. where one coordinate is identical and the other moves up or down by 1 modulo n. For each node v we have a *charge* α_v and for each edge e in the graph we have a variable x_e. A Tseitin formula is given by a set of linear equalities modulo 2. That is, for each vertex v in G we have

$$\sum_{e \ni v} x_e = \alpha_v.$$

The main case we consider, which we call "the Tseitin contradictions" is when $\alpha_v = 1$ for each v. We do use more general charges in intermediate steps and hence the following lemma from [Hås20] is useful for us.

Lemma A.2.1. *Consider the Tseitin formulas with charges α_v. If $\sum_v \alpha_v = 0$ this formula is satisfiable and has 2^{r_n} solutions where the positive integer r_n depends only on n and not on the value of α_v.*

As a converse to the above lemma, when $\sum_v \alpha_v = 1$ it is easy to see, by summing all equations, that the system is contradictory. In particular the Tseitin contradictions with $\alpha_v = 1$ for all v are indeed contradictions for graphs with an odd number of nodes. We note that each Tseitin formula for the grid graph can be written as a 4-CNF formula by having 8 clauses of length four for each node.

We are interested in proofs in the form of deriving the constant false from these axioms. The exact reasoning rules turn out not to be of central importance but are stated in Section A.6. The important properties of these rules are that they are sound and of constant size.

The sub-formulas that appear in this proof are allowed to contain only $\vee$-gates and negations. We simulate $\wedge$ using $\wedge F_i = \neg \vee \neg F_i$ and we define the depth of a formula to be the number of alternations of $\vee$ and $\neg$.

A.3 Properties of assignments on the grid and some games

We are interested in solutions to subsystems of the Tseitin contradictions. It follows from Lemma A.2.1 that as soon as we drop the constraints at a single node we have a consistent system and indeed many solutions.

On a set X of nodes we say that a partial assignment is *complete* if it gives values to exactly all variables with at least an endpoint in X. The support of a partial assignment α is denoted by $\text{supp}(\alpha)$ and is the set of nodes adjacent to a variable given a value. Note that the support of a complete assignment on X also includes the neighbors of X.

We consider partial assignments that give values to few variables and in particular we are interested in cases where the size of the set X is at most $2n/3$ and hence cannot touch all rows or columns of the grid. Let X^c denote the complement of X.

In this case, X^c contains a giant component containing almost all nodes of the grid. This follows as there are at least $n/3$ complete rows and columns in X^c and the nodes of these rows and columns are all connected. The other, small, components of X^c are important to control as an assignment on X might fail to extend in a consistent way to such a component. To avoid this problem, for a set X we let the *closure of* X, $\mathrm{cl}(X)$ denote all nodes either in X or in small connected components of X^c. Note that $\mathrm{cl}(X)^c$ is exactly the giant component of X^c.

Definition A.3.1. An assignment α with $X = \mathrm{supp}(\alpha)$ is *locally consistent* if it can be extended to a complete assignment on $\mathrm{cl}(X)$ that satisfies all parity constraints on this set.

We extend this definition to say that two assignments are consistent with each other if they do not give different values to the same variable and when you look at the union of the two assignment this gives a locally consistent assignment. The following lemma from [Hås20] is many times useful.

Lemma A.3.2. *Suppose α is a locally consistent assignment where $|\mathrm{supp}(\alpha)| \leq n/2$ and x_e a variable not in $\mathrm{supp}(\alpha)$. Then there is a locally consistent assignment α' that extends α and gives a value to x_e.*

We are interested in complete assignments on some sets X and the grid and in particular how it looks from the outside. Let a *border assignment* be an assignment to the variables with one end-point in X and one outside X. Such an assignment α is *achievable* iff there is an assignment that has the border assignment α and satisfies the parity conditions on X.

Lemma A.3.3. *Let X be a connected set. The a border assignment α is achievable iff the parity of the bits α equals the parity of the size of X.*

Proof. By induction on $|X|$, and the base case when X is a singleton is obvious. For the induction step take any v such that removing v keeps X connected. Of the variables next to v some are forced by the border assignment. Fix the rest of the variables next to v that satisfies the parity constraint at v. Apply induction to X with v removed and the border assignment including the just made assignment to the variables next to v. □

By a simple extension we have the following.

Lemma A.3.4. *Let X be a connected set. The a border assignment α is achievable by a locally consistent assignment iff the parity of the bits α equals the parity of the size of X and this is true also for the border assignment of any small connected component of the complement of X.*

A process that is useful is the following dynamic matching game. We have two players, one adversarial player that supplies nodes while the other, matching player P_M, is supposed to dynamically create a matching that contains the nodes given by the adversarial player. Our strengthened lower bound for the size of a proof uses the same combinatorial lemma as the proof in [Hås20] namely the following.

Lemma A.3.5. [*Hås*20] *When the dynamic matching game is played on the $n \times n$ grid, P_M can survive for at least $n/2$ moves.*

The purpose of this lemma is to find which variables to include in the extended decision tree used. As discussed in the introduction our new lower bound for the number of lines of a proof with short lines needs a multi-switching lemma and it turns out that the decision which variables to include is described by a more complicated combinatorial game. We now discuss this game. The reader that wants motivation for this game is encouraged to first read the proof of the standard switching lemma to find the reason for Lemma A.3.5 and then start reading the proof of the multi-switching lemma.

A.3.1 Another game on the grid

The game is played on the grid between an adversary A and a player P. They take turns picking vertices and edges on the $n \times n$ grid. Once a vertex is picked it can never be picked again. The set of picked vertices is called S. The vertices outside S are called "free". The total number of picked nodes always remains less than $n/2$ and hence there is always a large connected component in the complement of S. Other connected components in the complement are called "small". Some of the picked elements are called "active".

The task of P is to pick as few vertices as possible such that the following properties hold.

1. The number of picked nodes has the correct parity in some special components of S described below.
2. The size of any small component in the complement of S is even.

The game starts with an empty grid, and takes place in rounds where A decides when to start the next round. A can do two types of moves.

1. Pick an arbitrary new vertex v and make it active. This is called a "simple" move.

2. Declare that a round is over. In this case A can make any edge between an active vertex and a free vertex active. Each connected component must have an even number of activated edges leading in to it.

   This second type of move is called a *completion* move. When this move is completed all vertices become inactive and the next round starts.

After a simple move P must pick some vertices to form a connected component of even size jointly with the just placed vertex. P must also make sure that each connected component of the complement is of even size. Any vertex picked by P in response to a simple move becomes active. Note that in this situation P picks an odd number of vertices and hence at least one.

After a completion move P must pick the free vertices with at least one adjacent active edge. It may pick some more vertices to achieve the following.

1. The parity of the size of each connected component of the just picked vertices must equal to the parity of active edges adjacent to it.

2. The number of vertices in any small connected component in the complement is even.

Although this looks complicated please note that if the there is only one active edge going in to the nodes P must pick and these do not split any connected component of the complement, then these forced nodes is all that P needs to select.

What forces P to act in general is the creation of small odd size components in the complement of S due to making the "obvious" choices. For any such component C, P needs to add vertices to S to make it of even size. It is also restricted to only adding vertices adjacent to a supplied starting vertex. This vertex is in S but connected to at least some vertex in C. We call this "evenizing" with starting point w. All connected components of the complement created in this process must be made of to be of even size. It is simple to see that this can always be done, simply add any vertex adjacent to w. If this does not split C in to at least two components then P is done. Otherwise P can simply recurse on any created component of odd size with the chosen vertex as the starting point. We must prove that, over the course of the entire game, A cannot force P to add too many vertices.

To get some understanding of the problems, let us first give an example where P is forced to make many moves.

Example. Suppose C consists of the vertices $(1, x)$ for $1 \leq x \leq t$ jointly with $(2, x)$ for even x at most $t-1$ and $(0, x)$ for odd x at most $t-1$. This has an odd number of vertices (in fact $2t-1$) and suppose the starting vertex is $(1, 0)$. P needs to add $(1, 1)$ since this is the only vertex in C connected to $(1, 0)$. This creates the isolated vertex $(0, 1)$ and a component of size $2t-3$ that is very similar to the starting component. It is easy to see that P ends up picking all vertices of this component.

We set up a potential function to prove that such massive responses as in the example can only happen rarely. For each connected component of the complement consider its edges to elements of S. For each edge to an active vertex we assign four points and for each other edge one point. Suppose the total number of points for component C_i is f_i and this number is called the *score* of C_i. We have a parameter T and we say that each component of size at most T is *ultra small*. We later fix T to a suitable constant. A component that is not ultra small is called *sizeable*. This includes the large component. We now define the potential as

$$\sum_i f_i + G - D(F-1)$$

where the sum is over components that are sizeable, F the number of components that are sizeable, D is a constant to be chosen suitably, and G is the number of ultra small components. For G we only count a component the first time it becomes ultra small. Further splitting of an ultra small component is ignored. The reason for using $(F-1)$ is that we want to start the potential at 0 and hence not count the large component in this number.

We want to prove that this potential increases by at most a constant for each simple move and decreases by at least one half for other moves. By setting T large enough (after we have chosen D) we make sure that $f_i \geq 2D$ for any component of size at least T.

Let us first analyze simple moves. When A chooses a vertex it might increase $\sum_i f_i$ by at most 16. This might also cause a component of the complement of S to split. To analyze the cost of such a split we first pay the increase by the addition of the extra vertex to S in the form of increase to f_i. We then see how the splitting of a component of the complement affects the potential. First note that splitting an ultra small component does not affect the potential (remember that we do not count this as an increase in G) and thus we are interested in splitting a sizeable component. We have sequence of simple lemmas.

Lemma A.3.6. *If a sizeable component splits into ultra small components then the potential decreases by at least* $D-4$*.*

Proof. Suppose C_i splits. This means that the term $f_i - D$ disappears in the potential. By construction this is at least D. We might have an increase of G by 4 but no other increase. The lemma follows. □

Next we have.

Lemma A.3.7. *If a sizeable component splits and the result contains at least two sizeable components then the potential decreases by at least* D*.*

Proof. The creation of a sizeable component increases F. Any ultra small component created increases G by one but at the same time its score is removed from the sum causing a decrease of that sum by at least 4. □

Finally we analyze the third possibility.

Lemma A.3.8. *If a sizeable component splits into a sizeable component and one or more ultra small components the potential decreases by at least three for each component split off.*

Proof. The value of F does not change. Any ultra small component created increases G by one but its score of at least 4 is removed from the sum $\sum_i f_i$. □

The above lemmas imply that the splitting of components only decreases the potential. What remains is to analyze the cost when P is forced to evenize an odd size component. By "cost" we here mean increase in potential. We might have a negative cost which is a decrease in potential.

Lemma A.3.9. *The cost of evenizing a component with an active starting point is at most* $11 - m/2$ *where* m *is the number of moves made by* P *in sizeable components. The cost of evenizing an ultra small component is* 0*.*

Proof. We prove the lemma by induction over the size of the component. If the component is ultra small then no term of the potential can change so there is no cost.

As a first attempt let P pick an arbitrary vertex, v, next to the starting which we call w. If this does not result in any new odd size component we are done. We have added at most three more edges at cost four each while eliminating the cost of (v, w). As a result f_i might have increased by at most 8 giving the same increase in the potential and P has made one move in sizeable component.

Now suppose that choosing v creates some new odd size components that have to be evenized. We know that this number must be even and since any component has to be adjacent to v and since v has at most three neighbors other than w there must be exactly two such components and call them C_1 and C_2. Let v_i be an element in C_i that is a neighbor of v. Suppose the scores of these two components are f_1 and f_2 and the score of the component that splits is f. Note that f is measured before v is placed in S while f_1 and f_2 are measured after this has happened and thus we need to keep track of what happens to edges next to v. One fact to our advantage is that while (v, w) was counted in f its four points do not appear in neither f_1 or f_2.

There are a number of cases depending on the status of the fourth neighbor of v (on top of w, v_1 and v_2). It can be in a third, new, component, be an element of S, or belong to C_1 or C_2. In the first case that third component is of even size and hence need not be evenized. If it is sizeable we get a decrease in potential of at least D and if it is ultra small by at least three. In either case we are doing strictly better then if this node belongs to S and hence we can ignore this case and we may assume that we get exactly two new components.

If neither of these two new components is ultra small, then the potential decreases by Lemma A.3.7. The total increase in the score is bounded by 8 as we eliminate the score of (v, w) and add at most 3 new edges with three points each. The cost, by induction, to evenize C_1 and C_2 is at most $22 - (m_1 + m_2)/2$ where m_i is the number of moves of P made in sizeable components when evenizing C_i. The total change to the potential is thus bounded from above by $30 - (m_1 + m_2)/2 - D$ and making sure that $D \geq 20$ the lemma follows in this case.

Now suppose C_1 and C_2 are both ultra small. Then G increases by two but we have a decrease of D in potential by Lemma A.3.6 and in this case we in fact have a total decrease in the potential and no more moves in sizeable components. This establishes the lemma in this case.

Finally assume that C_1 is ultra small while C_2 is not. In this case we get an increase of G by one while F does not change. We need to analyze the change in scores and the cost of possible recursive calls.

If the fourth neighbor of v (on top of v_1, w and v_2) does not belong to C_2 then $f_2 \leq f - 3$. This follows as the only new edge in C_2 that was not present in C is (v_2, v) but this is compensated by (v, w) being present in C but not in C_2. On top of this at least three points have disappeared from f when forming C_1. For the recursive costs we have that, by induction, the cost to evenize C_2 is at most $11 - m_2/2$. As C_1 is ultra small we have no cost for its recursive call. The net cost is thus bounded from above by $1 - 3 + 11 - m_2/2 \leq 10 - m/2$ and the lemma follows also in this case.

Finally consider the case when the missing neighbor of v, call it v_2',

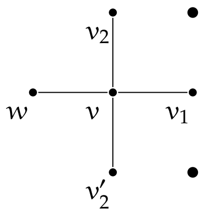


Figure A.1: The larger circles are elements of S

belongs to C_2. This causes the potential addition of 4 to f_2 by the edge (v, v_2') and this needs to be addressed. Unfortunately this leads to a rather tedious case analysis.

Suppose without loss of generality that w is to the left of v. We first have three cases whether v_1 is to right, above, or below v. The cases above and below are symmetric so in fact we can drop the case of v_1 being below v.

Let us assume that v_1 is to the right of v and v_2 above and v_2' is below. The situation looks like in Figure A.1, where we note that the vertices to the right of v_2 and v_2' must be in S to make removing v disconnect C_1 and C_2.

Now suppose that we can remove v_2 from C_2 and keep it connected. Then P can pick v and v_2 and the remaining part of C_2 is even and there is no recursive call. Let us compare f_2 and f. We have lost at least 3 points from f that now belongs to C_1. We also lost 4 points from (w, v) becoming internal of S. We gain 4 points from (v, v_2'). Finally we can have two new edges next to v_2 (going left and up). There is a net gain in potential of at most 5 and the lemma follows also in this case as P only made two moves.

The case when we can remove v_2' and keep C_2 is connected is symmetric and hence we need to consider the case when both create new components and thus we can assume that both removing v_2 and v_2' splits C_2. The two components that C_2 splits to when v_2 is removed must then be connected to v_2 from top and from the left and for v_2' the two components attach from left and below.

Put both of v_2 and v_2' into S. Then C_2 splits in to three components, two of which might have to be evenized. If two of these are sizeable then we have increased F by one. The analysis is very much as before and the extra decrease of D provided by Lemma A.3.7 well compensates for the two recursive evenizing calls.

The case when C_2 splits into three ultra small components is also very similar to previous cases. There is no recursive call and Lemma A.3.6 provides a large decrease. The case that remains to analyze is that we have

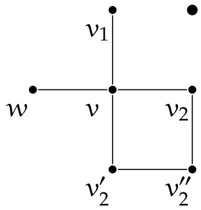


Figure A.2: The larger circle is an elements of S

exactly two ultra small components.

We have (remember we also have C_1) created three ultra small components. Each decreases the score by at least three and increase G by three for a net decrease of 6. The edge (v, w) is now interior to S while it is counted in f. The only new edges in f_2 are one from each of v_2 and v_2'. Thus we have net decrease of 2. We still have one recursive call on the remaining component that is sizeable but this costs, by induction, at most $11 - m/2$ where m is the number of nodes chosen by P in this recursive call. The lemma follows in the case when both v_2 and v_2' disconnect C_2.

We have the final case when v_2 is to the right of v and v_2' is below. Suppose first that adding either v_2 or v_2' to S does not disconnect C_2. Then if P removes this vertex and v and there is no recursive call. Suppose it removes v_2' (the case of v_2 being similar). Then we might get three new edges costing four points next to v_2' and the edge (v, v_2) is of the same cost while the cost of (v, w) disappears. At least three points disappear with the creation of C_1 while there is an increase of one for G. This implies that there is an increase of at most 10 and as P has picked two vertices the lemma follows in this case.

We need to analyze the situation when both removing either v_2 and v_2' disconnects C_2. Let us first observe that the vertex v_2'' in the picture belong to C_2 since otherwise removing v_2 does not disconnect C_2. The situation is like in Figure A.2.

Now, consider putting all four vertices v, v_2 v_2', and v_2'' in S. This splits C_2 into a number of components as we may have one component hanging off each side of the square. If at least two are sizeable we get an increase in F and Lemma A.3.7 takes care of of the local costs and we can apply induction. Similarly if all components are ultra small Lemma A.3.6 tells us that there is a decrease in potential. We need to analyze what happens when exactly one component is sizeable.

In fact we must have three ultra small components hanging off the square each giving a net decrease of at least 2. Indeed we have C_1 and the ultra small components created when v_2 and v_2' are removed. Since we have a sizeable component we must have one component hanging off each of the four sides of the cube, as we cannot have two components attaching to the same side.

We only have one recursive call with a cost of $11 - m/2$, and we have net decrease in 6 from the ultra small components. Finally for edges, we do not any more count the cost of (v, w) and we can only have two new edges entering the component of the recursive call. The new edges to the ultra small components do not count. Thus apart from the recursive call we have a net decrease of 2 and this compensates for the four points added by P. □

The above takes care of all simple moves. Let us look at completion moves.

Lemma A.3.10. *A completion decreases the potential by at least the number of active edges chosen by A. This includes the forced response by P*

Proof. The first that happens is that an edge which costs 4 is replaced by an inactive vertex next to it. This results in at most three edges of cost one and is hence a decrease of at least one in potential. If several active edges go to the same vertex P has to add two vertices but this gives a decrease of at least two. Now unless this causes a split of a component we are done.

If it splits an ultra small component then there is no further change in the potential. If it splits a sizeable component then we might have to evenize two components and the following lemma is what we need.

Lemma A.3.11. *The cost of evenizing a component with an inactive starting point is at most $3 - m$ where m is the number of vertices added by P to sizeable components. The cost of evenizing an ultra small component is at most 0.*

Let us assume this lemma then finish the proof of Lemma A.3.10. As many times previously unless we get exactly one non ultra small component it is easy to prove that there is a decrease so assume that this is the case. Each ultra small component decreases the potential by a least three and this is sufficient to pay for the evenizing of the component and this is demonstrated by Lemma A.3.11. □

The proof of Lemma A.3.11 is surprisingly much simpler than the proof of Lemma A.3.9. The key difference is that new edges added only cost one and not four. This makes it much easier to compensate the cost of new edges by the loss in potential due to the appearance of ultra small components.

Proof of Lemma A.3.11. If the response of putting a vertex, v, next to the starting vertex is sufficient then we have $m = 1$ and the potential increases by at most 2 as three edges are added and one is removed. The lemma is thus true in this case and let us analyze what happens to the potential if v causes the component of the complement to split. As before, unless it is a sizeable component that splits and the result is exactly one sizeable component and one or more ultra small components, we do have a substantial decrease in the potential due to the loss of a term D.

As in the previous proof the worst case is when v splits the component into two components C_1 and C_2 where the first is sizeable and the the second is ultra small and the third neighbor of v belongs to C_1. In this case we have added two more edges of v into C_1. We have removed one edge (between v and the starting point) and lost the cost of at least three edges that are now part of C_2. This is a net loss of two to the potential. We need to evenize C_1 and this cost by induction at most $3 - m_1$ if P picks m_1 vertices in this process. Finally we have one more ultra small component and thus the total cost is at most $2 - m_1$. Since P picks $m_1 + 1$ vertices in total, the lemma follows. □

We finally state the conclusion of this section.

Lemma A.3.12. *If A makes s simple moves in the game, then the total number of moves is bounded by $O(s)$.*

Proof. The potential increases by $O(1)$ for each simple move of A. The evenizing of any odd component created costs at most $O(1)$ but is decreased by $1/2$ for any vertex chosen by P is a sizeable component. We conclude that the total number of moves in sizeable components is at most $O(s)$.

As the number of ultra small component created is bounded by the potential, their number is $O(s)$. In each such component there are $O(1)$ moves. □

A.4 Restrictions

We use (essentially) the same space of random restriction as [Hås20]. The only difference is the choice in the number of live centers in the partial restrictions. This is the parameter k below which changes its value from $Cs(n/T)^2$ to $C \log n(n/T)^2$. For completeness we repeat all definitons from [Hås20] but we keep the description brief and for intuition and motivation we refer to [Hås20].

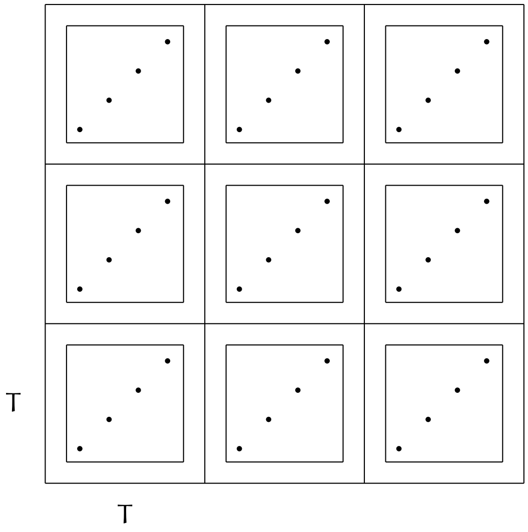


Figure A.3: The centers and central areas

A.4.1 Full restrictions

In an $n \times n$ grid we make sub-squares of size $T \times T$ where T is odd. In each sub-square we choose[2] $\Delta = \sqrt{T}/2$ of the nodes and call them *centers*. These are located evenly spaced on the diagonal of the $3T/4 \times 3T/4$ central sub-square. This implies that they have separation $3\sqrt{T}/2 = 3\Delta$ in both dimensions. A schematic picture of this is given in Figure A.3.

The centers in neighboring sub-squares are connected by paths that are edge-disjoint except close to the endpoints. Let us describe how to connect a given center to a center in the sub-square on top. As there are $T/4 = \Delta^2$ rows between the two central areas, for each pair of centers (the jth center, c_j in the bottom sub-square and ith center c'_i in the top sub-square) we can designate a unique row, r_{ij} in this middle area.

To connect c_j to c'_i we first go i steps to the left and then straight up to the designated row r_{ij}. This is completed by starting at c'_i and then going j steps to the right and down to the designated row. We finally use the appropriate segment from the designated row to complete the path (which might be in either direction). A picture of this is given in Figure A.4. We index the centers from 1 to Δ and hence each path consists of 5 non-empty

[2] For simplicity we assume that some arithmetical expressions that are supposed to be integers are in fact exact as integers. By a careful choice of parameters this can be achieved but we leave this detail to the reader.

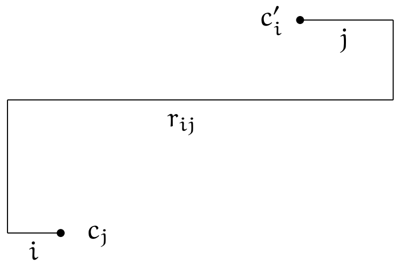


Figure A.4: A path

segments. The first and last segments are totally within the central area while the middle segment is totally in the area between the central areas. Segments two and four go from the central areas to the area in-between.

Connecting c_j to a center c'_i in a sub-square to the left is done in an analogous way. There is a unique column c_{ij} reserved for the pair and the path again consists of five non-empty segments. The first and last segments consist of i vertical edges up from c_j, and j vertical edges down from c'_i. We add horizontal segments connecting to the designated column c_{ij} the and middle segment is along this column. The below lemma is proved in [Hås20].

Lemma A.4.1. *The described paths are edge-disjoint except for the at most Δ edges closest to an endpoint. For each edge e, if there is more than one path containing e, these paths all have the same endpoint closest to e.*

We let the term *closest endpoint* of an edge denote the closest endpoint breaking ties in an arbitrary way. The key property we need is that the "closest endpoint" of a path through an edge is uniquely defined by the edge.

We define the *direction* of a path to be the relative positions of the sub-squares of its two endpoints. It is true that the paths are undirected but at times when we consider paths from a fixed center v it is convenient to think of such paths as starting at v and thus speak of paths going left or right from v rather than sideways. We note that apart from having the same closest endpoint, all paths through one fixed edge e have the same direction.

A restriction is defined by first choosing one center in each $T \times T$ sub-square and then the paths described above connecting these centers. Note that these paths are edge-disjoint. The chosen centers naturally form a $n/T \times n/T$ grid if we interpret the paths between the chosen centers as edges.

We proceed to make the correspondence more complete by assigning values to variables.

We choose a solution to the Tseitin formula with charges 0 at the chosen centers and 1 at other nodes. As the number of chosen centers is odd, by Lemma A.2.1, there are many such solutions. For variables not on the chosen paths these are the final values while for variables on the chosen paths we call them *suggested* values.

For each path P between two chosen centers we have a new variable x_P and for each variable x_e on P it is replaced by x_P if the suggested value of x_e is 0 and otherwise it is replaced by $\bar{x}_P$.

We claim that with these substitutions we have reduced the Tseitin problem on an $n \times n$ grid to the same problem on an $n/T \times n/T$ grid. This is true in the sense that we have an induced grid when we interpret paths as new edges and we need to see what happens to the axioms.

Given a formula F we can apply a restriction σ to it in the natural way resulting in a formula denoted by $F\lceil_\sigma$. Variables given constant values are replaced by constants while surviving variables are replaced by the appropriate negation of the corresponding path-variable. A restriction has a natural effect on the Tseitin contradiction as follows.

- The axioms for nodes not on a chosen paths are all reduced to true as all variables occurring in them are fixed in such a way that the axioms are true.
- The axioms for interior nodes of a chosen path are reduced to tautologies as the axiom is true independent of the value of the involved variable(s) x_P. This is true as flipping a single x_P changes the value of two variables next to any such node.
- The axioms at the chosen centers turn into the axioms of the smaller instance.

These just defined restrictions are called *full restrictions* and a typical full restriction is denoted by σ. Note that these full restrictions are really "affine restrictions" in the vocabulary of [RST15] as they do not only assign values to variables but also identify several old variables with the same new variable that might also be negated. For simplicity, however, we keep the simpler term "restrictions".

A.4.2 Partial restrictions and pairings

A typical partial restriction is called ρ and as we mostly discuss partial restrictions we simply call them "restrictions" while we use the term "full restrictions" when that is what we have in mind. At the same time as describing partial restrictions we give a probability distribution on such restrictions.

Let k be an odd integer of the form $C \log n(n/T)^2$ for a constant C to be determined. The first step of constructing ρ is picking k centers uniformly at random from the set of all $\Delta(n/T)^2$ centers defined in the previous section. These are the *alive* centers. In the future we only consider the case when the number of live centers in each sub-square is between a factor .99 and 1.01 of its expected value $C \log n$. By choosing C appropriately the probability of this being false is can be made to be $1/n$.

We define charges that are 0 for all live centers and 1 for dead centers. As the number of live centers is odd we can apply Lemma A.2.1 and pick a random solution with these charges to the Tseitin formula. For edges not on paths between live centers these are final values while for variables on such paths we call them *preferred* values.

The choice of the centers together with the fixed and preferred variables is denoted by ρ. The choice of ρ is the main probabilistic event. Note that by Lemma A.2.1 the number of possible values for fixed and preferred values is independent of which centers are alive and even of k as long as it is odd.

We now describe how to turn a partial restriction ρ into a full restriction σ. Choose one center to survive in each sub-square[3]. These are called the *chosen centers* and paths between such centers correspond to the variables that remain and are called *chosen paths*. Centers that were alive through the first part of the process but are not chosen are called *non-chosen*. The centers killed already by ρ are simply called dead. We proceed to define a pairing.

Definition A.4.2 (pairing)**.** A *pairing* π is a graph supported on the non-chosen centers. Each component of π is either a single edge or a star of size four with one center and three nodes of degree one. Connected centers are located in adjacent sub-squares.

The following lemma follows from the proof of the corresponding lemma from [Hås20] which had the paramter s instead of $\log n$.

Lemma A.4.3. *If each sub-square has between* $.99C \log n$ *and* $1.01C \log n$ *non-chosen centers, a pairing* π *exists.*

Let us establish some notation. As the original grid is also a graph with edges we from now on use the term "grid-edges" to refer to edges in the original grid. The chosen centers form a smaller grid and this also

[3] This choice can be done in an arbitrary way but to be definite let us choose the center from the lowest numbered row.

has edges and we call these “new grid-edges”. We only consider paths in the original grid and we keep the shorter term “path” for these. In other words, from now on an “edge” is a connection between two live centers and corresponds to a path in the grid-graph. A “new grid-edge” corresponds to a chosen path and is thus also an edge in the graph of the live centers. We say that two chosen centers are neighbors if they are in adjacent sub-squares.

Some edges are conflicting in that we do not allow them to be present in the graph at the same time. More precisely we allow at most one path in each of the four directions from a center. As picking a path corresponds to changing the variables on this path this is the same as saying that the variables can only change values at most once.

As stated above π makes it possible to turn ρ into σ. Variables not on live paths take their fixed values. Variables on live paths but not on chosen paths take their preferred values unless they are on a path chosen by π in which case these values are negated. On the chosen paths, the preferred values now becomes suggested and this completes the description of σ. Thus σ is obtained deterministically from ρ and π and when we want to stress this dependence we sometimes write $\sigma(\rho, \pi)$.

We use the term “preferred values” as a vast majority of the variables will eventually be fixed to these values as very few variables appear on the paths of π or turn into suggested values. On the other hand “suggested values” are much less certain as the path-variables should be thought of as equally likely to be 0 and 1 and thus these variables are equally likely to take also the non-suggested value.

As an intermediate between ρ and the full restriction σ we have ρ and some information in the form of existence or non-existence of edges. We have the following definition.

Definition A.4.4 (information piece)**.** An *information piece* is either in form of an edge (v, w) for two centers v and w or $(v, \delta, \perp)$ where v is a center and δ is a direction (i.e. “left”, “right” “up” or “down”). The former says that there is an edge from v to w while the latter says that there is no edge from v in the direction δ.

We note that, as edges are undirected, (v, w) and (w, v) denote the same information. We also use sets of information pieces.

Definition A.4.5 (consistent information set)**.** An *information set* I is a collection of information pieces. Its *support*, denoted by $\mathrm{supp}(I)$, is the set of centers mentioned in these pieces. An information set I is *consistent* if

1. it does not have two different pieces of information from the same center in one fixed direction, and

2. if I has information in all four directions from a center v then it has an odd number of edges touching v.

A partial assignment to some path-variables naturally corresponds to a set of information pieces. An assignment of 0 to a path-variable gives two non-edges, in the appropriate directions, with closest end-points at the two chosen centers connected by this path. An assignment of 1 gives an information piece in the form of an edge between the two chosen centers. We use the term "consistent" both for sets of information pieces and partial assignments. Consistency for assignments requires an odd number of ones adjacent to any center that has all its variable assigned and this exactly corresponds to the property of information pieces in all four directions in the definition above. This makes the two notions close and hence using "consistent" for both should hopefully not confuse the reader.

Jointly with ρ an information set fixes the values of some more variables as follows.

Definition A.4.6 (forcing). Let ρ be a restriction and I an information set. A variable x_e is considered *forced by* (ρ, I) iff either its closest endpoint, v, is not live in ρ or if the information of v in the direction of e is contained in I. It is forced to its preferred value in ρ unless the relevant information piece states that there is an edge from v in the direction of e that corresponds to a path that passes through e in which case it takes the opposite value. Variables not on live paths take the value given by ρ.

There are other situations where the value of a variable might be determined by ρ and I, such as the lack, or scarcity, of live centers in a sub-square but we do not use this information in the reasoning below. We need the notion of a closed information set.

Definition A.4.7 (closed information set). An information set I is *closed* if I is consistent and for each $v \in \mathrm{supp}(I)$, the set I contains the information in all four directions.

The definition implies that for any $v \in \mathrm{supp}(I)$, in any direction δ where there is not an element of $\mathrm{supp}(I)$, we have a non-edge $(v, \delta, \perp)$. When considered as a graph such an information set is an odd-degree graph (with degrees one and three) on the centers of $\mathrm{supp}(I)$.

Note that if we have a closed information set I then if we consider all variables forced by (ρ, I) this can be described by a restriction where the centers in the $\mathrm{supp}(I)$ are killed. We simply negate the values of any preferred variable on any path in I and then forget that the centers in $\mathrm{supp}(I)$ were alive.

Thus, if we let such a closed information set operate on a restriction ρ we get a restriction with fewer live centers where the number of killed centers is exactly the number of centers in the support of the corresponding graph.

A.4.3 Generalized restrictions

In our proof we allow generalized (partial) restrictions. These are like standard restrictions but we allow the violation of the Tseitin condition at some dead centers. Such centers are called *bad* and we keep a close track on their number. These generalized restrictions are only used for book-keeping reasons.

A.5 Decision trees

We have decision trees where each internal node is marked with a variable and the outgoing edges are marked with 0 and 1. The leaves of a decision tree are labeled by 0 and 1. We allow decision trees of depth 0 which are constants 0 or 1.

All decision trees considered in this paper have a depth that is smaller than half the dimension of the grid we are currently considering. For each branch in a decision tree there is minimal partial assignment, τ such that any extension of this partial assignment creates an assignment that follows this path. We use this τ to identify that branch and we call it *consistent* if τ is consistent in the sense of Definition A.3.1.

We trim decision trees to maintain the property that all branches of a decision tree are consistent. When a decision tree is created this is not a problem but trimming takes place when we consider what happens under a partial assignment τ or a full restriction σ. In that latter case, the initial decision tree uses the variables x_e while the resulting decision tree uses the new variables x_P.

We sometimes think of a decision tree T as the set of all branches leading from the root to the leaves. These have labels and fit together in a tree structure and each corresponds to a partial assignment τ' as discussed above. When creating the decision tree after τ or σ the idea is to keep all branches that are consistent with the new information.

In the case of a partial assignment τ we keep all branches corresponding to τ' such that τ and τ' are consistent as discussed after Definition A.3.1. In the case of a full restriction σ the situation is not difficult but slightly more complicated so let us define this explicitly.

The assignment τ' assigns values to some variables x_e. Some of these are given values by σ while the rest are now on chosen paths. To be consistent we require that for the variables given values by both σ and τ', the two values agree. For each variable x_e given a value by τ' we get a value for the corresponding path-variable x_P. For σ and τ' to be consistent we require that no x_P gets two conflicting values and that the values x_P are consistent in the sense of Definition A.3.1 when considered as an assignment on the smaller grid.

The key property that we need is that if the depth of T is small enough then at least some branch of T is consistent with τ or σ. In the former case we make sure that the total number of assigned variables under τ and τ' is at most half the dimension of the grid and in the latter case that the depth is a most half the dimension of the grid after σ. This together with the fact for each internal node of T has out-degree two and Lemma A.3.2 makes sure that some branch is consistent.

Once we have identified which branches remain it is easy to see that they form a decision tree. In fact it is also possible to define the new decision tree by a dynamic process where we start at the root of T and consider each node in the tree. As we walk down the tree we can, for each node, check whether both values of the current variable are consistent with the partial assignment of the branch so far jointly with τ or σ. For a full restriction σ we of course take into account that once we have determined the value on one variable on a path, all the other variables on the same path are determined. If only one value is consistent we eliminate the other sub-tree while if both values are consistent we have found a node in the new tree. In some situations we might get a tree which has a single branch consistent with τ or σ. This is considered a depth-0 tree with only one leaf. For a decision tree T we let $T\lceil_\tau$ we the decision tree after we have applied τ.

We let a *1-tree* be a decision tree where all leaves are labeled 1 and define a *0-tree* analogously. Special cases of such trees are trees of depth 0. We say $T\lceil_\pi = b$ if the decision tree given by $T_i\lceil_\pi$ is a b-tree.

We say that a decision tree $\mathcal{T}$ is an ℓ common partial decision tree for $T_1, \ldots, T_m$ of depth t if

1. $\mathcal{T}$ is of depth t, and

2. for every T_i and branch π in $\mathcal{T}$ there are decision trees $T(i, \pi)$ of depth ℓ such that the following holds. Let $\mathcal{T}_i$ be the decision tree obtained from $\mathcal{T}$ by appending the trees $T(i, \pi)$ at the corresponding leaf π of $\mathcal{T}$. Then, if a branch π' in $\mathcal{T}_i$ ends in a leaf labeled b, it holds that $T_i\lceil_{\pi'} = b$.

Next we turn to a procedure of representing formulas by decision trees of small depth.

A.6 Basics for t-evaluations

The concept of t-evaluations was introduced by Krajíček et al. [KPW95] and is a very convenient tool for proving lower bounds on proof size. The content of this section is standard and we follow the presentation of Urquhart and Fu [UF96] while using the notation of Håstad [Hås20]. We need a generalization of previous notions essentially as introduced by Pitassi et al. [PRT22].

A tree T represents $T_1 \vee \ldots \vee T_s$ if for every branch π of T ending in a leaf labeled 1 it holds that there is an $i \in [s]$ such that $T_i\lceil_\pi = 1$, and if π ends in a leaf labeled 0, then for all $i \in [s]$ it holds that $T_i\lceil_\pi = 0$. The set of formulas Γ has a t-evaluation φ, mapping formulas from Γ to decision trees of depth at most t, if the following holds.

1. φ maps constants to the appropriate decision tree of depth 0,

2. axioms are mapped by φ to 1-trees,

3. if $\varphi(F) = T$ then $\varphi(\neg F)$ is a decision tree with the same topology as T but where the value at each leaf is negated, and

4. if $F = \vee_{i\in[s]} F_i$, then $\varphi(F)$ represents $\vee_{i\in[s]}\varphi(F_i)$.

Each line of a proof has its own t-evaluation. In order to argue about the proof we need that these different t-evaluations are consistent, as explained next.

Let us first define what it means for decision trees to be consistent. Two decision trees T_1, T_2 are consistent if for every branch π of T_1 ending in a leaf labeled b it holds that $T_2\lceil_\pi = b$ and vice-versa. Further, T_1 and T_2 are $\neg$-consistent, if for every branch π of T_1 ending in a leaf labeled b, it holds that $T_2\lceil_\pi = \neg b$ and vice-versa.

Let us say that two formulas are isomorphic if they only differ in the order of the or's, and let us say that two formulas $F, G = \neg G'$ are $\neg$-isomorphic if F and G' are isomorphic.

Consider a t-evaluation φ defined over a set of formulas Γ and similarly let φ' be a t-evaluation defined over the set of formulas Γ'. The two t-evaluations φ and φ' are consistent if

1. for all isomorphic formulas $F \in \Gamma$ and $F' \in \Gamma'$ it holds that $\varphi(F)$ and $\varphi'(F')$ are consistent, and

2. for all $\neg$-isomorphic formulas $F \in \Gamma$ and $F' \in \Gamma'$ it holds that $\varphi(F)$ and $\varphi'(F')$ are $\neg$-consistent.

We say that a Frege proof has a t-evaluation if for every line ν in the proof we have a t-evaluation φ^{ν} and for all lines ν, ν' it holds that φ^{ν} and $\varphi^{\nu'}$ are consistent.

Let us consider a Frege proof of depth d and for a line ν in the proof let Γ^{ν} be the set of subformulas occuring on line ν. In the following we construct a sequence of restrictions $\sigma_1, \sigma_2, \ldots, \sigma_d$ such that for every line and all formulas of depth at most k we have consistent $t(k)$-evaluations if the formulas are hit by the concatenation σ_k^* of the first k restrictions in the sequence. When considering proof size we in fact have that all $t(k)$ are equal to the same value t, while in the proof when we lower bound the number of small lines, the value $t(k)$ grows as a function of k. In fact, in the latter situation, each line has a common part to all decision trees of that line and this common part increses in size with k.

Getting back to $t(k)$-evaluations, put different we build by induction on k for every line ν a $t(k)$-evaluation for all formulas in

$$\Gamma_k^{\nu} = \{F\lceil_{\sigma_k^*} \mid F \in \Gamma^{\nu} \wedge \mathsf{depth}(F) \leq k\}$$

that are pairwise consistent and we look to extend these $t(k)$-evaluations to Γ_{k+1}^{ν}. To make sure that the domain of the t-evaluations does not decrease when we apply a restriction we use the lemma below from [Hås20]. The fact that we allow consistent $t(k)$-evaluations, instead of a single $t(k)$-evaluation for the entire proof, does not change the proof which is a simple and fairly formal verifiction and hence omitted.

Lemma A.6.1. *Let φ and φ' be two consistent t-evaluations respectively defined on the set of formulas Γ and Γ', and let σ be a full restriction whose output is a grid of size n. Then, provided that $t < n/4$, $\varphi(F)\lceil_{\sigma}$ and $\varphi'(F)\lceil_{\sigma}$ are consistent t-evaluations whose domain includes $\Gamma\lceil_{\sigma}$, and $\Gamma'\lceil_{\sigma}$ respectively.*

The important step of the argument is to use a switching lemma to extend the domain of the $t(k)$-evaluation from Γ_k^{ν} to Γ_{k+1}^{ν}. We give that argument in the next section and here we turn to formulating the punch line once we have a $t(k)$-evaluation for a small Frege proof, where we think of $t(k)$ as small.

It turns out that under these assumptions all lines in the proof are represented by 1-trees. As the the constant false is represented by a 0-tree we can thus not derive the desired contradiction. Hence in order to obtain the desired contradiction the Frege proof must be large, respectively long in the case of Frege proofs of bounded line size.

In order to formalize this argument we need to fix a Frege system so we can argue about the derivation rules. By a result of Cook and Reckhow [CR79] the precise choice of the Frege system is not important and we choose the same system as [PRST16; Hås20; PRT22]. This system consists of the following rules.

- (Excluded middle) $(p \vee \neg p)$
- (Expansion rule) $p \rightarrow (p \vee q)$
- (Contraction rule) $(p \vee p) \rightarrow p$
- (Association rule) $p \vee (q \vee r) \rightarrow (p \vee q) \vee r$
- (Cut rule) $p \vee q, \neg p \vee r \rightarrow q \vee r$

These rules should be understood in the following manner: a depth d Frege proof can at any time, by excluded middle, write down a line of the form $(p \vee \neg p)$ for any formula p if the line is of depth at most d. Similarly the expansion rule says that if we have derived the formula p, then we can write down the line $(p \vee q)$ for any formula q such that the line is of depth at most d. The crucial lemma is as follows.

Lemma A.6.2. *Suppose we have a derivation using the above rules starting from the Tseitin axioms defined on the $n \times n$ grid, that also has a t-evaluation. Then, if $t \leq n/8$, each line in the derivation is mapped to a 1-tree. This, in particular, implies that we cannot derive contradiction.*

The proof in the standard case of this lemma is again a tedious and formal verification and can be found in full in [Hås20]. The proof is by induction over the number of derivation steps and the key property is to take any path that leads to 0 in the derived formula and find a path that leads to a 0 in one of the assumptions. The fact that all decision trees are of depth less than $n/8$ ensures that it is possible to find a branch of any decision tree that is consistent with the given 0-branch.

In the current case, where each line has its own t-evaluation, due to consistency, not much is different. We can again take any 0-branch in the decision tree of a derived formula and find a 0-branch in one of the assumptions. Instead of repeating all cases let us do only the most interesting one: the cut rule.

We have $F = (q \vee r)$ derived on line ν and suppose $\varphi^{\nu}(F)$ is not a 1-tree. Take a supposed leaf with label 0 in $\varphi^{\nu}(F)$ and let τ be the assignment leading to this leaf. We know that $\varphi^{\nu}(q)\lceil_{\tau}$ and $\varphi^{\nu}(r)\lceil_{\tau}$ are both 0-trees by the definition of a t-evaluation.

Now suppose $(p \vee q)$ was dervied on line $\nu' < \nu$ and $(\neg p \vee r)$ was derived on line $\nu'' < \nu$. By consistency of ν and ν' we know that $\varphi^{\nu'}(q)\lceil_{\tau}$ is a 0-tree and, as also ν and ν'' are consistent, so is $\varphi^{\nu''}(r)\lceil_{\tau}$.

Now, if any branch in $\varphi^{\nu'}(p)\lceil_\tau$ ends in a leaf labeled 0, then $\varphi^{\nu'}(p \vee q)\lceil_\tau$ can be extended to reach a 0-leaf. This is in contradiction to the inductive assumption. For similar reasons $\varphi^{\nu''}(\neg p)\lceil_\tau$ is a 1-tree. This contradicts the assumed consistency of ν' and ν''.

A.7 Proofs of the main theorems

We first reprove the main theorem of [Hås20] with improved parameters.

Theorem A.7.1. *For $d \leq O(\frac{\log n}{\log \log n})$ the following holds. Any depth-d Frege refutation of the Tseitin contradiction defined on the $n \times n$ grid requires size*

$$\exp\left(\Omega(n^{1/(2d-1)}(\log n)^{O(1)})\right) .$$

As outlined in the previous section, we construct a t-evaluation for all sub-formulas occurring in a short and shallow Frege proof. By Lemma A.6.2 we then conclude that all shallow Frege proofs of the Tseitin contradiction must be long. For the total size lower bound we in fact do not create distinct t-evaluations per line but rather a single one, used on each line. Such a t-evaluation is clearly consistent and hence satisfies our needs. Let Γ denote the set of sub-formulas occurring in the alleged proof. Our plan is to proceed as follows for $i = 0, 1, 2, \ldots, d$.

- We have a t-evaluation for all formulas of Γ that were originally of depth i.
- Pick a random full restriction σ_i and extend the t-evaluation to all formulas of $\Gamma\lceil_{\sigma_i}$ of original depth at most $i + 1$.

At the starting point, $i = 0$, each formula is a literal which is represented by a natural decision tree of depth 1. In order to extend the t-evaluation to larger depth we use the following lemma, central to the argument.

Lemma A.7.2 (Switching Lemma)**.** *There is a constant A such that the following holds. Suppose there is a t-evaluation that includes $F_i, 1 \leq i \leq m$ in its domain and let $F = \vee_{i=1}^m F_i$. Let σ be a random full restriction from the space of restrictions defined in Section A.4. Then the probability that $F\lceil_\sigma$ cannot be represented by a decision tree of depth at most $s \geq t$ and the number of live variables in each center is in the interval* $[.99C \log n, 1.01C \log n]$ *is at most*

$$\left(A(\log n)^{27} t\Delta^{-1}\right)^{s/108} .$$

We postpone the proof of this lemma to Section A.8 and see how to use it when studying a refutation of size N. We start with a t_1-evaluation with

$t_1 = 1$ for single literals and apply the lemma with $s = \Omega(\log N)$ in the first step, while we choose $t_i = s$ in later steps. We set $\Delta_i = \Omega(t_i(\log n)^{27})$ and hence have that $T_i = 4\Delta_i^2$ for each step.

We start with the original Tseitin contradiction on the $n \times n$ grid. Start with $n_0 = n$ and set $n_{i+1} = n/T_i$ for $i = 0, 1, \ldots, d-1$. We are going to choose a sequence of full restrictions σ_i mapping a grid of size n_i to a grid of size n_{i+1} randomly. Let σ_i^* be the composition of $\sigma_0, \sigma_1, \ldots, \sigma_i$. Let Γ be the set of sub-formulas that appear in an alleged proof and we let

$$\Gamma_i = \{F\lceil_{\sigma_i^*} \mid F \in \Gamma \wedge \mathrm{depth}(F) \leq i\} \enspace .$$

Let f_i be the number of sub-formulas of depth at most i in Γ.

Lemma A.7.3. *With probability* $1 - f_i 2^{-\Omega(s)}$ *there is a* t*-evaluation* φ_i *whose domain includes* Γ_i*.*

Proof. This is essentially collecting the pieces. We prove the lemma by induction over i. For $i = 0$ we have the t-evaluation that maps each literal to its natural decision tree of depth 1.

When going from depth i to depth $i + 1$ we need to define φ_{i+1} on all formulas originally of depth at most $i + 1$ and consider any such F.

1. Each F of depth at most i is, by induction, in the domain of φ_i and we can appeal to Lemma A.6.1.
2. If F is of depth i then $\varphi_{i+1}(\neg F)$ is defined from $\varphi_{i+1}(F)$ negating the labels at the leaves.
3. For $F = \vee F_i$ where each F_i is of at most depth i we apply Lemma A.7.2.

The only place where the extension might fail is under step three but, by Lemma A.7.2, the probability of failure for any individual formula is at most $2^{-\Omega(s)}$ and as we have at most $f_i - f_{i-1}$ formulas of depth exactly i the induction is complete. □

Fixing parameters we reprove the main theorem from [Hås20] with stronger parameters.

Proof of Theorem A.7.1. Suppose we have a refutation of size

$$N \leq \exp(c_1(n^{1/(2d-1)}(\log n)^{-c_2})) \enspace ,$$

for suitable positive constants c_1 and c_2. In the first iteration we use Lemma A.7.2 with $t = 1$ and $\Delta = (2tA(\log n)^{27})^{-1}$ and $s = 110 \log N$. In later applications we use $t = s$. It is easy to see that with these numbers we have successful switching at each round with high probability. The number of live centers are in the desired interval and we are always able to construct the new t-evaluation.

Up to polylogarithmic factors we have that the final side length of the grid after all the restrictions is $n(\log N)^{-2(d-1)}$ and it is a t-evaluation with $t = O(\log N)$. Thus if $\log N$ is a polylogarithmic factor smaller than $n^{1/(2d-1)}$ we get a contradiction to Lemma A.6.2. □

Let us turn our attention to the main result of the present paper.

Theorem A.7.4. *For any Frege proof of the Tseitin principle defined over the $n \times n$ grid graph the following holds. If each line of the proof is of size M and depth d, then the number of lines in the proof is*

$$\exp\left(\Omega\left(\frac{n}{\left((\log n)^{O(1)} \log M\right)^{2d}}\right)\right) .$$

The strategy of the proof is similar to the proof of Theorem A.7.1: we again build a t-evaluation for a supposed Frege proof. The main difference is that instead of creating a single t-evaluation for the entire proof we in fact independently create t-evaluations for each line. These t-evaluations turn out to be consistent, as defined in Section A.6, and we thus obtain the claimed bounds.

Suppose we are given a Frege refutation of the Tseitin principle defined over the $n \times n$ grid consisting of N lines, where each line is a formula of size M and depth d. We denote by Γ^v the set of sub-formulas of line v in the proof and continue to construct a sequence of restrictions $\sigma_1, \sigma_2, \ldots, \sigma_d$ such that all formulas of depth at most k have consistent $t(k)$-evaluations if hit by the concatenation σ_k^* of the first k restrictions in the sequence, where $t(k)$ is some function dependent on k to be fixed later. That is, for every line v we have a $t(k)$-evaluation φ_k^v for all formulas in the set

$$\Gamma_k^v = \{F\lceil_{\sigma_k^*} \mid F \in \Gamma^v \wedge \mathrm{depth}(F) \leq k\},$$

and all these $t(k)$-evaluations are consistent. In addition to these $t(k)$-evaluations, for each line v we also maintain a decision tree $\mathcal{T}_k(v)$. We maintain the property that $\mathcal{T}_k(v)$ is a t common partial decision tree for all $t(k)$-evaluations $\varphi_k^v(\Gamma_k^v)$ of bounded depth.

These partial common decision trees $\mathcal{T}_k(v)$ are useful to extend the $t(k)$-evaluations φ_k^v to larger depths. In each such step, increasing k, we apply for each branch π from $\mathcal{T}_k(v)$ the following multi-switching lemma to the set of decision trees $\varphi_k^v(\Gamma_k^v)\lceil_\pi$ of depth at most t. We then extend $\mathcal{T}_k(v)$ in each leaf π by the the partial common decision tree from the lemma to obtain $\mathcal{T}_{k+1}(v)$ of slightly larger depth.

Lemma A.7.5 (Multi-switching Lemma). *There are constants* A, c_1, *and* c_2 *such that the following holds. Consider formulas* F_i^j, *for* $j \in [M]$ *and* $i \in [m_j]$, *each associated with a decision tree of depth at most* t *and let* $F^j = \vee_{i=1}^{m_j} F_i^j$. *Let* σ *be a random full restriction from the space of restrictions defined in Section A.4. Then the probability that the number of live variables in each center is in the interval* $[.99C\log n, 1.01C\log n]$ *and* $(F^j\lceil_\sigma)_{j=1}^M$ *cannot be represented by an* ℓ *common partial decision tree of depth at most* s *is at most*

$$M^{s/\ell}\left(A(\log n)^{c_1}t\Delta^{-1}\right)^{s/c_2} .$$

We defer the proof of this lemma to Section A.9. We apply Lemma A.7.5 with mostly the same parameters so let us fix these. We choose $\ell = t = \log M$ and $\Delta = D \cdot t \cdot (\log n)^{c_1}$, for a sufficiently large constant D. The parameter s depends on k and is fixed to $s = s_k = 2^{k-1}\log N$. With these parameters in place we can finally also fix $t(k) = \sum_{i\le k} s_i + \log M \le 2^k \log N + \log M$.

Lemma A.7.6. *Suppose that for every line* $\nu \in [N]$ *we have consistent* $t(k-1)$-*evaluations* φ_{k-1}^ν *for formulas in* Γ_{k-1}^ν *along with a* t *common partial decision tree* $\mathcal{T}_{k-1}(\nu)$ *for* $\varphi_{k-1}^\nu(\Gamma_{k-1}^\nu)$ *of depth* $\sum_{i<k} s_i$. *Then, with probability* $1 - N^{-1}$, *there is a full restriction* σ_k *whose output grid is of dimension* n *and, assuming that* $t(k) \le n/8$, *for every line* $\nu \in [N]$ *there is a consistent* $t(k)$-*evaluation* φ_k^ν *for formulas in* Γ_k^ν *and a* t *common partial decision tree* $\mathcal{T}_k(\nu)$ *for* $\varphi_k^\nu(\Gamma_k^\nu)$ *of depth* $\sum_{i\le k} s_i$.

Proof. Let us first extend the common partial decision trees and then explain how to obtain φ_k^ν for different lines $\nu \in [N]$.

The interesting formulas of original depth k to consider are the ones with a top $\vee$ gate. Let us fix a line $\nu \in [N]$ and consider all sub-formulas $\{F^j = \vee_{i=1}^{m_j} F_i^j\}_{j=1}^{M_\nu}$ of line ν of original depth k with a top $\vee$ gate under the restriction σ_{k-1}^*. As the original depth of every F_i^j is at most $k-1$, all these formulas are in the domain of φ_{k-1}^ν. Let us further fix a path π in $\mathcal{T}_{k-1}(\nu)$ and recall that all decision trees $\varphi_{k-1}^\nu(F_i^j)\lceil_\pi$ are of depth at most t.

For every $\nu \in [N]$ and branch π of $\mathcal{T}_{k-1}(\nu)$ we apply Lemma A.7.5 to the set of formulas $F_i^j\lceil_\pi$ with associated trees $\varphi_{k-1}^\nu(F_i^j)\lceil_\pi$ of depth at most t. The probability of failure of a single application is bounded by N^{-2^k-1}, assuming an appropriate choice of the constant D. As we invoke Lemma A.7.5 at most $N \cdot 2^{\sum_{i<k} s_i} \le N^{2^k}$ times, by a union bound, with probability at least $1 - N^{-1}$, there is a full restriction σ_k such that for every line $\nu \in [N]$ and every branch $\pi \in \mathcal{T}_{k-1}(\nu)$ we get a t common partial decision tree of depth at most s_k for the formulas $(F^j\lceil_{\pi\sigma_k})_{j=1}^{M_\nu}$. Let us denote this common decision tree by $\mathcal{T}(\nu,\pi)$ and attach it to $\mathcal{T}_{k-1}(\nu)$ at

the leaf π to obtain $\mathcal{T}_k(\nu)$. The trees $\mathcal{T}_k(\nu)$ are of depth at most $\sum_{i\leq k} s_k$ as required.

Let us explain how to define φ_k^ν for a fixed line $\nu \in [N]$. Consider any formula F in Γ_k^ν.

- If F is of depth less than k, then F is in the domain of φ_{k-1}^ν and we can appeal to Lemma A.6.1.
- If F is of depth $k-1$ then $\varphi_k^\nu(\neg F)$ is defined from $\varphi_k^\nu(F)$ negating the labels at the leaves.
- For $F = \vee_i F_i$ of depth k we use the previously constructed common partial decision trees. We define $\varphi_k^\nu(F)$ to be the decision tree whose first $\sum_{i\leq k} s_i$ levels are equivalent to $\mathcal{T}_k(\nu)$ followed by t levels unique to F obtained from the multi-switching lemma.

Let us check that the decision trees $\mathcal{T}_k(\nu)$ are indeed t common partial decision trees for $\varphi_k^\nu(\Gamma_k^\nu)$. By construction this clearly holds for formulas of depth k with a top $\vee$ gate. As $\mathcal{T}_k(\nu)$ is equivalent to $\mathcal{T}_{k-1}(\nu)$ on the upper levels, and restrictions only decrease the depth of decision trees, by the initial assumptions this also holds for formulas of depth less than k. As the $t(k)$-evaluations of formulas of depth k with a top $\neg$-gate are defined in terms of formulas of depth less than k, we also see that $\mathcal{T}_k(\nu)$ is a t common partial decision tree for such formulas.

Last we need to check that each φ_k^ν is a $t(k)$-evaluation plus that these are pairwise consistent.

By Lemma A.6.1 all the properties hold for formulas of depth less than k. Let us verify the $t(k)$-evaluation properties for formulas of depth k.

Property 1 is immediate, as $k > 0$. As we only consider consistent decision trees, property 2 also follows. Further, property 3 is satisfied by construction. Property 4 can be established by checking the property for each branch π in $\mathcal{T}_{k-1}(\nu)$ separately; for a fixed π we see by Lemma A.7.5 that this indeed holds.

Finally we need to establish that two $t(k)$-evaluations φ_k^ν and $\varphi_k^{\nu'}$ are consistent for formulas of depth k. By the inductive hypothesis we clearly have that $\neg$-isomorphic formulas are $\neg$-consistent. Further, isomorphic formulas with a top $\neg$ gate are consistent. Hence we are only left with checking consistency for isomorphic formulas of depth k with a top $\vee$ gate.

Let $F = \vee_i F_i$ and $F' = \vee_i F'_i$ be two isomorphic formulas from Γ_k^ν and $\Gamma_k^{\nu'}$ respectively. For the sake of contradiction suppose $\varphi_k^\nu(F)\lceil_\pi = 1$ but $\varphi_k^{\nu'}(F')\lceil_\pi = 0$ for some assignment π. In the following we use that $t(k) \leq n/8$ and hence there are consistent branches as claimed. By property 2 we know that for some F_i it holds that $\varphi_k^\nu(F_i)\lceil_\pi = 1$. As F and F' are isomorphic formulas we know that there is an F'_j such that F_i and F'_j are isomorphic formulas. As such formulas have consistent decision trees (by induction and Lemma A.6.1) we get that $\varphi^{\nu'}_k(F'_j)\lceil_\pi = 1$. But this cannot be as by property 4 of a $t(k)$-evaluation this implies that $\varphi^{\nu'}_k(F')\lceil_\pi = 1$. This establishes that the different $t(k)$-evaluations are consistent, as required. □

With all pieces in place we are ready to prove Theorem A.7.4.

Proof of Theorem A.7.4. Suppose we are given a proof of length

$$N = \exp(n/((\log n)^c \log M)^{2d}) \ ,$$

for some constant c. We may assume that $M \le \exp(n^{1/2d-1/2d(2d-1)})$, as otherwise we can apply Theorem A.7.1.

In order to create the consistent $t(k)$-evaluations φ^ν for each line $\nu \in [N]$ we consecutively apply Lemma A.7.6 d times. We start with φ^ν_0 which maps constants to the appropriate depth 0 decision tree and literals to the corresponding depth 1 decision trees. The partial common decision trees $\mathcal{T}_0(\nu)$ are all empty.

After applying Lemma A.7.6 d times we are left with a $t(d)$-evaluation for the proof. We need to ensure that $t(d)$ is upper bounded by the dimension of the final grid: $t(d) \le 2^d \log N + \log M$, while the final side length of the grid is $n \cdot (4\Delta^2)^{-d} = n \cdot (2D(\log n)^{c_1} \log M)^{-2d}$. For our choice of N and the assumption on M this indeed holds and by Lemma A.6.2 the theorem follows. □

A.8 The improved standard switching lemma

This section is dedicated to the proof of the switching lemma, restated here for convenience.

Lemma A.7.2 (Switching Lemma)**.** *There is a constant A such that the following holds. Suppose there is a t-evaluation that includes $F_i, 1 \le i \le m$ in its domain and let $F = \vee_{i=1}^m F_i$. Let σ be a random full restriction from the space of restrictions defined in Section A.4. Then the probability that $F\lceil_\sigma$ cannot be represented by a decision tree of depth at most $s \ge t$ and the number of live variables in each center is in the interval $[.99C \log n, 1.01C \log n]$ is at most*

$$\left(A(\log n)^{27} t\Delta^{-1}\right)^{s/108} \ .$$

The proof very much follows the proof of [Hås20]. In fact large parts of the proof are the same. We repeat the proofs to make it possible for a reader not familiar with the mentioned proof to follow the argument.

To make the argument slightly shorter we do not repeat all proofs of the various lemmas.

For the benefit of the reader completely on top of [Hås20] let us outline the differences in the following section. This section can be safely skipped by the less experienced reader.

A.8.1 Changes in the Argument

The key number that has changed is the parameter k, the total number of centers that are alive. In the definition of a partial restriction this parameter k has changed from $Cs(n/T)^2$ to $C \log n(n/T)^2$. The fact that we had $\Omega(s)$ live centers in each square was crucial in finding live centers to extend the information sets J_j to closed sets γ_j. This process needed $O(s)$ fresh centers from specific squares and there is nothing that prevents these from all being required to be in the same square. In the current proof we allow γ_j to be not closed and this implies that the restriction ρ^* is a generalized restriction where the Tseitin condition is violated at some vertices. This only happens when we have $\Omega(\log n)$ exposed non-chosen centers in a sub-square and results in a single violating vertex. As the there are at most $O(s)$ exposed centers over all we can have at most $O(s/\log n)$ violating centers. The number of generalized restrictions with B violating centers is at most a factor n^{2B} more than the the number of ordinary restrictions. This number is $2^{O(s)}$ and this factor can be absorbed in the constant A in the statement of the switching lemma.

A.8.2 Proof Overview

Let us recall the setup. We have a full restriction σ as defined in Section A.4 that is made up of a restriction ρ and a pairing π. The restriction ρ has $(1 \pm 0.01)C \log n$ many live centers in each sub-square, for a large enough constant C. We have a formula $F = \vee_{i=1}^{m} F_i$ and a t-evaluation φ that includes each F_i in its domain and let $T_i = \varphi(F_i)$. As φ is a t-evaluation each such tree T_i is of depth at most t.

In the following we construct a decision tree $\mathcal{T}$ for $F\lceil_\sigma$ which is with high probability, over the choice of ρ, of depth at most s. The decision tree $\mathcal{T}$ is created in a similar manner as the canonical decision tree is usually constructed: we proceed in stages, where in each stage the current branch τ is extended by querying variables related to the first 1-branch ψ in the trees $T_1\lceil_{\sigma\tau}, T_2\lceil_{\sigma\tau}, \ldots, T_m\lceil_{\sigma\tau}$. For now it is not so important what the related variables of ψ precisely are and we can simply think of these as the variables on the branch ψ. Once all these variables have been queried, we check in each new leaf of the tree whether we traversed the path ψ. If so, then we label the leaf with a 1 and otherwise we continue with the next stage. If there are no 1-branches left, we label the leaf with a 0.

It is not so hard to see that this process indeed results in a tree $\mathcal{T}$ that represents $\vee_{i=1}^{m} T_i\lceil_{\sigma}$: for each leaf τ of $\mathcal{T}$ that is labeled 1 it holds that there is an $i \in [m]$ such that $T_i\lceil_{\sigma\tau} = 1$ and if τ is labeled 0, then for all $i \in [m]$ we have that $T_i\lceil_{\sigma\tau} = 0$, as required. It remains to argue that $\mathcal{T}$ is with high probability of depth at most s.

We analyse this event using the labeling technique of Razborov [Raz95]. The idea of this technique is to come up with an (almost) bijection from restrictions ρ that give rise to a decision tree $\mathcal{T}$ of depth larger than s to a set of restrictions that is much smaller than the set of all restrictions. In a bit more detail, given such a bad ρ, we create a restriction ρ^* with fewer live centers such that with a bit of extra information we can recover ρ from ρ^*. As the restriction ρ^* has roughly s fewer live centers than ρ, and the inversion requires little extra information, we obtain our statment.

Let us explain how to obtain ρ^* from a ρ that gives rise to a decision tree $\mathcal{T}$ of depth larger than s. To this end, we first need to slightly refine the construction process of $\mathcal{T}$. Namely, we need to discuss what the related variables of a branch ψ are. Instead of thinking of this as a set of variables we rather want to think of it as an information set J, as introduced in Section A.4. The information set J is a minimal set that forces, along with the already collected information set on the branch τ, the branch ψ. Once we identified such a set J, we then query all necessary variables to see whether we agree with J (along with some further variables).

Recall that we are trying to explain how to construct ρ^* from a ρ that gives rise to a decision tree $\mathcal{T}$ of large depth. Fix a long branch τ in $\mathcal{T}$ and consider all the sets $J_1, J_2, \ldots, J_g$ identified on τ. For this proof overview, let us assume that each J_j is closed and the support of these information sets are pairwise disjoint. Let us stress that this is a simplification and does not hold in general. Assuming this holds, note that the union $J^* = \cup_{i=1}^{g} J_j$ is also closed and recall from Section A.4 that all variables forced by (ρ, J^*) can be described by a restriction where the centers in $\mathrm{supp}(J^*)$ are killed. This defines the restricion ρ^*: it is the restriction that forces all variables forced by (ρ, J^*). Assuming that the support of J^* is large, we see that ρ^* has much fewer centers that are alive.

What remains is to argue that we can cheaply recover ρ from ρ^*. The idea is to remove the set J_j, starting with $j = 1$, one-by-one from ρ^*. To do this cheaply we use the decision trees $T_1, \ldots, T_m$. Recall that the information set J_1 determines all variables on the first 1-branch ψ_1. This implies in particular that ρ^* traverses the branch ψ_1. Hence identifying ψ_1 is for free: it is the first 1-branch in $T_1, \ldots, T_m$ traversed by ρ^* (assuming that the set J_1

is pairwise disjoint from all later sets J_j). Once we identified the branch ψ_1, we want to recover the first part of the long branch τ so that we can repeat this argument with J_2. As ψ_1 is of length at most t, using only $\log t$ bits per variable, we indicate which variables are different on τ from J_1. This lets us cheaply recover τ along with the centers killed by J_1. Repeating this argument g times lets us recover ρ.

This completes the proof overview. We allowed ourselves several simplifications and left out a fair number of details. The most significant simplification is the assumption that all the information sets J_j are closed. In the actual proof we extend each set J_j into a closed set γ_j and then take the union of these to define ρ^*. The process of closing a set J_j may even fail at times and therefore ρ^* has to slightly bend the rules of being a restriction. It turns out that ρ^* is a generalized restriction as mentioned in Section A.4.3. The step of closing up the J_j is the main source of technical difficulty in the full proof.

The proof is split into four separate sections. In Section A.8.3 we define the extended canonical decision tree $\mathcal{T}$ and in the susequent Section A.8.4 we prove some crucial properties of these decision trees. Section A.8.5 explains how to extend the sets J_j into closed information sets γ_j in order to construct the restriction ρ^*. Finally, in Section A.8.6 we show how to cheaply recover ρ from ρ^* and thereby prove Lemma A.7.2.

A.8.3 Extended Canonical Decision Trees

Let us construct an *extended canonical* decision tree $\mathcal{T}$ for $F\lceil_\sigma$. We start with $\mathcal{T}$ the empty tree and extend it for each branch τ separately. For every branch τ we maintain the following objects throught the creation of $\mathcal{T}$:

1. a set $S = S(\tau, \sigma)$ of centers, called the *exposed centers*,
2. a set $I = I(\tau, \sigma)$ of information pieces as defined in Definition A.4.5, and
3. a (state of a) matching game $\mathcal{G} = \mathcal{G}(\tau, \sigma)$, as described in Section A.3, played on the chosen centers of σ.

Initially the sets S and I are empty, and the matching game $\mathcal{G}$ is a new game with no vertices matched. We require that S, I and $\mathcal{G}$ satisfy the following invariants.

1. No element is ever removed from S or I. In other words, the sets S and I only become larger throughout the creation of a branch τ.
2. The matched nodes in the game $\mathcal{G}$ are precisely the chosen centers in S.

3. The information set I does not contain a path between a chosen center and a non-chosen center.

4. For non-chosen centers in S, the set I consists of the closed information pieces corresponding to their component in π (both edges and non-edges). If one center of such a connected component belongs to S, then so does the entire component. Thus for non-chosen centers in S we have information pieces in all four directions.

5. For every chosen center in S we have queried all incident variables x_P in τ and this is the information that is present as information pieces in I. The one-answers are recorded in the form of a path while the zero answers as two non-edges, one at the neighboring chosen center in the appropriate direction which may or may not be an element of S. Observe that the value of x_P jointly with ρ determines the value of all variables x_e on the chosen path P.

Let us stress the fact that information about π comes from the restriction σ and hence in Invariant 4 we do not query a variable in $\mathcal{T}$. However, querying a variable x_P, as done in Invariant 5, causes a query in the decision tree $\mathcal{T}$.

Further, observe that there is a crucial difference between Invariant 4 and Invariant 5: on the non-chosen centers we have information pieces in I only on the centers in S. In contrast I may contain information pieces from chosen centers that are not in S.

Let us discuss the creation of $\mathcal{T}$. We proceed in stages. In each stage we fix a branch τ in $\mathcal{T}$. We go over the decision trees $T_i = \varphi(F_i)$ one by one. Suppose we consider T_i. Take the first (in some fixed order) branch ψ in T_i that leads to a leaf labeled 1 which is consistent with τ and σ. If there is no such branch, then we continue with T_{i+1} and if there is no such branch ψ for any T_i, then we label the τ leaf of $\mathcal{T}$ by 0 and continue with a different branch τ' of $\mathcal{T}$ until all leaves of $\mathcal{T}$ are labelled. But for now let us assume that there is a branch ψ as described.

For the variables appearing on ψ we have unique values required to reach this leaf. We let a *possible forcing information* J be an information set that jointly with I and ρ forces[4] all variables on ψ to take these unique values. Let us call ψ the *forceable branch*. The intuition is that if the information set J agrees with the actual input, then indeed ψ is followed and we can safely end with a 1-leaf. In most cases, however, the actual input does not agree with J and we need to continue evaluating the extended canonical decision tree $\mathcal{T}$. We require the following properties of J.

[4] Recall from Definition A.4.6 that a variable is forced if we have the relevant information at its closest endpoint.

1. If J contains a non-edge from a chosen center it also contains a non-edge in the "reverse direction". As an example if it contains a non-edge going left from a chosen center v then it contains a non-edge going right from the chosen center in the sub-square to the left of v.
2. The information set J does not contain a path between a chosen center and a non-chosen center.
3. The information sets I and J are consistent and disjoint.
4. The part of J on the non-chosen centers is closed and consistent with π, that is, J contains a subset of the components of π in the form of a closed set of information pieces.
5. J is minimal given the above properties and the fact that, along with I, it should determine the values of all the variables on the forceable branch ψ.

Note that a set J may not be unique for a given path ψ. If there are several sets as described above, choose one in a fixed but otherwise aribitrary manner. While the choice is not essential for what follows, we do need to establish that whenever some T_i can still reach a 1-leaf, then there is a possible forcing information J. We postpone this to the following section (see Lemma A.8.1) and for now assume that such a set J exists whenver we have a branch ψ as described.

Denote by U the set of closest endpoints of variables on ψ that are chosen centers but not contained in S. A somewhat subtle point to note is that U may contain a closest endpoint of a variable that is determined by I: the set I may contain information pieces about chosen centers outside the set of exposed centers S. The set U is needed to ensure that we treat such centers correctly.

Let us continue the construction of the extended canonical decision tree $\mathcal{T}$ at τ. Add U and all centers in $\mathrm{supp}(J)$ to S along with the centers described next. Let the adversary in the game $\mathcal{G}$ supply U along with all chosen centers in $\mathrm{supp}(J)$. We apply Lemma A.3.5 and add all nodes provided by P_M to S (we tacitly assume throughout that $|S| \leq n/2$). Observe that this game is played on nodes of the grid and does not take into account any other information from I or J.

Finally we need to update I and extend $\mathcal{T}$. This is straightforward for the non-chosen centers added to S: for every such non-chosen center v we add the information from v's connected component in π to I (in the form of edges and non-edges).

For every chosen center added to S we query all the incident variables, thereby extending $\mathcal{T}$. For every newly created consistent extension τ'

of τ we need to update the set I. Record one-answers as an edge and zero-answers as two non-edges including the other endpoint of a potential chosen path, i.e., the chosen center in the adjacent sub-square in the given direction. Recall that we only consider consistent branches τ' (as assignments) and hence we create consistent information sets.

Finally, for every consistent τ' extending τ, we check whether the information set $I(\tau', \sigma)$ traversed the forceable branch ψ of T_i. This can clearly be done: all variables on ψ have their closest endpoint in S and each exposed center has information pieces in all four directions. If ψ is indeed followed, we label the leaf τ' with a 1. Otherwise, if the forceable branch is not followed, then we proceed with the next stage.

This completes the description of the creation of the extended canonical decision tree $\mathcal{T}$ for $F\lceil_\sigma$. It is straightforward to check that the invariants hold after every completed stage.

A.8.4 Some Properties of Extended Canonical Decision Trees

In this section we prove two important properties of extended canoncial decision trees, along with some auxilary lemmas. The first important property is that the decision tree $\mathcal{T}$ does indeed represent $\vee_{i=1}^{m} T_i\lceil_\sigma$. Secondly, we show that the construction process of $\mathcal{T}$ is independent of the choice of the negations of the preferred values along the paths between chosen centers. This allows us to focus on long branches that have well-behaved information sets I.

Before proving these two statements, recall that we postponed the proof of the claim that if it is possible to each a 1-leaf of T_i, then there is a possible forcing information J. Let us establish this fact. Observe that at any point when forming the extended canonical decision tree, the information I comes from information in π and from queries already done in the decision tree $\mathcal{T}$ with answers τ. Remember that σ includes all the information from π.

Lemma A.8.1. *If there is a 1-branch ψ in $T_i\lceil_\sigma$ that is consistent with τ, then there is a possible forcing information J for ψ.*

Proof. Let ψ' be the branch in T_i that gives rise to ψ. Consider the assignment τ' to the path variables x_P such that the 1-leaf of $T_i\lceil_\sigma$ is reached. Let us find a possible J such that ψ' is followed.

The information pieces next to chosen centers are simply those given by τ'. These are, by definition, consistent with τ and can hence be included in J.

The information pieces next to non-chosen centers are the relevant information pieces from π. As all information pieces from π are consistent, consistency is automatically satisfied for these pieces.

Dropping any non-required piece and all the pieces already in I makes J disjoint from I and minimal. Clearly J forces ψ' to be followed. This completes the proof of the lemma. □

As an immediate corollary we have that the decision tree $\mathcal{T}$ is indeed a legitimate choice for $\varphi(F\lceil_\sigma)$.

Corollary A.8.2. *The extended canonical decision tree $\mathcal{T}$ represents $\vee_{i=1}^m T_i\lceil_\sigma$.*

The creation of the extended canononical decision tree depends on ρ and π but not, in a serious way, on the negations of the preferred values along the paths between the chosen centers. The following lemma makes this intuition precise.

Lemma A.8.3. *Let σ_1 be obtained from ρ_1 and π and σ_2 from ρ_2 and π where ρ_1 and ρ_2 pick the same set of centers and fixed values. Assume furthermore that the only difference between ρ_1 and ρ_2 is that for each chosen path P there is a bit c_P such that for each grid-edge e on P the preferred values of x_e differ by c_P in ρ_1 and ρ_2. Then the only difference between the extended canonical decision trees of $F\lceil_{\sigma_1}$ and $F\lceil_{\sigma_2}$ is the labeling of the internal edges.*

Proof. This follows by inspection of the procedure for forming the extended canonical decision tree. The only difference is that variables on chosen paths in one case are forced by a path and in the other case by two non-edges. This does not cause any difference in the construction of $\mathcal{T}$ as the supports of the two corresponding sets J_1 and J_2 are identical by Property 1 of a possible forcing information. □

This lemma is crucial in our analysis. It allows us to focus on long branches whose information set I is well-behaved in the following sense.

Definition A.8.4 (Closed branch)**.** Let $\mathcal{T}$ be an extended canonical decision tree. A branch τ in $\mathcal{T}$ is *closed* if the information set $I(\tau, \sigma)$ contains a path between two chosen centers u, v if and only if the matching game $\mathcal{G}(\tau, \sigma)$ matched u to v.

This slightly overloads the notion "closed" but as the information pieces given by the answers on a closed branch τ is (essentially) a closed information set we hope that this causes no confusion. The following lemma is an immediate consequence of Lemma A.8.3.

Lemma A.8.5. *If the probability that $F\lceil_\sigma$ needs a decision tree of depth s is at least q, then the probability that the extended canonical decision tree of $F\lceil_\sigma$ contains a closed branch of length at least s is at least $2^{-s}q$.*

This lemma allows us to only analyze closed branches. The main advantage of considering closed branches is that the information sets I have a nice structure. We use the following property throught the proof.

Lemma A.8.6. *On a closed branch, after the completion of a stage, I consists of a closed part on the exposed vertices S jointly with a set of non-edges from chosen centers not in S towards chosen centers in S.*

Proof. The information in I about non-chosen centers in S is from π and thus by definition closed. Further, because we are on a closed branch, the set I is also closed on the chosen centers in S. The only other information pieces in I are non-edges from chosen centers not in S towards chosen centers in S. □

Lastly we have an auxillary lemma regarding the size of the set of exposed centers S.

Lemma A.8.7. *In each stage at most $8t$ vertices are added to the set of exposed vertices S.*

Proof. A forceable branch ψ is of length at most t as the trees T_i are of depth at most t. For each variable x_e on ψ there are at most 2 chosen centers in $\text{supp}(J) \cup U$ if the closest endpoint of x_e is chosen and at most 4 non-chosen centers if the closest endpoint is non-chosen.

When adding $\text{supp}(J) \cup U$ to S we add at most 1 extra center per chosen center in $\text{supp}(J) \cup U$ to S. We conclude that at most $8t$ vertices are added to S in a given stage. □

A.8.5 From ρ to ρ^*

We want to bound the number of restrictions ρ (as defined in Section A.4) that give rise to an extended canonical decision tree $\mathcal{T}$ of depth at least s. In light of Lemma A.8.5 we can focus on $\mathcal{T}$ that contain a *closed* branch τ of length at least s. Let us fix such a ρ along with the extended canonical decision tree $\mathcal{T}$ and the closed branch τ of length at least s.

The goal of this section is to construct a restriction ρ^* that is related to ρ but has fewer live variables. In the following section we then show how to recover ρ from ρ^* with a bit of extra information. As ρ^* has fewer live variables there are fewer such restrictions and, assuming we require only little extra information to recover ρ, we thus establish that there are very few ρ that cause the extended canonical decision trees to be of depth at least s.

Recall that an information set γ^* is closed if for every center v in $\mathrm{supp}(\gamma^*)$ the set γ^* contains information in all four directions of v and, furthermore, γ^* has an odd number of edges incident to every such v. The idea is to reduce the number of live variables with the help of a closed information set γ^*. Consider all variables forced by (ρ, γ^*). Observe that (ρ, γ^*) can be described by a restriction ρ^* where all the centers in $\mathrm{supp}(\gamma^*)$ are killed: negate the values of any preferred variable on any path in γ^*. In the following we are going to construct a closed information set γ^* with large support (linear in s) such that ρ can be recovered from the resulting ρ^* with a bit of extra information.

As suggested in the proof outline, we would like to choose γ^* to be the union of all the possible forcing information sets used when creating the long branch τ. Unfortunately this does not work: a possible forcing information is not always closed and insisting on a possible forcing information to be closed creates a dependence between $\mathcal{T}$ and the negations of the preferred values along paths between chosen centers. As such it becomes difficult to prove the crucial Lemma A.8.5.

So it is not obvious how to guarantee that the possible forcing information is closed. What we can do, however, is to close these information sets *after* we have found a long closed branch τ. We can then take γ^* to be the union of these newly closed possible forcing information sets. Let us proceed by explaining how to close the possible forcing information sets.

As τ is a branch of length s, there is a first stage g such that at the end of stage g at least $s/4$ centers are exposed: only variables incident to exposed centers are queried and each exposed center causes at most 4 queries on the branch τ. Put different, if we let $\tau_g \subseteq \tau$ be the closed path constructed by the end of stage g, then the set of exposed centers $S^*_g = S(\tau_g, \sigma)$ is for the first time of size at least $s/4$. We analyze the event of ever reaching such a stage g.

Note that $|S^*_g| < s/4 + 8t$ by Lemma A.8.7 and $g \leq s/4$ as in each stage at least one center is added to the exposed centers S. For $j \in [g]$ we let the forceable branch of stage j in the decision tree T_{i_j} be denoted by ψ_j, let J_j be the corresponding possible forcing information and $\tau_j \subseteq \tau_g$ be the branch in $\mathcal{T}$ created by the end of stage j. Denote the information set added at stage j by I_j and let $I^*_j = I^*(\tau_j, \sigma)$, or equivalently $I^*_j = \cup_{i=1}^{j} I_i$, be the information set gathered during the first j stages. In the following we explain how to extend the information sets J_j into (usually) closed sets γ_j. Sometimes this extension may fail to produce a closed set γ_j but this happens rarely and hence enough centers are killed in ρ^* to finish the argument.

Consider the sets $J_1, \dots, J_g$ in order. Initially we set $\gamma_j = J_j$ and extend it as follows. Recall that when we create the extended canonical decision tree $\mathcal{T}$, in stage j, we add a set U_j of chosen centers to $S^*_{j-1} = S(\tau_{j-1}, \sigma)$ that are closest endpoints of variables on ψ_j. Add all information pieces in I^*_{j-1} incident to a chosen center in U_j to γ_j. Note that because τ_g is a closed branch and U_j is disjoint from S^*_{j-1}, by Lemma A.8.6, all these added information pieces are non-edges towards chosen centers in S^*_{j-1}.

We need to close the set γ_j. Let us consider each center $v \in \mathrm{supp}(\gamma_j)$ separately. We want to close γ_j at v, meaning that (1) there are information pieces in all directions next to v and (2) an odd number of these edges are present. Note that the non-chosen part in γ_j is already closed as this part is closed in J_j. Hence we only need to add information pieces next to chosen centers and we thus focus on the case when v is a chosen center. We claim that if v has information pieces in all four directions in γ_j, then γ_j is closed at v: since I^*_{j-1} and J_j are consistent (by Property 3) there is an odd number of edges next to v.

Otherwise, if v has no information piece in some direction(s), add a non-edge in all but one such direction to γ_j. In case v already has an odd number of edges next to v, add another non-edge in the final direction. Else we need to add an edge to an appropriately selected center in the suitable sub-square R. At this point we slightly bend the rules and allow to connect the chosen center v to a non-chosen center in R.

Namely, we add an edge from v to a so-called *fresh center* in R, unless there are no fresh centers available. A fresh center is a non-chosen but alive center that is not a member of S^*_g and is not an element of any of the sets $\mathrm{supp}(\gamma_1), \dots, \mathrm{supp}(\gamma_{j-1})$. If we add a fresh center we also add non-edges from the fresh center in the other three directions, ensuring that γ_j is closed. Let us emphasize that we choose which fresh centers to add to γ_j *after* the long branch τ_g has been constructed. This allows us to ensure that these centers do *not* appear in S^*_g.

If there is no such fresh center available in R, then we do not add anything and let γ_j have a center of even degree. Let us call these centers *bad*. This completes the description of the construction of the sets $\gamma_1, \dots, \gamma_g$.

In the following we want to argue that the union of the different γ_j is closed if we disregard the bad centers. We establish this by arguing that the γ_j have pairwise disjoint supports.

Lemma A.8.8. *For $j \neq j'$ it holds that $\mathrm{supp}(\gamma_j) \cap \mathrm{supp}(\gamma_{j'}) = \emptyset$.*

Proof. Let us assume that $j' < j$. By definition (Property 3) J_j and I^*_{j-1} are disjoint but their supports may intersect. As I^*_{j-1} contains information pieces in all directions of every center in S^*_{j-1} (Invariants 5 and 4), the supports of J_j and I^*_{j-1} can only intersect in centers that are not in S^*_{j-1}. Because the support of $J_{j'}$ was added to the set of exposed centers at the end of stage j' we have that $\operatorname{supp}(J_{j'}) \subseteq S^*_{j-1}$. This implies that $\operatorname{supp}(J_{j'})$ does not intersect $\operatorname{supp}(J_j) \cup U_j$ as U_j is disjoint from S^*_{j-1} by definition.

Further, because $U_{j'}$ is a subset of S^*_{j-1} and $\operatorname{supp}(J_j) \cup U_j$ is disjoint from S^*_{j-1}, we have that $U_{j'}$ and $\operatorname{supp}(J_j) \cup U_j$ are disjoint. As the support of γ_j consist of the support of J_j along with U_j and the added fresh centers, we conclude that the support of γ_j and the support of $\gamma_{j'}$ are disjoint. □

The bad centers are the reason that ρ^* is a generalized restriction as defined in Section A.4.3. Before formally defining ρ^* let us bound the number of bad centers in the information sets $\gamma_1, \ldots, \gamma_g$.

Lemma A.8.9. *The number of bad centers in the information sets* $\gamma_1, \ldots, \gamma_g$ *is at most* $O(s/\log n)$.

Proof. Only chosen centers can become bad. For a chosen center in γ_j to become bad, each non-chosen center in a neighboring square either occurs in S^*_g or in one of the supports of γ_i, for $i < j$.

We claim that $\sum_{j \le g} |\operatorname{supp}(\gamma_j)| = O(|S^*_g|)$. This is readily verified: when defining γ_j we start out with the support being $\operatorname{supp}(J_j) \cup U_j$ and then enlarge it by at most a single center per element in the support. Lemma A.8.8 implies in particular that

$$\sum_{j \le g} \left| \operatorname{supp}(J_j) \cup U_j \right| = \left| \bigcup_{j \le g} \operatorname{supp}(J_j) \cup U_j \right| \le |S^*_g| \ , \tag{A.1}$$

and thus the claim follows.

By definition of g and Lemma A.8.7 we have that $|S^*_g| < s/4+8t$. Further, by assumption it holds that $t \le s$ and thus $|S^*_g| + \sum_{j \le g} |\operatorname{supp}(\gamma_j)| = O(s)$. Finally, every square contains $\Omega(\log n)$ non-chosen centers and hence there are at most $O(s/\log n)$ many bad centers. □

As mentioned before, closed graphs can be used to define restrictions with fewer live centers. Let B denote the number of bad centers in the support of the different γ_j and let $\gamma^* = \cup_{j=1}^{g} \gamma_j$. As each γ_j is closed (except at the bad centers) and, by Lemma A.8.8, they have pairwise disjoint supports we conclude that γ^* has at most B bad centers. We define ρ^* to be the restriction defined by ρ composed with the information γ^*. As previously explained, the bad centers cause ρ^* to be a generalized restriction where all centers in $\operatorname{supp}(\gamma^*)$ are now dead. We call these the *disappearing* centers. By Lemma A.8.9 we have at most $B \le O(s/\log n)$ bad centers, while at least $|S^*_g| \ge s/4$ centers disappear.

A.8.6 Encoding ρ

We first need to introduce some more notation. Recall that U_j is the set of closest endpoints of variables on the jth forceable branch ψ_j that are chosen centers but not contained in S^*_{j-1}. We let a_j be the number of closest endpoints of variables on ψ_j that are also in $\mathrm{supp}(J_j) \cup U_j$ and let b_j be the number of additional centers in γ_j, i.e., $b_j = |\mathrm{supp}(\gamma_j)| - a_j$. We let $a = \sum_{j=1}^{g} a_j$, define b similarly and let $c = \left|\mathrm{supp}(I^*_g) \setminus \mathrm{supp}(\gamma^*)\right|$ be the number of centers in the support of I^*_g that do not appear in the support of γ^*. The main goal of this section is to prove the following lemma stating that a restriction ρ that causes the extended canonical decision tree to have a closed path of length at least s can be encoded using few bits, given ρ^* and $T_1, \dots, T_m$. Put different, the mapping from ρ to ρ^* can be inverted with a bit of extra information. Recall that Δ is the number of centers in each sub-square.

Lemma A.8.10. *Suppose we are given ρ^* as well as the decision trees $T_1, \dots, T_m$ each of depth at most t. Then*

$$a \log t + b \log \Delta + c \log \log n + O(a + b + c)$$

many bits are needed to encode ρ.

Before diving into the proof of this lemma let us show how the switching lemma follows from Lemma A.8.10. For the proof of the switching lemma we need one further lemma that relates the parameters a, b and c: it is not so hard to convince oneself that $b + c$ is of order $O(a)$. Indeed, it was shown by Håstad [Hås20] that $b + c$ is bounded by $25a$.

Lemma A.8.11 ([Hås20]). *It holds that $b + c \leq 25a$.*

Proof of Lemma A.7.2. Let us analyze the probability that a random ρ gives rise to a closed branch of length at least s. Let $m = \Delta(n/T)^2$ be the total number of centers and recall that $k = C \log n (n/T)^2$ is the total number of live centers.

Let us first count the number of restrictions ρ that give rise to a closed branch of length at least s. By Lemma A.8.10 this is upper bounded by the number of ways to choose ρ^* times $t^a \Delta^b (\log n)^c A^{a+b+c}$, for some absolute constant A. The number of ways to choose ρ^* is[5] at most $2^{1+r_n} \binom{m}{k-(b+a)} n^{2B}$, where 2^{r_n} is the number of possibilities for the choice of the fixed and preferred variables once the choice of centers is fixed, and B is the number of bad centers.

[5] We sum the binomial coefficient over possible values of $a + b$ but this sequence is exponentially increasing and thus dominated by twice the maximal term.

In order to bound the probability that a restriction gives rise to a closed branch of length at least s we also need to count the number of restrictions ρ. We can count these restrictions in a similar manner as we counted the restrictions ρ^*: there are $2^{r_n}\binom{m}{k}$ many such restrictions. Thus the probability of having a closed branch of length at least s is bounded by

$$\frac{t^a \Delta^b (\log n)^c A^{a+b+c} 2^{1+r_n} \binom{m}{k-(a+b)} n^{2B}}{2^{r_n}\binom{m}{k}} \quad . \tag{A.2}$$

The quotient of the the binomial coefficients can be bounded by

$$\begin{aligned}\prod_{i=0}^{a+b-1} \frac{k-i}{m+i-k} &\leq \left(\frac{k}{m-k}\right)^{a+b} \\ &= \left(\frac{C\log n}{\Delta - C\log n}\right)^{a+b} \\ &\leq \Delta^{-(a+b)}(\log n)^{a+b} A^{a+b} \quad ,\end{aligned} \tag{A.3}$$

for some different constant A. We conclude that the probability of a closed branch of length at least s appearing in the extended canonical decision tree is at most

$$\Delta^{-a}(\log n)^{a+b+c} t^a A^{a+b+c} n^{2B} \quad , \tag{A.4}$$

for a new constant A. Applying Lemma A.8.11 and modifying A again we can bound this by

$$\Delta^{-a}(\log n)^{26a} t^a A^a = (A(\log n)^{26} t \Delta^{-1})^a n^{2B} \quad . \tag{A.5}$$

Finally, as the number of exposed centers is at most $a + b + c$ and the number of queried variables is at most four times the number of exposed centers we have $a + b + c \geq s/4$ and hence $a \geq s/104$ by Lemma A.8.11. By Lemma A.8.9 we have that $n^{2B} \leq 2^{O(s)}$ and we can thus incorporate this factor into the constant A. This concludes the analysis of the probability of the event that a closed branch of length at least s appears in the extended canonical decision tree. Lemma A.7.2 now follows from Lemma A.8.5 and a final modification of the constant A. □

The rest of this section is dedicated to the proof of Lemma A.8.10. On a very high level, we want to remove γ^* from ρ^*. We do this in stages, where in each stage we remove a single γ_j from ρ^* by utilizing the decision trees $T_1, \ldots, T_m$ and reading a bit of extra information. Let us introduce some notation and note a simple observation in order to give a bit more detailed proof outline. For convenience let $I_0^* = \emptyset$, let $\gamma_{\geq j}^* = \cup_{i=j}^{g} \gamma_i$, and let $\rho_{\geq j}^*$ be the restriction obtained from composing ρ with the information $\gamma_{\geq j}^*$, i.e., $\rho_{\geq j}^*$ forces the same variables as $(\rho, \gamma_{\geq j}^*)$ forces.

Recall that the possible forcing information J_j along with I_{j-1}^* determines all variables on the forceable branch ψ_j of stage j. As γ_j extends J_j we observe that $(\rho, I_{j-1}^* \cup \gamma_j)$ traverses ψ_j. Further, as $\gamma_{\geq j}^*$ extends γ_j and is consistent with I_{j-1}^*, it also holds that $(\rho, I_{j-1}^* \cup \gamma_{\geq j}^*)$, or equivalently $(\rho_{\geq j}^*, I_{j-1}^*)$, traverses ψ_j. This observation allows us to pursue the following high level plan.

We proceed in stages $j = 1, \ldots, g$. At the beginning of each stage j we assume that we know the restriction $(\rho_{\geq j}^*, I_{j-1}^*)$. Note that because $I_0^* = \emptyset$ and $\rho_{\geq 1}^* = \rho^*$, we have that $(\rho_{\geq 1}^*, I_0^*)$ forces the same variables as ρ^* and we hence have the necessary information to start at stage $j = 1$. By above observation the restriction $(\rho_{\geq j}^*, I_{j-1}^*)$ traverses the forceable branch ψ_j. Let us assume for now that ψ_j is the first 1-branch traversed, which allows us to identify ψ_j for free. This branch can in turn be used to identify a good fraction of γ_j: as ψ_j is of length at most t we only need to spend $\log t$ bits per variable on ψ_j forced by J_j to identify the corresponding closest center that disappeared. To find the remaining elements of γ_j, along with its graph structure, we use some additional external information. This lets us "remove" γ_j from $\rho_{\geq j}^*$ to obtain $\rho_{\geq j+1}^*$. Before we can proceed with stage $j + 1$ we also need to recover I_j. As a good fraction of the support of I_j is already identified by γ_j we can again use some external information to obtain the final missing pieces.

Unfortunately there are some complications. Recall that when we closed up the information sets γ_j we potentially added information pieces to γ_j that correspond to paths between chosen and non-chosen centers. Such information pieces are *not* allowed in a potential forcing information J_j. So it may well be that the first 1-branch traversed by $(\rho_{\geq j}^*, I_{j-1}^*)$ is different from ψ_j. In order to find the correct forceable branch we introduce signatures.

Definition A.8.12 (signature)**.** Let v be a center in the support of γ_j. The *signature* of any disappearing center v consists of 9 bits. The first bit is 1 iff v is a chosen center. For each of the four directions there is a bit indicating whether v is the closest endpoint of a variable in this direction on the forceable branch ψ_j. For each of the four directions there is also a bit indicating whether there is an information piece in this direction in J_j.

Remark: Note that the stage j, although mentioned in the definition, is *not* part of the signature (as it was in [Hås20]). This change is mandated by our desire to get a tighter bound which requires a smaller signature.

By Lemma A.8.8 the supports of two distinct information sets γ_j and $\gamma_{j'}$ are disjoint and hence each center in the support of γ^* has a unique signature. Also, recall that J_j determines all variables on the forceable branch ψ_j that are not determined by I^*_{j-1}. Hence every variable on ψ_j that is not determined by I^*_{j-1} has a closest endpoint with a signature.

As elaborated previously we use signatures to rule out that a candidate 1-branch is equal to the forced branch ψ_j. Let us define what it means for a signature to be in conflict with a 1-branch and an information set I. To this end observe that a chosen center v along with its signature defines a partial assignment to the incident path variables: the variables in the domain of the partial assignment are all variables in the directions in which v is the closest endpoint of some variable on the forceable branch (according to the first set of four bits) and these variables take values as indicated by the second set of four bits.

Definition A.8.13 (conflict)**.** Let I be an information set, ψ be a branch and E be a set of tuples (v, sign) each consisting of a center v along with the signature sign of v. The set E is in *conflict* with ψ and I iff either

1. there is a tuple $(v, \mathrm{sign}) \in E$ such that the directions in which ψ has variables whose closest endpoint is v do not agree with sign, or

2. the partial assignment on chosen path variables obtained from I jointly with the assignments defined by the signatures $(v, \mathrm{sign}) \in E$, where v is a chosen center and there is a variable on ψ whose closest endpoint is v, is not consistent.

The following lemma states that if a set of signatures is not in conflict, then we have indeed identified the jth forceable branch ψ_j. This is the central lemma of the reconstruction process.

Lemma A.8.14. *Let E be the set of tuples (v, sign) where $v \in \mathrm{supp}(\gamma^*_{\geq j})$ and* sign *is the signature of v. If ψ is the first 1-branch traversed by $(\rho^*_{\geq j}, I^*_{j-1})$ such that E is* not *in conflict with I^*_{j-1} and ψ, then ψ is the jth forceable branch ψ_j.*

Proof. We need to establish that E is in conflict with I^*_{j-1} and all branches ψ before ψ_j. Suppose otherwise and let us construct a possible forcing information J_j that could have been used in stage j of the construction of the extended canonical decision tree to force the branch ψ.

On the non-chosen centers the set J_j contains the pieces of π needed to force all variables on ψ.

On the chosen centers the set J_j consists of information pieces as given by the partial assignments defined by signatures $(v, \mathrm{sign}) \in E$ such that there is a variable on ψ whose closest endpoint is v. These information pieces are consistent with I^*_{j-1} as E is not in conflict with I^*_{j-1} and ψ. Furthermore, these force the input to traverse ψ as these information pieces are the same as used in $\gamma^*_{\geq j}$. □

Before we describe the reconstruction procedure in detail we need a technical definition. Let I^{*-}_{j-1} be I^*_{j-1} except that we remove the information pieces that have at least one of their endpoints in $\mathrm{supp}(\gamma^*_{\geq j})$. Furthermore, let I^-_j be I_j with the same type of pieces taken away. The removed pieces are simple to describe. Recall that $S^*_j = \cup^j_{i=1} S(\tau_i, \sigma)$ is the set of exposed centers at the end of stage j.

Lemma A.8.15. *An information piece in* I^*_{j-1} *that is from a center in* $\mathrm{supp}(\gamma^*_{\geq j})$ *is in the form of a non-edge from a chosen center not in* S^*_{j-1} *in the direction of a chosen center in* S^*_{j-1}.

Proof. According to Lemma A.8.6 the information set I^*_{j-1} consists of a closed graph on S^*_{j-1} jointly with some non-edges from chosen centers not in S^*_{j-1}. Also, by Property 3, all information sets $J_{j'}$ with $j' \geq j$ are pairwise disjoint with I^*_{j-1}.

When extending $J_{j'}$ to $\gamma_{j'}$ we add the set $U_{j'}$ to the support, which may intersect with the support of I^*_{j-1} but is disjoint from S^*_{j-1}. Also, we add the fresh centers to the support but these are by definition disjoint from $S^*_g \supseteq S^*_{j-1}$. Hence no $\gamma_{j'}$ with $j' \geq j$ can intersect the closed part of I^*_{j-1}. The statement follows. □

Hence very few information pieces are in $I^*_{j-1} \setminus I^{*-}_{j-1}$. Furthermore, these information pieces are in some sense redundant – the set $\gamma^*_{\geq j}$ contains the removed information pieces from I^*_{j-1}.

Lemma A.8.16. *Any variable forced by* $(\rho^*_{\geq j}, I^*_{j-1})$ *is also forced by* $(\rho^*_{\geq j}, I^{*-}_{j-1})$.

Proof. By Lemma A.8.15 the pieces removed from I^*_{j-1} are next to centers that disappear in $\rho^*_{\geq j}$. As the information piece is a non-edge in both I^*_{j-1} and $\gamma^*_{\geq j}$ it is forced to the same value. □

Furthermore when considered as assignments on the path variables, even though we do not know the other endpoint we know that a particular path variables is 0. This implies that I^*_{j-1} and I^{*-}_{j-1} are equally powerful when considering consistent values of path variables.

We can finally explain the reconstruction procedure. Throughout the procedure we maintain the following objects. A counter j of the current stage to be reconstructed, the restriction $\rho^*_{\geq j}$, the information set I^{*-}_{j-1}, the exposed centers S^*_{j-1}, and a set E of (prematurely identified) disappearing centers along with their signatures. Initially we set $j = 1$, $\rho^*_{\geq 1} = \rho^*$, and $S^*_0 = I^{*-}_0 = E = \emptyset$. Let us formally define the reconstruction process.

1. Find the next 1-branch ψ traversed by the information $(\rho^*_{\geq j}, I^{*-}_{j-1})$.
2. If ψ and I^{*-}_{j-1} is in conflict with E, then go to Step 1.
3. Read a bit b to determine if there are more disappearing centers to be found as the closest endpoint of a variable on ψ.
4. If $b = 1$, then we read an integer i of magnitude at most t. This identifies the closest endpoint v of the ith variable on ψ as a disappearing center. Read the signature sign of v and add (v, sign) to E. If E is in conflict with ψ and I^{*-}_{j-1}, then go to Step 1. Otherwise repeat Step 3.
5. If $b = 0$, then we have found the forceable branch. Read some external information to determine γ_j and I^-_j (details below). Update $\rho^*_{\geq j}$ to $\rho^*_{\geq j+1}$, I^{*-}_{j-1} to I^{*-}_j and S^*_{j-1} to S^*_j, remove all closest endpoints of ψ from E, and set $j = j + 1$. If $|S^*_j| \geq s/4$, then terminate. Otherwise go to Step 1.

Recall that each exposed center leads to at most 4 queries in the extended canonical decision tree and thus if there is a branch of length s, then this gives rise to a set of exposed centers S^*_g of size at least $s/4$.

Let us note that for each variable identified on the forceable branch we have the signature of its closest endpoint as each such center belongs to E. Also, once we identified I^{*-}_j it is straightforward to recover the set of exposed centers S^*_j.

By Lemma A.8.14 and Lemma A.8.16 we indeed identify the jth forceable branch ψ_j in stage j. All that is left is to explain how to recover γ_j and I^-_j.

We start with the reconstruction of γ_j. We identified all the closest endpoints of variables on ψ_j and we know, by their signature, in which directions they need another center as the other endpoint of an edge. We read the identity of these other endpoints at a cost[6] of at most $\log \Delta$ for each center. This identifies J_j along with some non-edge information pieces next to chosen centers in U_j that are not contained in $\text{supp}(J_j)$. To finalize the description of γ_j we, unless a center is bad, read the identity of the unique fresh centers used to make γ_j closed. This is done at a cost of $\log \Delta$ for each such center. Having identified γ_j we turn to I^-_j. We first have a bit for each element in γ_j to indicate whether it is also an element of I^-_j.

[6] It might be the case that some of these centers were found previously and are part of E or that also the other endpoint is uniquely defined by occurring variable. In either case the cost, including the signature is $O(\log t)$ which is bounded by $\log \Delta$.

Recall that, by definition, any additional center in $\mathrm{supp}(I_j^-)$ does not belong to $\mathrm{supp}(\gamma^*_{\geq j+1})$. Thus any such center is still alive in $\rho^*_{\geq j}$ and can hence be identified using at most $\log\log n + \log 1.01C$ many bits as we know the sub-square to which it belongs.

What remains is to reconstruct the structure of I_j^-. Let us first reconstruct the non-chosen centers. For each non-chosen center in J_j, using $O(1)$ bits, we find out the size of the connected component in π and the directions of each edge. Then we identify the other endpoint of each such edge using $\log\log n + \log 1.01C$ many bits.

For the chosen centers we can again discover the graph part with $O(1)$ bits per center for structure and an integer of magnitude $1.01C\log n$ for the identity. The non-edges not in $\mathrm{supp}(\gamma^*_{\geq j})$ are also reconstructed using $\log\log n + \log 1.01C$ bits for the identity and $O(1)$ bits per center for direction.

Finally, for any center in γ_j we have 4 bits to describe whether the piece of information in the form of a non-edge in any direction should be added to I_j^{*-}.

This concludes the description of the reconstruction and we need to sum up the external information needed.

Recall that a_j is the number of disappearing centers that are discovered through being the closest endpoint of a discovered variable and are part of ψ_j and that b_j is the number of additional centers in γ_j. Furthermore let c_j be the number of centers needed to be discovered in I_j^- after γ_j was discovered. As before we let $a = \sum_{j=1}^{g} a_j$ and define b and c similarly. The following summarizes the amount of external information needed.

- The disappearing centers that are discovered as closest endpoints contribute $a\log t$ many bits.
- The other disappearing centers contribute at most $b\log\Delta$ bits (or less as discussed in Footnote 6).
- The signatures contribute at most $(a+b)\log(A)$ many bits for a constant A: signatures are only needed for disappearing centers.
- The centers discovered to be part of I contribute $c\,(\log 1.01C + \log\log n)$ bits.
- The graph structure of γ^* and I as well as the information which elements of γ_j are included in I_j contributes a factor $(a+b+c)\log B$, for some constant B.

- Throughout reconstruction at most $s + 8t + s/4$ bits b are read. This follows as we can have at most $s+8t$ bits that are 1 (as each time a disappearing variable is discovered, and this is bounded by Lemma A.8.7) and at most s bits that are 0 (as a stage is ended each time and $g \leq s/4$).

As the number of exposed centers is at most $a + b + c$ and the number of queried variables is at most 4 times the number of exposed centers, we have that $a + b + c \geq s/4$. As $t = O(s)$, we see that the final point requires at most $O(a + b + c)$ bits. The lemma follows.

A.9 The multi-switching lemma

The purpose of this section is to prove the multi-switching lemma, restated here for convenience.

Lemma A.7.5 (Multi-switching Lemma)**.** *There are constants A, c_1, and c_2 such that the following holds. Consider formulas F_i^j, for $j \in [M]$ and $i \in [m_j]$, each associated with a decision tree of depth at most t and let $F^j = \vee_{i=1}^{m_j} F_i^j$. Let σ be a random full restriction from the space of restrictions defined in Section A.4. Then the probability that the number of live variables in each center is in the interval $[.99C \log n, 1.01C \log n]$ and $(F^j\lceil_\sigma)_{j=1}^M$ cannot be represented by an ℓ common partial decision tree of depth at most s is at most*

$$M^{s/\ell}\left(A(\log n)^{c_1} t\Delta^{-1}\right)^{s/c_2} .$$

The proof of Lemma A.7.5 follows very much the proof of Lemma A.7.2. The strategy of the proof is essentially as follows.

If F^j is not turned in to a decision tree of depth ℓ, find the branch of in the extended canonical decision tree of length at least ℓ and put the variables on this branch in the common decision tree. Query those variables and some extra variables and recurse.

We again take any ρ for which the lemma fails and with the aid of the formulas we transform it in to ρ^*. This mapping can later be inverted by the use of some extra information. One complication to handle is that the answers to the variables found on the long branch in the extended decision tree of F^j and the the answers on the long branch in the common decision tree to the same variables are different. This leads to the more complicated game analyzed in Section A.3.1.

The common decision tree must consider all the F^js and once such a formula becomes true we need to consider other formulas. As we know from Lemma A.7.2 most F^j can be represented by shallow decision trees and those we can simply ignore. The factor $M^{s/\ell}$ in our bounds comes from all possible ways of choosing s/ℓ formulas that do not turn in to trees of depth at most ℓ. The high level idea to create a common canonical decision tree as follows.

- Set $j = 1, 2 \ldots M$.
- If F^j is represented by a depth ℓ decision tree under the restriction σ jointly with the answers so far in the common decision, proceed with next j.
- Otherwise create the extended canonical decision tree of F^j. Query the variables on the long branch in the common decision tree and also some extra variables. Repeat the step with the same j and these extra answers from the common decision tree.

The extended canonical decision tree is extended in slightly different compared to the standard case, but we keep the same name hoping no confusion arrives.

If our set of formulas does not allow a ℓ common partial decision tree of depth s some branch of the above procedure queries at least s variables. We use this to create a generalized restriction ρ^* exactly as in the standard switching lemma.

We have a set of exposed vertices S that starts out empty and we keep adding vertices to S. Nothing is ever removed. In particular, except for the first formula processed we already have some elements in S. We also have a set of information pieces which, when we start processing F^j contains pieces from π and answers from the common decision tree on path next to chosen centers. We let a "round" denote the processing of a specific F^j. Each round consists of a number of stages similar to the single formula switching.

The extended canonical decision tree for F^j follows closely the extended canonical decision tree in the standard case. We find the next forceable branch and forcing information J_i^j and we expose all vertices in its support. The chosen centers that are exposed are now the simple moves of the adversary in the game in Section A.3.1 and we expose also the vertices picked by P.

To find out whether the forceable branch is followed we get information sets I_i^j consisting of pieces from π and answers from the decision tree of F^j. Of course we also here record answers in the decision tree as edges or two non-edges.

We follow this approach until ℓ new centers have been exposed during the processing of F^j. We know that this happens as $F^j\lceil_\sigma$ cannot be computed by a decision tree of depth ℓ.

Once we have this long branch in the decision tree for F^j we ask all variables on this branch in the common decision tree. We now compare the answers to variables which go between one exposed center and one non-exposed center in the long branch in the decision tree for F^j and in the common decision tree. If these differ then this edge is chosen as an active edge in the completion move of the adversary in the grid game. As values in both decision trees are locally consistent the number of edges they differ at next to any connected component is even and thus it is a legitimate move for the adversary.

We now also expose the nodes in the response of P and ask all question next to these nodes in the common decision tree. These terminates the end of a round.

Note that at the end of this round the available information pieces are given by the answers from the common decision tree and π. The information pieces on the chosen centers used in the m-extended decision tree for F^j are now forgotten. These answers were only used to find the long branch in that decision tree.

Clearly the above process creates an ℓ partial common decision tree and we need to analyze the probability that we get a tree of depth at least s.

As in the standard switching lemma case, the creation of the common decision tree remains the same if we negate the answers, simultaneously in both the extended decision tree and the common decision tree, and the suggested values on the paths between chosen live centers that are exposed. We state this as a lemma.

Lemma A.9.1. *Let σ_1 be obtained from ρ_1 and π and σ_2 from ρ_2 and π where ρ_1 and ρ_2 pick the same set of centers and fixed values. Assume furthermore that the only difference between ρ_1 and ρ_2 is that for each chosen path P there is a bit c_P such that for each grid-edge e on P the preferred values of x_e differ by c_P in ρ_1 and ρ_2. Then the only difference between the common decision decision trees of $(F^j\lceil_{\sigma_1})_{j=1}^M$ and $(F^j\lceil_{\sigma_2})_{j=1}^M$ is the labeling of the internal edges.*

Next we need to define the notion of a closed path in the common decision tree. Informally we want any answer between an exposed center and a non-exposed center to be 0. As we query variables both in the decision tree for F^j and the common decision tree let us be more specific. We require the following questions to have answers 0.

- At any stage in the processing of F^j an edge between a center exposed in this stage and a non-exposed center. This is the answer in the m-extended decision tree.

- At the end of the round any answer between a center exposed in the round and a non-exposed center. This is the answer in the common decision tree.

Note that an edge of the first type, if it remains an edge between an exposed center and a non-exposed center also after the completion of the round, then the answer is 0 also in the common decision tree. Indeed if the answers differ in the decision tree for F^j and the common decision tree, the non-exposed node is exposed by the rules of the game.

Lemma A.9.2. *If the probability that* $(F^j\lceil_\sigma)_{j=1}^M$ *needs a* ℓ *partial common decision tree of depth* s *is at least* q*, then the probability that this happens with a closed execution of length at least* s *is at least* $2^{-s}q$.

Proof. We just need to show that there are locally consistent assignments that gives the required values. By the rules of our combinatorial game, at each stage of the game each connected component is of even size and hence by Lemma A.3.3 we can get border values that are all zero. As each connected component of the complement is of even size we can make the assignment also locally consistent.

Similarly at the end of round. An active edge corresponds to a value that is one in the common decision tree (as it is zero in the F^j decision tree and they are different). The condition that number of active edges next to any component is even is implied by local consistency. □

Once we have set up the machinery the proof parallels the proof in the standard switching case. We need to verify that it works but no new complications arise.

Using fresh centers we again extend J_i^j to make them closed forming information sets γ_i^j. There might be $O(s/\log n)$ centers for which this process fails and this gives $O(s/\log n)$ bad centers as in the standard switching lemma. The restriction ρ^* is obtained by applying these γ_i^j to ρ. We need to specify the information needed to invert this mapping. Each round is very similar to the standard switching of a single formula and we use the following information.

- The identities of which F^j are processed.

- The inverting information for each single formula, F^j, as used in the inversion process in the standard switching lemma.

- The difference in values of variables queried in the decision tree for F^j and the same variables in the common decision tree.
- The identities of the centers exposed at the end of each round.

The inversion process of each round runs completely parallel to the inversion for the standard switching lemma. The information of which F^j to process is here crucial as ρ does force many F^js to constants. We recover the information pieces used in the single formula process. At the end of the round we use the knowledge of the differences to turn this into the information pieces for the common decision tree. We recover the identities of vertices exposed at the end of each round and add these information pieces to our information set before starting the next round.

For the final calculation, as in the standard case, there is a profit of $\Omega(\log(\Delta) - \log\log n - \log t)$ bits for each center discovered as the closest endpoint of a variable on the forceable branch. This corresponds to the simple moves of the adversary in the combinatorial game.

All other exposed centers are retrieved at cost $O(\log\log n)$. The key to the analysis is Lemma A.3.12 that establishes that a constant fraction of all moves are profitable.

Of the extra information needed, only the identities of the processed formulas cannot be absorbed into the constant A or in polylogarithmic factors, and it gives the first factor of the lemma.

A.10 Conclusion

Of course our bounds are not exactly tight so there is always room for improvement. We could hope to get truly exponential exponential for a bounded depth Frege proof, i.e., essentially bounds 2^n where n is the number of variables. Since any formula given by a small CNF has a resolution proof this is the best we could hope for. As our formulas have $O(n^2)$ variables we are off by a square. If one is to stay with the Tseitin contradiction one would need to change the graph and the first alternative that comes to mind is an expander graph. We have not really studied this question but as our current proof relies heavily on properties of the grid; significant modifications are probably needed.

This brings up the question for which probability distributions of restrictions it is possible to prove a (multi) switching lemma. Experience shows that this is possible surprisingly often. It seems, however, that it needs to be done on a case by case basis. Probably it is too much to ask for a general characterization but maybe it could be possible to prove switching lemmas that cover several of the known cases.

Acknowledgments. We are grateful to Mrinal Ghosh, Björn Martinsson and Aleksa Stanković for helpful discussions on the topic of this paper.

References

[Ajt94] M. Ajtai, "The complexity of the pigeonhole principle", *Combinatorica*, vol. 14, no. 4, pp. 417–433, 1994, Preliminary version in *FOCS '88* (cit. on p. 50)

[Ben02] E. Ben-Sasson, "Hard examples for the bounded depth Frege proof system", *Computational Complexity*, vol. 11, no. 3-4, pp. 109–136, 2002 (cit. on p. 50)

[CR79] S. A. Cook and R. A. Reckhow, "The relative efficiency of propositional proof systems", *The Journal of Symbolic Logic*, vol. 44, no. 1, pp. 36–50, 1979, ISSN: 00224812. [Online]. Available: http://www.jstor.org/stable/2273702 (cit. on p. 75)

[FSS84] M. Furst, J. Saxe and M. Sipser, "Parity, circuits and the polynomial-time hierarchy", *Mathematical Systems Theory*, vol. 17, pp. 13–27, 1984 (cit. on p. 50)

[Hak85] A. Haken, "The intractability of resolution", *Theoretical Computer Science*, vol. 39, pp. 297–308, 1985 (cit. on p. 50)

[Hås86] J. Håstad, "Almost optimal lower bounds for small depth circuits", in *Proceedings of the eighteenth annual ACM symposium on Theory of computing*, ser. STOC '86, Berkeley, California, United States: ACM, 1986, pp. 6–20 (cit. on pp. 51, 52)

[Hås14] ——, "On the correlation of parity and small-depth circuits", *SIAM Journal on Computing*, vol. 43, pp. 1699–1708, 2014 (cit. on p. 52)

[Hås20] ——, "On small-depth frege proofs for tseitin for grids", *Journal of the ACM*, vol. 68, pp. 1–31, 2020 (cit. on pp. 49, 51–57, 65, 67, 69, 74–78, 82, 83, 94, 96)

[IMP12] R. Impagliazzo, W. Matthews and R. Paturi, "A satisfiability algorithms for AC^0", in *Proceeding of the 23rd Annual ACM-SIAM Symposium on Discrete Algorithms*, 2012, pp. 961–972 (cit. on p. 52)

[KPW95] J. Krajíček, P. Pudlák and A. Woods, ”An exponential lower bound to the size of bounded depth frege proofs of the pigeonhole principle”, *Random Structures & Algorithms*, vol. 7, no. 1, pp. 15–39, 1995. doi: 10.1002/rsa.3240070103. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/rsa.3240070103. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/rsa.3240070103 (cit. on pp. 50, 74)

[PRT22] T. Pitassi, P. Ramakrishnan and L. Tan, ”Tradeoffs for small-depth frege proofs”, in *2021 IEEE 62nd Annual Symposium on Foundations of Computer Science (FOCS)*, Los Alamitos, CA, USA: IEEE Computer Society, Feb. 2022, pp. 445–456. doi: 10.1109/FOCS52979.2021.00052. [Online]. Available: https://doi.ieeecomputersociety.org/10.1109/FOCS52979.2021.00052 (cit. on pp. 49, 51, 52, 74, 76)

[PBI93] T. Pitassi, P. Beame and R. Impagliazzo, ”Exponential lower bounds for the pigeonhole principle”, *Computational Complexity*, vol. 3, pp. 97–140, 1993, Preliminary version in *STOC ’92* (cit. on p. 50)

[PRST16] T. Pitassi, B. Rossman, R. A. Servedio and L.-Y. Tan, ”Poly-logarithmic Frege depth lower bounds via an expander switching lemma”, in *Proceedings of the Forty-Eighth Annual ACM Symposium on Theory of Computing*, ser. STOC ’16, Cambridge, MA, USA: Association for Computing Machinery, 2016, pp. 644–657, isbn: 9781450341325. doi: 10.1145/2897518.2897637. [Online]. Available: https://doi.org/10.1145/2897518.2897637 (cit. on pp. 51, 52, 76)

[Raz88] A. A. Razborov, ”Bounded-depth formulae over the basis {and, xor} and some combintorial problems (in russian)”, *Problems of Cybernetics. Complexity Theory and Applied Mathematical Logic*, pp. 149–166, 1988 (cit. on p. 51)

[Raz95] ——, ”Bounded arithmetic and lower bounds in boolean complexity”, in *Feasible Mathematics II*. Boston, MA: Birkhäuser Boston, 1995, pp. 344–386, Editors Peter Clote and Jeffrey Remmel, isbn: 978-1-4612-2566-9 (cit. on p. 84)

[RST15] B. Rossman, R. A. Servedio and L. Tan, ”An average-case depth hierarchy theorem for boolean circuits”, in *IEEE 56th Annual Symposium on Foundations of Computer Science, FOCS 2015, Berkeley, CA, USA, 17-20 October, 2015*, 2015, pp. 1030–1048 (cit. on p. 68)

[Sip83] M. Sipser, ”Borel sets and circuit complexity”, in *Proceedings of the fifteenth annual ACM symposium on Theory of computing*, ser. STOC ’83, New York, NY, USA: ACM, 1983, pp. 61–69, ISBN: 0-89791-099-0 (cit. on p. 50)

[Smo87] R. Smolensky, ”Algebraic methods in the theory of lower bounds for boolean circuit complexity”, in *Proceedings of the nineteenth annual ACM symposium on Theory of computing*, ser. STOC ’87, New York, New York, United States: ACM, 1987, pp. 77–82, ISBN: 0-89791-221-7 (cit. on p. 51)

[Tse68] G. S. Tseitin, ”On the complexity of derivation in the propositional calculus”, in *Studies in constructive mathematics and mathematical logic, Part II*, A. O. Slisenko, Ed., 1968 (cit. on p. 50)

[UF96] A. Urquhart and X. Fu, ”Simplified lower bounds for propositional proofs”, *Notre Dame Journal of Formal Logic*, vol. 37, no. 4, pp. 523–544, 1996 (cit. on pp. 50, 74)

[Yao85] A. C. Yao, ”Separating the polynomial-time hierarchy by oracles”, in *Foundations of Computer Science, 1985., 26th Annual Symposium on*, Oct. 1985, pp. 1–10. DOI: 10.1109/SFCS.1985.49 (cit. on p. 50)

Paper B

Perfect Matching in Random Graphs is as Hard as Tseitin

Per Austrin and Kilian Risse

Abstract

We study the complexity of proving that a sparse random regular graph on an odd number of vertices does not have a perfect matching, and related problems involving each vertex being matched some pre-specified number of times. We show that this requires proofs of degree $\Omega(n/\log n)$ in the Polynomial Calculus (over fields of characteristic $\neq 2$) and Sum-of-Squares proof systems, and exponential size in the bounded-depth Frege proof system. This resolves a question by Razborov asking whether the Lovász-Schrijver proof system requires n^{δ} rounds to refute these formulas for some $\delta > 0$. The results are obtained by a worst-case to average-case reduction of these formulas relying on a topological embedding theorem which may be of independent interest.

B.1 Introduction

Proof complexity is the study of certificates of unsatisfiability, initiated by Cook and Reckhow [CR79] as a program to separate NP from coNP. The main goal of this program is to prove size lower bounds on proofs of unsatisfiability of logical formulas. This is a daunting job – indeed we are far from proving general size lower bounds on certificates of unsatisfiability. As an intermediate step we study proof systems with restricted deductive power and prove size lower bounds for such restricted certificates of unsatisfiability. The most studied such proof system is resolution [Bla37] which is fairly well understood by now, see e.g., the proof complexity book by Krajíček [Kra19].

But resolution is by far not the only proof system. A closely related and quite general proof system is the bounded depth Frege proof system [CR79] which manipulates propositional formulas of bounded depth. While we have some results for the bounded depth Frege proof system, in this introduction we instead focus on two other systems as these were the primary motivation behind our work. These are the two proof systems Polynomial Calculus (PC) [CEI96; ABRW04] and Sum-of-Squares (SoS) [Sho87; Par00; Las01]. These proof systems do not rely on propositional logic, like resolution or Frege, but rather on algebraic reasoning and are examples of so-called (semi-)algebraic proof systems (see e.g. [GHP02]).

Both PC and SoS provide refutations of (satisfiability of) a set of polynomial equations $\mathcal{Q} = \{q_i(x) = 0 \mid i \in [m]\}$ over n variables $x_1, \ldots, x_n$. In the case of PC, these polynomials can be over any field $\mathbb{F}$ (finite or infinite), and in the case of SoS, these polynomials are over $\mathbb{R}$. A key complexity measure of a $\mathrm{PC}_{\mathbb{F}}$ or SoS refutation of $\mathcal{Q}$ is its degree, defined as the maximum degree of any polynomial appearing in the refutation. The degree of refuting $\mathcal{Q}$ in $\mathrm{PC}_{\mathbb{F}}$ or SoS, which we denote by $\mathrm{Deg}_{\mathrm{PC}_{\mathbb{F}}}(\mathcal{Q})$ and $\mathrm{Deg}_{\mathrm{SoS}}(\mathcal{Q})$ respectively, is the minimum degree of any $\mathrm{PC}_{\mathbb{F}}$ or SoS refutation of $\mathcal{Q}$. For Boolean systems of equations, meaning that $\mathcal{Q}$ contains the equations $x_i^2 - x_i = 0$ for all $i \in [n]$, strong enough degree lower bounds imply size lower bounds in both $\mathrm{PC}_{\mathbb{F}}$ [CEI96; IPS99] and SoS [AH19], where the size of a refutation is the total number of monomials appearing in it. For finite $\mathbb{F}$ the proof system $\mathrm{PC}_{\mathbb{F}}$ is incomparable to SoS [Raz98; Gri01; GHP02] whereas SoS can simulate $\mathrm{PC}_{\mathbb{R}}$ by the recent result of Berkholz [Ber18].

There is by now a large number of lower bound results for both PC [Raz98; IPS99; BGIP01; AR01; GL10; MN15], and SoS [Gri01; Sch08; MPW15; BHK+16; KMOW17; AH19; Pot20; AGK20], with SoS in particular having received considerable attention in recent years due to its close connection to the Sum-of-Squares hierarchy of semidefinite programming, a powerful "meta-algorithm" for combinatorial optimization problems [BS14].

In this paper we study the power (or lack thereof) of these proof systems when it comes to refuting the perfect matching formula $\mathrm{PM}(G)$ defined over sparse random graphs $G = (V, E)$ on an odd number of vertices. This formula can be viewed as a system of linear equations over $\mathbb{R}$ on a set of Boolean variables: for each edge $e \in E$ there is a variable $x_e \in \{0, 1\}$ (indicating whether the edge is used in the matching) and for each vertex $v \in V$ there is an equation $\sum_{e \ni v} x_e = 1$. Apart from being a natural well-studied problem on its own, the perfect matching formula is interesting because of its close relation to two other widely studied families of formulas, namely the pigeonhole principle (PHP), and Tseitin formulas.

PHP asserts that m pigeons cannot fit in $n < m$ holes (where each hole can fit at most one pigeon). This can be viewed as a bipartite matching problem on the complete bipartite graph with $m + n$ vertices, where each vertex on the large side (with m vertices) must be matched at least once, and each vertex on the small side (with n vertices) can be matched at most once. There are many variants of PHP (see e.g. the survey [Raz02]), and the one closest to the perfect matching formula is the so-called "onto functional PHP", in which each vertex on both sides must be matched exactly once (rather than at least/at most once). Equivalently, this formula is simply the perfect matching formula on a complete bipartite graph with $n + m$ vertices. While most variants of PHP are hard for PC [Raz98; MN15], the onto functional PHP variant is in fact easy to refute in PC over any field [Rii93]. In SoS, all variants of PHP are easy to refute [GHP02].

The Tseitin formula over a graph G claims that there is a subgraph of G such that each vertex has odd degree. As the sum of the degrees of a graph is even, this formula is not satisfiable if G has an odd number of vertices. In contrast to the PHP, the Tseitin formula is (almost) always hard: for $\mathrm{PC}_{\mathbb{F}}$ over fields $\mathbb{F}$ of characteristic distinct from 2 [BGIP01; AR01] and SoS [Gri01] these formulas require linear degree if G is a good vertex expander. We cannot hope to prove degree lower bounds over fields of characteristic 2 as the constraints become linear and we can thus refute the Tseitin formula using Gaussian elimination. As the perfect matching formula $\mathrm{PM}(G)$ implies the Tseitin formula, PC over fields of characteristic 2 can also easily refute $\mathrm{PM}(G)$ for G with an odd number of vertices.

In summary, the perfect matching formula lies somewhere in between PHP and Tseitin, of which the former is easy to refute in SoS (and easy to refute in PC in the onto functional variant), and the latter is hard to refute in SoS (as well as in PC with characteristic $\neq 2$). Hence it is natural to wonder whether SoS or PC requires large degree to refute the perfect matching formula over non-bipartite graphs.

The case of perfect matching in the *complete graph* on an odd number of vertices (sometimes called the "MOD 2 principle") is well-understood in both PC [BGIP01] and SoS [Gri01; Pot17], requiring degree $\Omega(n)$ in both proof systems unless the underlying field of PC is of characteristic 2. For sparse graphs, less is known. Buss et al. [BGIP01] obtained worst-case lower bounds in PC showing that there exist bounded degree graphs on n vertices requiring $\Omega(n)$ degree refutations. This is obtained by a reduction from Tseitin formulas and while the work of Buss et al. predates the current interest in the SoS system, it is not hard to see that the same reduction yields a similar $\Omega(n)$ degree lower bound for SoS (details provided in Section B.7).

However, for random graphs G little is known about the hardness of the perfect matching formula and, e.g., Razborov [Raz17] asked whether it is true that the Lovász-Schrijver hierarchy [LS91] (which is weaker than SoS) requires n^{ε} rounds to refute the perfect matching principle on a random sparse regular graph with high probability.

B.1.1 Our results

We show that indeed the perfect matching principle requires large size on random d-regular graphs (for some constant d) in the Sum-of-Squares, Polynomial Calculus, and bounded-depth Frege proof systems. Our results apply more generally to Tseitin-like formulas defined by linear equations over the reals induced by some graph, so let us now define these.

For a graph $G = (V, E)$ and integer vector $b \in \mathbb{Z}^V$, consider the system of linear equations over the reals having a variable x_e for each $e \in E$, and the equation $\sum_{e \ni v} x_e = b_v$ for each $v \in V$. Let $\mathrm{Card}(G, b)$ denote this system of linear equations along with the Boolean constraints $x_e \in \{0, 1\}$ (viewed as a quadratic equation $x_e^2 - x_e = 0$) for each edge – in Section B.2.2 the encoding is discussed in more detail. Note that $\mathrm{Card}(G, \vec{1})$ corresponds to the perfect matching problem in G and in general $\mathrm{Card}(G, b)$ can be viewed as asserting that G has a "matching" where each vertex is matched exactly b_v times. Note that whenever $\sum_{v \in V} b_v$ is odd, $\mathrm{Card}(G, b)$ is unsatisfiable (since the equations imply $\sum_v b_v = 2 \sum_e x_e$ which is even)[1].

We focus on the special case of $\mathrm{Card}(G, b)$ where G is d-regular and $b = \vec{t} = (t, t, \ldots, t)$ is the all-t vector for some $t \in [d]$. If in this scenario both n and t are odd (implying d is even) then as observed above $\mathrm{Card}(G, \vec{t})$ is unsatisfiable. On the other hand if n is odd and t is even then $\mathrm{Card}(G, \vec{t})$ is

[1] As pointed out to us by Aleksa Stanković, decidability of $\mathrm{Card}(G, b)$ is in polynomial time: starting with the all 0 assignment, iteratively build up an assignment that may match some vertices fewer times than required. If there is a satisfying assignment, then there is always an augmenting path along which the current assignment can be improved, i.e., more edges set to 1, by a similar argument as for matchings [Ber57]. Such a path can be found in polynomial time by an adaptation of the blossom algorithm [Edm65].

always satisfiable (because such G admits a 2-factorization). The remaining case when n is even may be either satisfiable or unsatisfiable, but for a random d-regular G with $d \geq 3$, $\mathrm{Card}(G, \vec{t})$ will be satisfiable with high probability (because such G can be partitioned into perfect matchings with high probability).

If we let $\mathcal{F}_D$ denote a Frege system restricted to depth-D formulas (see Section B.2.1), then our main theorem is as follows.

Theorem B.1.1. *There is a constant* d_0 *such that for all constants* $d \geq d_0$ *and* $t \in [d]$, *the following holds asymptotically almost surely over a random* d*-regular graph* G *on* n *vertices.*

1. $\mathrm{Deg}_{\mathrm{PC}_{\mathbb{F}}}(\mathrm{Card}(G, \vec{t})) = \Omega(n/\log n)$ *for any fixed field* $\mathbb{F}$ *with* $\mathrm{char}(\mathbb{F}) \neq 2$.

2. $\mathrm{Deg}_{\mathrm{SoS}}(\mathrm{Card}(G, \vec{t})) = \Omega(n/\log n)$.

3. *There is a* $\delta > 0$ *such that* $\mathrm{Size}_{\mathcal{F}_D}(\mathrm{Card}(G, \vec{t})) = \exp\left(\Omega(n^{\delta/D})\right)$, *for all* $D \leq \frac{\delta \log n}{\log \log n}$.

The interesting case of the above theorem is when both n and t are odd so that $\mathrm{Card}(G, \vec{t})$ is unsatisfiable; in the other cases $\mathrm{Card}(G, \vec{t})$ is satisfiable with high probability and the lower bounds are vacuous.

By known size-degree tradeoffs for Polynomial Calculus [IPS99; CEI96] and Sum-of-Squares [AH19] the degree lower bounds in Theorem B.1.1 imply near-optimal size lower bounds of $\exp\left(\Omega(n/\log^2 n)\right)$.

Apart from the perfect matching formula, another special case of $\mathrm{Card}(G, \vec{t})$ is the so-called even coloring formula, introduced by Markström [Mar06], which is the case when $t = \deg(v)/2$. An open problem of Buss and Nordström [BN20, Open Problem 7.7] asks whether these formulas are hard on spectral expanders for Polynomial Calculus over fields of characteristic $\neq 2$. Theorem B.1.1 partially resolves this open problem, establishing that it is hard on random graphs (rather than on all spectral expanders). See Section B.6 for some further remarks on what parts of our proof use the randomness assumption.

We will give a more detailed overview of how the results are obtained in Section B.1.3 below, but for now let us mention that we obtain them using embedding techniques, as introduced to proof complexity by Pitassi et al. [PRST16] (see discussion of related work in Section B.1.2). In particular for, say, the SoS lower bound, our starting point is the $\Omega(n)$ *worst-case* degree lower bound in sparse graphs, and we then prove that these hard instances can be embedded in a random d-regular graph in such a way that the hardness of refuting the formula is preserved.

To achieve this, one of the components we need is a new graph embedding theorem which may be of independent interest. Very loosely speaking, we show that any bounded-degree graph with $O(n/\log n)$ edges can be embedded as a *topological minor* in any bounded-degree α-expander on n vertices and sufficiently many edges. In addition, for our application to perfect matching (and more generally the $\mathrm{Card}(G, \vec{t})$ formulas), we need to be able to control the parities of the path lengths used in the topological embedding, and we show that as long as every large linear-sized subgraph contains an odd cycle of length $\Omega(1/\alpha)$, this is indeed possible.

Somewhat informally, we prove the following.

Theorem B.1.2 (Informal statement of Theorem B.3.3)**.** *Let G be a constant degree α-expander on n vertices. If H is a graph with at most $\frac{\varepsilon n}{\log n}$ edges and $\Delta(H) \ll \alpha^2 \cdot d(G)$, then G contains H as a topological minor. Furthermore, if all large vertex induced subgraphs of G contain an odd cycle of length $\Omega(1/\alpha)$, then one can choose the parities of the length of all the edge embeddings in the minor.*

This generalizes various classical results of a similar flavor (e.g. [KR96; KN19; CN19; Kri19]). See the next subsection for a discussion comparing these (and other) existing embedding results to ours.

As a further illustration of the applicability of this theorem we partially resolve a question of Filmus et al. [FLM+13]. They prove that with high probability for random d-regular graphs G, where $d \geq 4$, PC requires *space* $\Omega(\sqrt{n})$ to refute the Tseitin formula, and conjecture that PC in fact requires space $\Omega(n)$. On the other hand, Galesi et al. [GKT19] considered it plausible that the $\Omega(\sqrt{n})$ bound is optimal. We (almost) resolve this question by proving $\Omega(n/\log n)$ space lower bounds for the Tseitin formula defined on vertex expanders, but only of large enough (constant) average degree.

Theorem B.1.3. *For all $\alpha > 0$ there is a d_0 such that the following holds. Let G be a bounded degree α-expander on n vertices of average degree at least d_0. Then over any field $\mathbb{F}$ it holds that $\mathrm{PC}_{\mathbb{F}}$ requires space $\Omega(n/\log n)$ to refute the Tseitin formula defined on G.*

Let us mention that the constant hidden in the lower bound $\Omega(n/\log n)$ depends on the maximum degree of G. Unlike Theorem B.1.1, vertex expansion is sufficient and we require no randomness. This lower bound is obtained by embedding a worst-case instance, due to Filmus et al., into a vertex expander. We provide more details in Section B.6.1.

B.1.2 Related work

Proof Complexity Lower Bounds Using Embedding Techniques There are a few other papers that employ embedding techniques in proof complexity [PRST16; GI19; GIRS19; IRSS19], though none of these use the embedding techniques in connection with algebraic systems like PC or SoS. As far as we are aware the first such work is that of Pitassi et al. [PRST16], who apply embedding techniques to obtain Tseitin lower bounds for Frege Systems, and their use is most similar to ours. They rely on a result of Kleinberg and Rubinfeld [KR96] that guarantees that any small enough graph is a minor of an expander (note that we require *topological* minors). This is in contrast to the other results that rely on the fundamental result that a graph of large enough treewidth contains the grid graph as a minor [RS86].

Connection to Constraint Satisfaction Problems For a k-ary predicate $P : \{0,1\}^k \to \{0,1\}$, an instance of the CSP(P) problem consists of a set of constraints over n Boolean variables $x_1, \ldots, x_n$, each constraint being an application of P on a list of k variables. The $\mathrm{Card}(G, \vec{t})$ formulas we study can be viewed as instances of CSP(P) where each variable appears in exactly two constraints and $P : \{0,1\}^d \to \{0,1\}$ is the constraint that exactly t of the d inputs are 1.

CSP problems have been extensively studied throughout the years, and fairly general conditions under which CSP(P) is hard for PC and SoS are known [AR01; KMOW17]. To be more accurate, these results are for the more general CSP($P^{\pm}$) problem in which each constraint is an application of P on k *literals* rather than variables. In particular, Alekhnovich and Razborov [AR01] showed that if P is, say, 8-immune[2] over the underlying field $\mathbb{F}$, then any $\mathrm{PC}_{\mathbb{F}}$ refutation of a random CSP($P^{\pm}$) instance with a linear number of constraints requires degree $\tilde{\Omega}(n)$. For SoS, Kothari et al. [KMOW17] showed that, if there exists a pairwise uniform distribution[3] μ over $\{0,1\}^k$ supported on satisfying assignments of P, then with high probability a random CSP($P^{\pm}$) instance on $m = \Delta n$ constraints needs degree $\tilde{\Omega}(n/\Delta^2)$ to be refuted by the SoS proof system.

The predicates we study are linear equations over $\mathbb{R}$ and are neither immune nor do they support a pairwise uniform distribution. As such, our results provide CSP lower bounds that fall outside the immunity and pairwise independence frameworks, which are the source of a majority of existing CSP lower bounds in PC and SoS. To the authors' best knowledge the only other attempt to overcome this framework in the average-case setting is the paper by Deshpande et al. [DMO+19], showing lower bounds for the basic SDP of random regular instances of CSP($\mathrm{NAE}_3^{\pm}$), where NAE_3

[2] P is r-immune over $\mathbb{F}$ if there is no degree-r polynomial $q : \{0,1\}^k \to \mathbb{F}$ such that for all satisfying assignments $\alpha \in \{0,1\}^k$ of P it holds that $q(\alpha) = 0$.

[3] A distribution μ over $\{0,1\}^k$ is said to be pairwise uniform if for all $1 \le i < j \le k$, the marginal distribution of μ restricted to coordinates i and j is uniform.

is the not-all-equal predicate on three bits. In contrast to their work we show (almost) linear degree lower bounds for the stronger Sum-of-Squares hierarchy, but only for a very wide predicate of some large (but constant) arity.

Embedding Theorems There is a rich literature on embeddings of graphs as minors or topological minors into expander graphs. We focus here on the ones most closely related to Theorem B.1.2.

The classical result of Kleinberg and Rubinfeld [KR96] shows that a regular expander G on n vertices contains every graph H with $O(n/\text{polylog}(n))$ vertices and edges as a minor. Krivelevich and Nenadov [KN19] simplified and strengthened this by improving the bound on the size of H to $O(n/\log n)$. These results differ from ours in two key ways: (i) we want topological minors, and (ii) we want to be able to control the parities of the path lengths in the embedding. We now discuss these two aspects separately.

Results on topological minors, while somewhat less common, also exist. A result similar to ours is the result of Broder et al. [BFSU96] that with high probability the random graph $\mathcal{G}(n, m)$ on n vertices and $m = \Omega(n \log n)$ edges contains any graph H with $\Delta(H) = O(m/n)$ and at most $O(n/\log n)$ edges (and at most $n/2$ vertices) as a topological minor.

For our second property, the possibility to choose the parities of the paths used in the topological embedding, we are not aware of any previous work studying this question. A related notion are so called *odd minors* which are more general than topological minors with odd length paths. This notion has been considered in connection with a strengthening of Hadwiger's Conjecture, see e.g., the survey by Seymour [Sey16]. This line of research mostly considers complete odd minors, e.g., [GGR+09], and thus is not directly applicable to our situation.

Recently Draganić et al. [DKN20] independently obtained a new embedding theorem similar to ours. They assume the somewhat stronger property that the host graph G is a spectral expander but also obtain a stronger conclusion: each path of the topological embedding is of equal (odd) length and the embedding even works in an adversarial setting. Namely, the adversary is allowed to fix the embedding of the vertices, as long as no neighborhood in G contains too many vertex embeddings.

The embedding theorem of Draganić et al. can be used to implement our proof strategy. The results are unaffected by this change except in the setting of Theorem B.1.3. There, instead of considering vertex expanders, we need to consider regular spectral expanders with the benefit that the required average degree d_0 is considerably decreased.

Extended Formulations There has been a fair amount of work studying the *extension complexity* of the perfect matching polytope [Yan88; Rot17], but these lower bounds do not have any direct implications for the PC and SoS degree of the perfect matching formula. Let us elaborate.

Suppose we have a convex polytope $\mathcal{P}$ consisting of many facets. A natural question is whether there is simpler polytope $\mathcal{Q}$ in a higher dimensional space so that $\mathcal{P}$ is the "shadow" of $\mathcal{Q}$, or a bit more formally that there is a linear projection π such that $\pi(\mathcal{Q}) = \mathcal{P}$. Such a $\mathcal{Q}$ is then called a linear extension of $\mathcal{P}$ and the extension complexity of a polytope $\mathcal{P}$ is the minimum number of facets of any linear extension of $\mathcal{P}$.

Rothvoss [Rot17] proved that the perfect matching polytope of a complete n-node graph has extension compexity $\exp(\Omega(n))$ for n even. This result is incomparable to our lower bounds: as the graphs we consider do not contain a perfect matching, their perfect matching polytope is empty and thus has extension complexity 0. Rather than linear programs, i.e., polytopes, we consider semidefinite programs which are more expressive. The extension complexity in the semidefinite setting has also been studied before [LRS15; BBH+17] but these results are incomparable for the same reason just mentioned. While these results are incomparable, it is worth mentioning that there is a connection between Sherali-Adams (a proof system weaker than SoS) and extended formulations [CLRS16; KMR17].

B.1.3 Overview of Proof Techniques

As previously mentioned, our high level approach is to first obtain worst-case perfect matching lower bounds and to then embed these into the $\mathrm{Card}(G, \vec{t})$ formula for G a random regular graph. The worst-case lower bounds are obtained by a gadget reduction from Tseitin to perfect matching, due to Buss et al. [BGIP01]. Using known lower bounds for the Tseitin formula in the corresponding proof systems [BGIP01; Gri01; Hås20] we then obtain the desired worst-case lower bounds for the perfect matching formula.

A naïve attempt to obtain average-case lower bounds from a sparse worst-case instance H on n vertices is to topologically embed the worst-case instance into a random regular graph G on $O(n \log n)$ vertices using Theorem B.1.2. One would then like to argue that $\mathrm{PM}(G)$ is hard.

Suppose each path p_{uv} in the embedding of H in G corresponding to some edge $\{u, v\} \in E(H)$ is of odd length. Then it is straightforward to verify that the perfect matching formula defined over the embedding is at least as hard to refute as the worst-case instance $\mathrm{PM}(H)$: map each variable y_e, for $e \in p_{uv}$, alternatingly to x_{uv} or $\bar{x}_{uv}$ such that the first and last edges of p_{uv} are mapped to x_{uv} (using that p_{uv} is of odd length).

This simple projection maps the perfect matching formula defined over the embedding of H to $\mathrm{PM}(H)$ and thus shows that the hardness of $\mathrm{PM}(H)$ should be inherited.

But having such a worst-case instance as a topological minor is *not* sufficient to conclude that $\mathrm{PM}(G)$ is hard. For instance G may contain an isolated vertex and it is then trivial to refute $\mathrm{PM}(G)$. On the other hand if we could guarantee that there is a perfect matching m in the subgraph of G induced by the vertices *not* used in the embedding of H, we can conclude that $\mathrm{PM}(G)$ is hard: hit the formula with the restriction corresponding to the matching m and by the argument from the previous paragraph we are basically left with the worst-case formula.

Thus if we can ensure that H is a topological minor of G with the two additional properties that (i) every path used in the embedding of H has odd length, and (ii) there exists a perfect matching in the subgraph of G induced by the vertices *not* used in the embedding of H, then we obtain average-case lower bounds for the perfect matching formula $\mathrm{PM}(G) \equiv \mathrm{Card}(G, \vec{1})$. The lower bounds for $\mathrm{Card}(G, \vec{t})$ for $t > 1$ can then be obtained by a reduction to the $t = 1$ case: after fixing the value of the edges in $\lfloor t/2 \rfloor$ cycle covers of G to 1, a restriction of $\mathrm{Card}(G, \vec{t})$ is obtained which behaves like $\mathrm{Card}(G', \vec{1})$ for a somewhat sparser random regular graph G'.

Let us elaborate a bit further on the properties required from the topological minor of H in G. As mentioned previously, our embedding theorem can ensure that all paths are of odd length. To ensure the second property, we in fact do not embed H directly into G but rather into a suitably chosen vertex induced subgraph $G[T]$ with the crucial property that for any set of vertices $U \subseteq T$ of odd cardinality the induced subgraph $G[V \setminus U]$ has a perfect matching. As the embedding of H will consist of an odd number of vertices we then obtain property (ii) above. Since we now want to apply Theorem B.1.2 not to G but to $G[T]$, we have to ensure that $G[T]$ satisfies all the conditions of that theorem. We prove what we refer to as the Partition Lemma, which asserts that an induced subgraph $G[T]$ exists that satisfies both the perfect matching property described above, as well as all conditions of Theorem B.1.2. The proof of the Partition Lemma relies primarily on the Lovász Local Lemma and spectral bounds to obtain the desired properties.

For the proof of our embedding theorem (Theorem B.1.2), we extend an argument due to Krivelevich and Nenadov [KN19] (see also [Kri19]) for ordinary minors (rather than topological minors). In order to obtain a minor embedding of H in G, the idea there is to embed the vertices one by one from H in G while maintaining an "unused" subgraph G' of G which is a slightly worse expander than G is. During this process it may happen that some vertex embedding cannot be connected to a neighbor. If this happens, the embedding of that vertex is removed and it needs to be embedded again.

In order to obtain topological embeddings, we need to adapt this procedure. Since we now want vertex-disjoint paths connecting the embedded vertices, we would ideally like to embed each vertex of H as a large star, and then embed the edges of H as paths connecting different leaves of these stars. In order to make this work out, rather than embedding the vertices as actual stars, we embed them as "star-like" subgraphs of G (more precisely defined in Definition B.5.3) that consist of a central vertex connected to many large vertex-disjoint connected subgraphs of G and show (Lemma B.5.4) that we can always embed the vertices of H as such "star-like" subraphs of G.

With this in place, obtaining control of the parities of the path lengths used in the embedding (under the assumption on odd cycles in Theorem B.1.2) is relatively straightforward: almost by definition, when embedding an edge of H into a path of G, we can route it via an odd cycle and can then choose which of the two halves of the odd cycles to use, obtaining two possible embeddings with different path length parity, and can choose the one with the appropriate parity.

B.1.4 Organization

We give some preliminaries in Section B.2, formally defining the used proof systems and encodings used, and recalling some general background results. In Section B.3 we provide most of the proof of Theorem B.1.1 while deferring the proofs of two key results, the aforementioned Partition Lemma and our embedding theorem. The proof of the Partition Lemma is given in Section B.4, and the proof of the embedding theorem can be found in Section B.5.

In Section B.7 we recall the reduction of Buss et al. [BGIP01] from Tseitin to perfect matching and show that it yields lower bounds not only for Polynomial Calculus but also for Sum-of-Squares and bounded depth Frege.

Acknowledgements. The authors are grateful to Susanna de Rezende, Johan Håstad, Jakob Nordström, Dmitry Sokolov and Aleksa Stanković for helpful discussions. In particular we would like to thank Jakob Nordström who suggested to consider the even coloring formula and brought the problem about Polynomial Calculus space to our attention. Finally we would like to thank the anonymous conference and jounral referees for their very detailed reviews and many helpful remarks.

B.2 Preliminaries

Natural logarithms (base e) are denoted by ln, whereas base 2 logarithms are denoted by log. For integers $n \geq 1$ we introduce the shorthand $[n] = \{1, 2, \ldots, n\}$ and sometimes identify singletons $\{u\}$ with the element u. For a set U we denote the power set of U by 2^U and a transversal A of a family of sets $\mathcal{B} = \{B_1, B_2, \ldots B_n\}$ is a set such that there is a bijective function $f : A \to \mathcal{B}$ satisfying that $a \in f(a)$ for all elements $a \in A$.

B.2.1 Proof Systems

Let $\mathcal{P} = \{p_1 = 0, \ldots, p_m = 0\}$ be a system of polynomial equations over the set of variables $X = \{x_1, \ldots, x_n, \bar{x}_1, \ldots, \bar{x}_n\}$. Each p_i is called an axiom, and throughout the paper we always assume $\mathcal{P}$ includes all axioms $x_i^2 - x_i$ and $\bar{x}_i^2 - \bar{x}_i$, ensuring that the variables are boolean, as well as the axioms $1 - x_i - \bar{x}_i$, making sure that the "bar" variables are in fact the negation of the "non-bar" variables.

Sum-of-Squares (SoS) is a static semi-algebraic proof system. An SoS proof of $f \geq 0$ from $\mathcal{P}$ is a sequence of polynomials $\pi = (t_1, \ldots, t_m; s_1, \ldots, s_a)$ such that

$$\sum_{i \in [m]} t_i p_i + \sum_{i \in [a]} s_i^2 = f \quad . \tag{B.1}$$

The *degree* of a proof π is

$$\mathrm{Deg}(\pi) = \max\{\max_{i \in [m]} \deg(t_i) + \deg(p_i), \max_{i \in [a]} 2\deg(s_i)\} \quad . \tag{B.2}$$

An *SoS refutation of* $\mathcal{P}$ is an SoS proof of $-1 \geq 0$ from $\mathcal{P}$, and the SoS degree to refute $\mathcal{P}$ is the minimum degree of any SoS refutation of $\mathcal{P}$: if we let π range over all SoS refutations of $\mathcal{P}$, we can write $\mathrm{Deg}_{\mathrm{SoS}}(\mathcal{P}) = \min_\pi \mathrm{Deg}(\pi)$.

Definition B.2.1 (Pseudoexpectation)**.** A degree d pseudo-expectation for $\mathcal{P}$ is a linear operator $\widetilde{\mathbb{E}}$ on the space of real polynomials of degree at most d, such that

(i) $\widetilde{\mathbb{E}}[1] = 1$,

(ii) $\widetilde{\mathbb{E}}[tp] = 0$ for all polynomials t and $p \in \mathcal{P}$ with $\deg(t) + \deg(p) \leq d$, and

(iii) $\widetilde{\mathbb{E}}[s^2] \geq 0$ for all polynomials s of degree $\deg(s) \leq d/2$.

It is easy to check that if there is a degree d pseudo-expectation for $\mathcal{P}$, then there is no SoS refutation of $\mathcal{P}$ of degree at most d: if $\widetilde{\mathbb{E}}$ is applied to both sides of (B.1), where $f = -1$, then the right side is equal to -1 while the left is greater than or equal to 0.

The size of an SoS refutation π, $\mathrm{Size}(\pi)$, is the sum of the number of monomials in each polynomial in π and the size of refuting $\mathcal{P}$ is the minimum size over all refutations $\mathrm{Size}_{\mathrm{SoS}}(\mathcal{P}) = \min_\pi \mathrm{Size}(\pi)$.

Polynomial Calculus is a dynamic proof system operating on polynomial equations over a field $\mathbb{F}$. Let $\mathcal{P}$ be over $\mathbb{F}$. Polynomial Calculus over $\mathbb{F}$ ($\mathrm{PC}_\mathbb{F}$) consists of the derivation rules

- linear combination $\dfrac{p = 0 \qquad q = 0}{\alpha p + \beta q = 0}$, where $p, q \in \mathbb{F}[X]$ and $\alpha, \beta \in \mathbb{F}$, and
- multiplication $\dfrac{p = 0}{xp = 0}$, where $p \in \mathbb{F}[X]$ and $x \in X$.

A PC refutation of $\mathcal{P}$ is a sequence of polynomials $\pi = t_1, \ldots, t_\ell$ such that $t_\ell = 1$ and each polynomial t_i is either in $\mathcal{P}$ or can be derived by one of the derivation rules from earlier polynomials. The degree of a refutation is the maximum degree appearing in the sequence $\mathrm{Deg}(\pi) = \max_{i \in [\ell]} \mathrm{Deg}(t_i)$ and the $\mathrm{PC}_\mathbb{F}$ degree of refuting $\mathcal{P}$ is the minimum degree required of any refutation $\mathrm{Deg}_{\mathrm{PC}_\mathbb{F}}(\mathcal{P}) = \min_\pi \mathrm{Deg}(\pi)$. Similarly, the size of a refutation π is the sum of the number of monomials in each line of π and the $\mathrm{PC}_\mathbb{F}$ size of refuting $\mathcal{P}$ is the minimum size required of any refutation $\mathrm{Size}_{\mathrm{PC}_\mathbb{F}}(\mathcal{P}) = \min_\pi \mathrm{Size}(\pi)$.

Frege System Let us describe a Frege system due to Shoenfield, as presented in [UF96]. As Frege systems over the basis $\vee$, $\wedge$ and $\neg$ can polynomially simulate each other [CR79], the details of the system are not essential and hold for any Frege system over the mentioned basis.

Schoenfield's Frege system works over the basis $\vee$ and $\neg$. We treat the conjunction $A \wedge B$ as an abbreviation for the formula $\neg(\neg A \vee \neg B)$ and let 0, 1 denote "false" and "true" respectively. If A is a formula over variables $p_1, \ldots, p_m$, and σ maps the variables $p_1, \ldots, p_m$ to formulas $B_1, \ldots, B_m$, then $\sigma(A)$ is the formula obtained from A by replacing the variable p_i with $B_i = \sigma(p_i)$ for all $i \in [m]$.

A *rule* is a sequence of formulas written as $A_1, \ldots, A_k \vdash A_0$. If every truth assignment satisfying all of $A_1, \ldots, A_k$ also satisfies A_0, then the rule is *sound*. A formula C_0 is inferred from $C_1, \ldots, C_k$ by the rule $A_1, \ldots, A_k \vdash A_0$ if there is a function σ mapping the variables $p_1, \ldots, p_m$, over which

$A_0, \dots, A_k$ are defined, to formulas $B_1, \dots, B_m$ such that for all $i \in \{0, \dots, k\}$ it holds that $C_i = f(A_i)$.

The Frege system $\mathcal{F}$ that we consider consists of the following rules:

$$\begin{aligned} &\vdash p \vee \neg p && \text{Excluded Middle,} \\ p &\vdash q \vee p && \text{Expansion rule,} \\ p \vee p &\vdash p && \text{Contraction rule,} \\ p \vee (q \vee r) &\vdash (p \vee q) \vee r && \text{Associative rule,} \\ p \vee q, \neg p \vee r &\vdash q \vee r && \text{Cut rule.} \end{aligned}$$

An *$\mathcal{F}$-refutation* of an unsatisfiable formula $A = C_1 \wedge \ldots \wedge C_m$ is a sequence of formulas $F_1, F_2, \dots, F_\ell$ such that $F_\ell = 0$ and every formula F_i is either one of $C_1, \dots, C_m$ or inferred from formulas $F_{j_1}, \dots, F_{j_k}$ earlier in the sequence by a rule in $\mathcal{F}$. As $\mathcal{F}$ is sound and complete a formula A has a refutation if and only if it is unsatisfiable.

The size of a formula is the number of connectives in the formula and the size of a refutation π, denoted by $\mathrm{Size}(\pi)$, is the sum of the sizes of all formulas in the refutation. The depth of π is the maximum depth of any formula $F \in \pi$. We denote by $\mathcal{F}_d$ the proof system $\mathcal{F}$ restricted to formulas of depth at most d.

B.2.2 Propositional Formulas

As we are only interested in constant degree graphs all our axioms are of constant size. Hence the precise encoding of the axioms is not significant as we can change the encoding in constant size/degree.

As the encoding is not essential, we view a propositional formula $\mathcal{F}$ over the Boolean variables $x_1, \dots, x_n$ as a family of functions $\mathcal{F} = \{f_1, \dots, f_m\}$ where each $f_i : \{0,1\}^n \to \{\mathrm{True}, \mathrm{False}\}$ is a function that depends on a constant number of variables. The formula $\mathcal{F}$ is satisfied by an assignment $\alpha \in \{0,1\}^n$ if under α all functions evaluate to True: $f_i(\alpha) = \mathrm{True}$ for all $i \in [m]$.

For a map $\rho : \{x_1, \dots, x_n\} \to \{0, 1, x_1, \dots, x_n, \bar{x}_1, \dots, \bar{x}_n\}$ and a function $f : \{0,1\}^n \to \{\mathrm{True}, \mathrm{False}\}$, denote by $f\lceil_\rho$ the function defined by $f\lceil_\rho(x_1, \dots, x_n) = f(\rho(x_1), \dots, \rho(x_n))$. We extend this notation to formulas in the obvious way, i.e., $\mathcal{F}\lceil_\rho = \{f_1\lceil_\rho, f_2\lceil_\rho, \dots, f_m\lceil_\rho\}$.

Two formulas $\mathcal{F}$ and $\mathcal{F}'$ are equivalent, denoted by $\mathcal{F} \equiv \mathcal{F}'$ if the formulas are element-wise equivalent, disregarding functions that are constant True. We say that a formula $\mathcal{F}'$ is an *affine restriction of* $\mathcal{F}$ if there is a map $\rho : \{x_1, \dots, x_n\} \to \{0, 1, x_1, \dots, x_n, \bar{x}_1, \dots, \bar{x}_n\}$ such that $\mathcal{F}' \equiv \mathcal{F}\lceil_\rho$. The following lemma states that a formula $\mathcal{F}$ is at least as hard as any of its affine restrictions.

Lemma B.2.2. *Let $\mathcal{F}, \mathcal{F}'$ be formulas such that $\mathcal{F}'$ is an affine restriction of $\mathcal{F}$ and each axiom of $\mathcal{F}$ depends on a constant number of variables. Then,*

(i) *for any field $\mathbb{F}$ it holds that* $\mathrm{Deg}_{\mathrm{PC}_{\mathbb{F}}}(\mathcal{F}) \in \Omega\big(\mathrm{Deg}_{\mathrm{PC}_{\mathbb{F}}}(\mathcal{F}')\big)$*,*

(ii) $\mathrm{Deg}_{\mathrm{SoS}}(\mathcal{F}) \in \Omega\big(\mathrm{Deg}_{\mathrm{SoS}}(\mathcal{F}')\big)$*, and*

(iii) *for all* $d \geq 2$ *it holds that* $\mathrm{Size}_{\mathcal{F}_d}(\mathcal{F}) \in \Omega\big(\mathrm{Size}_{\mathcal{F}_{d+1}}(\mathcal{F}')\big)$.

Proof. Suppose we have a refutation π of $\mathcal{F}$ in one of the mentioned proof systems. We want to show that if we hit the proof with the restriction ρ such that $\mathcal{F}\lceil_\rho \equiv \mathcal{F}'$ then we obtain a proof $\pi' = \pi\lceil_\rho$ of $\mathcal{F}'$.

First we need to ensure that we can derive all the axioms of $\mathcal{F}'$. These may be encoded in a different manner, but as these proof systems are implicationally complete, and each axiom only depends on a constant number of variables, this can be done in constant degree (constant size).

This shows that the SoS degree of the resulting refutation is at most a constant factor larger. For Polynomial Calculus and Frege the statement is readily verified by an inductive argument over the proof. □

For concreteness let us also define the encoding of the formulas that we are interested in.

Perfect Matching and Card $(G, \vec{b})$ The Perfect Matching formula $\mathrm{PM}(G)$ encodes the claim that the graph G contains a perfect matching. For every edge $e \in E(G)$ introduce a boolean variable $x_e \in \{0,1\}$ and add for every vertex $v \in V(G)$ an axiom claiming that precisely one incident edge is set to true. As a polynomial over $\mathbb{R}$, we encode this claim as

$$q_v^{\mathrm{PM}} = \sum_{e \ni v} x_e - 1 \ , \tag{B.3}$$

which is satisfied under an assignment α if $q_v^{\mathrm{PM}}(\alpha) = 0$. Over other fields we encode this as a sum over indicator polynomials (see example for Tseitin below). For the Frege proof system we encode the vertex axiom as the propositional formula

$$q_v^{\mathrm{PM}} = \bigvee_{e \ni v} x_e \wedge \bigwedge_{\substack{e,e' \ni v \\ e \neq e'}} \bar{x}_e \vee \bar{x}_{e'} \ . \tag{B.4}$$

The formula $\mathrm{Card}(G, \vec{b})$ is encoded in a similar fashion: in the polynomial encoding replace the 1 with b_v, whereas in the propositional encoding we let the latter $\wedge$ range over edge-tuples of size $b_v + 1$.

Tseitin Formula The Tseitin formula $\tau(G)$ claims that the edges of the graph G can be labeled by $0, 1$ such that the number of 1-labeled edges incident to any vertex is odd. For every edge $e \in E(G)$ introduce a boolean variable $y_e \in \{0, 1\}$, denote the set of variables corresponding to edges incident to v by $Y_v = \{y_e \mid v \in e\}$ and let $A_v \subseteq \{0, 1\}^{Y_v}$ contain all assignments to the variables Y_v that set an odd number of variables to 1. We encode the claim that an odd number of edges incident to $v \in V(G)$ are set to 1 as the polynomial

$$q_v^\tau = \sum_{\alpha \in A_v} 1\{Y_v = \alpha\} - 1 \ , \tag{B.5}$$

where $1\{Y_v = \alpha\} = \prod_{\substack{y \in Y_v \\ \alpha(y)=1}} y \prod_{\substack{y \in Y_v \\ \alpha(y)=0}} \bar{y}$ is the indicator polynomial that is 1 iff the variables in Y_v are set according to α. As before, we also add the boolean axioms to ensure that the variables take values in $\{0, 1\}$.

For the Frege system we encode the claim that an odd number of edges incident to $v \in V(G)$ is set to 1 as the propositional formula

$$q_v^\tau = \bigvee_{\alpha \in A_v} 1\{Y_v = \alpha\} \ , \tag{B.6}$$

where the indicator is now encoded as the formula $1\{Y_v = \alpha\} = \bigwedge_{\substack{y \in Y_v \\ \alpha(y)=1}} y \wedge \bigwedge_{\substack{y \in Y_v \\ \alpha(y)=0}} \bar{y}$.

B.2.3 Graph Theory

This paper only considers simple, undirected graphs: all graphs have no self-loops nor multiple edges. For a graph $G = (V, E)$ the neighborhood of a vertex $u \in V$ is $N(u) = \{v \in V \mid \{u, v\} \in E\}$, the neighborhood of a set of vertices $U \subseteq V$ is $N(U) = \bigcup_{u \in U} N(u)$ and for sets $U, W \subseteq V(G)$ the neighborhood of U in W is $N(U, W) = N(U) \cap W$. We denote by $\deg(v) = |N(v)|$ the degree of a vertex $v \in V$, by $\Delta(G)$ the maximum degree, $\delta(G)$ the minimum degree and by $d(G)$ the average degree of G. The edges between two vertex sets $U, W \subseteq V$ are denoted by $E(U, W) = \{\{u, w\} \in E \mid u \in U, w \in W\}$. For a set $U \subseteq V$, we denote by $G[U] = (U, E(U, U))$ the *induced subgraph* of U in G. For a set $T \subseteq V$ we also use $G \setminus T$ as a shorthand for the induced subgraph $G[V \setminus T]$. For a path p in G we denote by $|p|$ the number of edges and by $V(p) \subseteq V(G)$ the set of vertices of p. For two vertices vertices $u, v \in V(p)$, we let $p[u, v]$ denote the subpath of p between (and including) the vertices u and v. The distance between two vertices $u, v \in V$ is the length of the shortest path from u to v and the distance between two sets $U, W \subset V$ is the minimum distance between any pair of vertices $u \in U$ and $w \in W$. Let $\operatorname{diam}(G)$ denote the diameter of G, that is, the maximum distance between any two vertices in G. For a vertex set $U \subseteq V$, and an integer $r \in \mathbb{N}$, let $B_r^G(U) \subseteq V(G)$ be the *ball around* U *of radius* r *in* G: $B_r^G(U)$ contains all vertices $v \in V$ that are at distance at most r from U.

A graph G on n vertices is an *α-expander* (has *vertex expansion* α) if for all sets $U \subseteq V(G)$ of size $|U| \leq n/2$ it holds that $|N(U, V\setminus U)| \geq \alpha|U|$. We denote the *uniform distribution over d-regular graphs on n vertices* by $\mathcal{G}(n,d)$ and tacitly assume throughout this paper that nd is even. A graph G contains H as a *topological minor* if there is an injective map $\sigma : V(H) \to V(G)$ and for every $\{u,v\} \in E(H)$ there is a path $p_{uv} \subseteq G$ from $\sigma(u)$ to $\sigma(v)$ that is pairwise vertex-disjoint from all other paths except in the endpoints. The paths p_{uv} are the *edge embeddings* of the minor.

Let us record the well-known fact that vertex expanders have small diameter.

Lemma B.2.3 ([Kri19]). *Let G be an α-expander on n vertices. Then the diameter of G is upper bounded by $\lceil \frac{2(\log n-1)}{\log(1+\alpha)} \rceil + 1 = O_\alpha(\log n)$.*

As this constant will show up in a few places, let $D_\alpha^{\o} = \frac{2}{\log(1+\alpha)} + 3$ and hence $\operatorname{diam}(G) \leq D_\alpha^{\o} \cdot \log n$, if G is an α-expander.

The following lemma states that even if a small set of vertices is removed from a vertex expander, large sets still have many vertices at small distance.

Lemma B.2.4. *Let G be an α-expander on n vertices. Then for all $r \geq 0$ and all disjoint $S, T \subseteq V(G)$ satisfying $|T| \geq \frac{2}{\alpha}|S|$ it holds that $|B_r^{G\setminus S}(T)| \geq \min\{n/2, (1+\alpha/2)^r|T|\}$.*

Proof. Using expansion and $|S| \leq \frac{\alpha}{2}|T| \leq \frac{\alpha}{2}|B_r^{G\setminus S}(T)|$ we have that for all $r \geq 0$

$$|B_{r+1}^{G\setminus S}(T)| \geq (1+\alpha)|B_r^{G\setminus S}(T)| - |S| \geq (1+\alpha/2)|B_r^{G\setminus S}(T)| \ ,$$

unless $B_r^{G\setminus S}(T)$ is already as large as $n/2$. □

A simple consequence of this is that two large sets are connected by short paths even after the removal of a small set of vertices.

Corollary B.2.5. *Let G be an α-expander on n vertices. Then for all sets $S, T, U \subseteq V(G)$ satisfying that $T, U \neq \emptyset$, that $S \cap (T \cup U) = \emptyset$, and $|T|, |U| \geq \frac{2}{\alpha}|S|$ it holds that in $G \setminus S$ the distance between T and U is at most $D_{\alpha/2}^{\o} \log n$.*

Proof. Apply Lemma B.2.4 to S, T and $r = \lceil \frac{\log n}{\log(1+\alpha/2)} \rceil$ to conclude that at distance r from T there are at least $n/2$ vertices in the graph $G \setminus S$. Applying the same argument to U and S, we see that also from U there are at least $n/2$ vertices reachable by length r paths in $G \setminus S$. But this implies that there is a path of length at most $2r + 1 \leq D^{\varnothing}_{\alpha/2} \log n$ between T and U. □

B.2.4 Probabilistic Bounds

We use the following version of the multiplicative Chernoff bound.

Theorem B.2.6 (Chernoff)**.** *Suppose $X_1, \ldots, X_n$ are independent random variables taking values in $\{0, 1\}$. Let X denote their sum and let $\mu = \mathbb{E}[X]$. Then, for every $0 \leq \delta \leq 1$ we have*

$$\Pr[|X - \mu| \geq \delta\mu] \leq 2\exp(-\delta^2\mu/3) \ .$$

We also need a similar bound for Poisson random variables.

Theorem B.2.7 ([MU05], Theorem 5.4)**.** *Let X be a Poisson random variable with parameter μ. If $x > \mu$, then*

$$\Pr[X \geq x] \leq e^{-\mu} \left(\frac{e\mu}{x}\right)^x \ .$$

Finally we also need the following form of the Lovász local lemma.

Lemma B.2.8 (Lovász local lemma; [AS00], Lemma 5.1.1)**.** *Let $A_1, A_2, \ldots, A_n$ be events in an arbitrary probability spacce. A directed graph $D = (V, E)$ on the set of vertices $V = \{1, 2, \ldots n\}$ is called a dependency digraph for the events $A_1, \ldots, A_n$ if for each i, $1 \leq i \leq n$, the event A_i is mutually independent of all the events $\{A_j \mid (i, j) \notin E\}$. Suppose that $D = (V, E)$ is a dependency digraph for the above events and suppose there are real numbers $x_1, \ldots x_n$ such that $0 \leq x_i < 1$ and $\Pr[A_i] \leq x_i \prod_{(i,j)\in E}(1 - x_j)$ for all $1 \leq i \leq n$. Then $\Pr[\wedge_{i=1}^{n} \bar{A}_i] \geq \prod_{i=1}^{n}(1 - x_i)$.*

B.3 Lower Bounds on Average

In this section we establish our main result Theorem B.1.1 giving average-case lower bounds in PC, SoS and bounded depth Frege for the $\mathrm{Card}(G, \vec{t})$ formulas.

B.3.1 Lower Bounds for Perfect Matching

Recall that we aim to prove that any sparse graph H (in particular a graph where $\mathrm{PM}(H)$ is hard to refute) can be topologically embedded into a random graph such that all paths in the embedding have odd length. In order to do this, we need to assume that the graph is far from bipartite (since otherwise H would need to be bipartite as well, and $\mathrm{PM}(H)$ is easy for bipartite graphs). Furthermore our embedding theorem relies on all large induced subgraphs of G having sufficiently large maximum degree. The two following definitions capture that both properties hold for all large induced subgraphs of G.

Definition B.3.1. A graph G on n vertices is (κ, d)-*max-degree-robust* if for all $U \subseteq V(G)$ of size $|U| \geq \kappa n$ it holds that the maximum degree of the induced subgraph $G[U]$ is $\Delta(G[U]) \geq d$.

Definition B.3.2. A graph G on n vertices is (κ, α, ℓ)-*odd-cycle-robust* if for all $U \subseteq V(G)$ of size $|U| \geq \kappa n$ and such that $G[U]$ is an α-expander it holds that the induced subgraph $G[U]$ contains an odd cycle C of length $\ell \leq |C| \leq 3D^{\varnothing}_{\alpha/2} \log n$.

Note that in the latter definition, assuming that $G[U]$ is an α-expander, the diameter of $G[U]$ is at most $D^{\varnothing}_{\alpha} \log(n) \leq D^{\varnothing}_{\alpha/2} \log(n)$ which means that, unless $G[U]$ is bipartite, it certainly has short odd cycles of length at most $1 + 2D^{\varnothing}_{\alpha/2} \log n$. But a priori these may all be shorter than ℓ. The definition asks for short odd cycles of length at least ℓ, at the cost of a slightly worse upper bound on the cycle length.

Both properties are clearly monotone in κ: if the properties hold for some $\kappa_0 > 0$, then they also hold for all $\kappa \geq \kappa_0$. With these definitions at hand we can state our embedding theorem.

Theorem B.3.3 (Embedding Theorem)**.** *For $\alpha > 0$ there are $\epsilon, n_0 > 0$ such that the following holds. Let G be an α-expander on $n > n_0$ vertices, let $k \geq 6$, and let H be a graph on at most $\epsilon n / k \log n$ vertices and edges. If G is $(1-4/k, 550\Delta(H)/\alpha^2)$-max-degree-robust, then G contains H as a topological minor. Furthermore, if G is also $(1 - 2/k, \beta, 1 + 2/\beta)$-odd-cycle-robust, for $\beta = \frac{\alpha}{3(1+\alpha)}$, then one can choose the parities of the lengths of all the edge embeddings in the minor.*

Let us highlight that k may depend on the graph G. We have made no attempt to optimize the constants. The proof of the embedding theorem can be found in Section B.5.

As mentioned before we need to ensure that once we obtain an embedding of the worst-case graph H in G, that there is a matching in the graph G with the embedding of H removed. To ensure this we will in fact not embed H directly in G but rather in a subgraph of G: first we identify a set of vertices $T \subseteq V(G)$ such that no matter what set $U \subseteq T$ of odd cardinality is removed from G, the graph $G \setminus U$ still contains a perfect matching. We then proceed to show that the graph $G[T]$ satisfies all the properties required in order to embed H into it. The following lemma captures these properties[4].

Lemma B.3.4 (Partition Lemma)**.** *There is a d_0 such that for all $d > d_0$ there is an n_0 such that the following holds. Let $n > n_0$ be odd and $G \sim \mathcal{G}(n, d)$. Then, asymptotically almost surely, there is a set $T \subseteq V(G)$ of size $|T| \geq n/8$ such that $G[T]$ is a $1/3$-expander, $(1/2, 1/12, 25)$-odd-cycle-robust, $(1/3, d/32)$-max-degree-robust and for any set $U \subseteq T$ of odd cardinality it holds that $G \setminus U$ has a perfect matching.*

The partition lemma is proved in Section B.4. The constants in Lemma B.3.4 are rather arbitrarily chosen and their precise values are not significant – the interested reader can find the precise dependencies between them in the proof. With Lemmas B.3.4 and B.3.3 at hand, we can now easily state and prove our lower bounds for the perfect matching formula (i.e., the special case $t = 1$ of Theorem B.1.1).

Theorem B.3.5. *There is a d_0 and an $\varepsilon > 0$ such that for all $d > d_0$ the following holds. For n and $n' \leq \frac{\varepsilon n}{\log n}$ both odd, let $G \sim \mathcal{G}(n, d)$ and H be any graph on n' vertices of degree $\Delta(H) \leq 5$. Then, asymptotically almost surely,* $\mathrm{PM}(H)$ *is an affine restriction of* $\mathrm{PM}(G)$.

Using the graphs from Section B.7 (i.e., the graphs from Theorems B.7.4, B.7.3 and B.7.1) as our choice of H and combining Theorem B.3.5 with Lemma B.2.2 finishes the proof of Theorem B.1.1 for the perfect matching formula.

Proof of Theorem B.3.5. Let $G \sim \mathcal{G}(n, d)$ as in the statement. Apply Lemma B.3.4 to G to obtain a set T with the mentioned properties. In order to apply Theorem B.3.3 to $G[T]$ and the graph H to obtain a topological minor $B_H \subseteq G[T]$, where all edge embeddings in B_H are of odd length, we need to check that (for our choice $\alpha = 1/3, k = 6$)

(i) $G[T]$ is a $1/3$-expander,

(ii) $G[T]$ is $(1/3, 550 \cdot 5 \cdot 9)$-max-degree-robust,

(iii) $G[T]$ is $(2/3, 1/12, 1 + 2 \cdot 12)$-odd-cycle-robust, and

(iv) H is a graph on at most $\epsilon n/6 \log n$ vertices and edges, for some $\epsilon > 0$.

[4] For clarity of exposition we say that an event holds for *odd* n asymptotically almost surely as $n \to \infty$ if $n = 2n' + 1$ for some non-negative integer n' and the event holds asymptotically almost surely as $n' \to \infty$.

From the guarantees of Lemma B.3.4 we see that (i) is satisfied, that for d large (ii) holds and also that (iii) holds as odd-cycle-robustness is monotone in the first argument. Lastly, (iv) holds if we let $\varepsilon = \epsilon/6$.

With the topological minor B_H of H in G at hand, we proceed to construct a restriction ρ to argue that $PM(H)$ is an affine restriction of $PM(G)$. As all edge embeddings in B_H are of odd length and the number of vertices in H is odd, we see that $|V(B_H)|$ is odd. Hence Lemma B.3.4 guarantees that there exists a perfect matching M in the graph $G' = G \setminus V(B_H)$. The restriction ρ sets all variables outside of B_H to 0 or 1 depending on whether the edge $e \in M$.

We still need to specify how ρ maps the variables in B_H. For every edge embedding p_{uv} of B_H, choose an arbitrary edge $e_{uv} \in p_{uv}$ and map the edge variables x_e, for $e \in p_{uv}$, alternatingly along p_{uv} to either $x_{e_{uv}}$ or $\bar{x}_{e_{uv}}$ such that the first and last edge of p_{uv} are mapped to $x_{e_{uv}}$ (where we use that $|p_{uv}|$ is odd). By inspection we see that $PM(H)$ is an affine restriction of $PM(G)$ as claimed. □

B.3.2 Lower Bounds for Card $(G, \vec{t})$

In the following we prove the average-case lower bounds on the $\text{Card}(G, \vec{t})$ formulas for $G \sim \mathcal{G}(n, d)$. We consider the special case when n and $t \leq d$ are odd and thus d is even. Without loss of generality, assume that $t \leq d/2$: otherwise "flip" the roles of 0 and 1.

The idea is to split the edge set of the graph G into $\lfloor t/2 \rfloor$ 2-regular graphs $G_1, \ldots, G_{\lfloor t/2 \rfloor}$ and one d_0-regular graph G_0, where $d_0 = d - 2\lfloor t/2 \rfloor$. Then we want to set all variables that correspond to an edge in any of the 2-regular graphs $G_1, \ldots, G_{\lfloor t/2 \rfloor}$ to 1 so that we are left with the perfect matching formula $PM(G_0)$, on which we will embed the worst-case instance of Section B.7.

In order to be able to apply Theorem B.3.5 to $PM(G_0)$, we need to argue that G_0 is a random d_0-regular graph. Also, we need to show that it is in fact possible to decompose a random d-regular graph into $\lfloor t/2 \rfloor$ 2-regular graphs plus a d_0-regular graph. For this, we use the notion of *contiguity*. Intuitively, two sequences of probability measures are contiguous, if all properties that hold with high probability in one also hold with high probability in the other measure.

Definition B.3.6. Let $(P_n)_1^\infty$ and $(Q_n)_1^\infty$ be two sequences of probability measures, such that for each n, P_n and Q_n both are defined on the same measurable space $(\Omega_n, \mathcal{F}_n)$. The two sequences are *contiguous* if for every sequence of sets $(A_n)_1^\infty$, where $A_n \in \mathcal{F}_n$, it holds that

$$\lim_{n\to\infty} P_n(A_n) = 0 \Leftrightarrow \lim_{n\to\infty} Q_n(A_n) = 0 \ .$$

We denote contiguity of two sequences by $P_n \approx Q_n$.

For two random graphs $\mathcal{G}_n$ and $\mathcal{H}_n$ on the same set of n vertices, we denote by $\mathcal{G}_n \oplus \mathcal{H}_n$ the union of two independent samples conditioned on the result being simple. If $\mathcal{G}_n = \mathcal{G}(n, d)$ and $\mathcal{H}_n = \mathcal{G}(n, d')$ are uniform distributions over random regular graphs we can think of this as a proccess where we first sample $G \sim \mathcal{G}_n$ and then repeatedly sample $H \sim \mathcal{H}_n$ until the union of G and H is simple.

Theorem B.3.7 (Corollary 9.44, [JŁR00])**.** *For all constants $d \geq 3$, $m \geq 1$ and $d_1, \ldots, d_m \geq 1$ satisfying $d = \sum_{i=1}^m d_i$ it holds that*

$$\mathcal{G}(n, d_1) \oplus \cdots \oplus \mathcal{G}(n, d_m) \approx \mathcal{G}(n, d) \ .$$

In other words, if we can show that e.g. SoS requires linear degree for a formula over $G \sim \mathcal{G}(n, d_0) \oplus \bigoplus_{i\in\lfloor t/2\rfloor} \mathcal{G}(n, 2)$ with high probability, then this also holds for the same formula over graphs $G \sim \mathcal{G}(n, d)$. Implementing our idea in the former probability distribution is straightforward and we have the following theorem.

Theorem B.3.8. *There is a d_0 and an $\varepsilon > 0$ such that for all $d \geq d_0$ the following holds. Let $n, n' \leq \frac{\varepsilon n}{\log n}$ and $t \in [d]$ all be odd, let $G \sim \mathcal{G}(n, d)$ and H be a graph on n' vertices of degree $\Delta(H) \leq 5$. Then, asymptotically almost surely,* $\mathrm{PM}(H)$ *is an affine restriction of* $\mathrm{Card}(G, \vec{t})$.

Analogously to how Theorem B.3.5 implied the $t = 1$ case of Theorem B.1.1, this theorem implies the general case of Theorem B.1.1.

Proof of Theorem B.3.8. As n is odd d must be even. Note that we may assume that $t \leq d/2$: if $t > d/2$, let us flip the role of 1 and 0 in the formula to obtain $\mathrm{Card}(G, \overrightarrow{d - t})$. Let $d_0 = d - 2\lfloor t/2\rfloor \geq d/2$ and sample

$$G' = G_0 \cup \bigcup_{1\leq i\leq\lfloor t/2\rfloor} G_i \sim \mathcal{G}(n, d_0) \oplus \bigoplus_{1\leq i\leq\lfloor t/2\rfloor} \mathcal{G}(n, 2) \ . \tag{B.7}$$

By Theorem B.3.7, if we show the statement for G', then it also holds for $G \sim \mathcal{G}(n, d)$.

Set all variables in $G_1, \ldots G_{\lfloor t/2\rfloor}$ to 1. When $\mathrm{Card}(G, \vec{t})$ is hit with this restriction we are left with the formula $\mathrm{PM}(G_0)$. As G_0 is distributed according to $\mathcal{G}(n, d_0)$, we may apply Theorem B.3.5 to conclude that $\mathrm{PM}(H)$ is an affine restriction of $\mathrm{Card}(G, \vec{t})$. □

B.4 Proof of the Partition Lemma

In this section we prove Lemma B.3.4, restated here for convenience.

Lemma B.3.4 (Partition Lemma)**.** *There is a d_0 such that for all $d > d_0$ there is an n_0 such that the following holds. Let $n > n_0$ be odd and $G \sim \mathcal{G}(n, d)$. Then, asymptotically almost surely, there is a set $T \subseteq V(G)$ of size $|T| \geq n/8$ such that $G[T]$ is a $1/3$-expander, $(1/2, 1/12, 25)$-odd-cycle-robust, $(1/3, d/32)$-max-degree-robust and for any set $U \subseteq T$ of odd cardinality it holds that $G \setminus U$ has a perfect matching.*

We proceed as follows. First, we partition $V(G) = S \mathbin{\dot\cup} T$ into two sets such that every vertex $v \in V(G)$ has a good fraction of its neighbors in S.

Definition B.4.1. A (c, ε)-*degree-balanced cut* of a graph G is a partition $S \mathbin{\dot\cup} T = V(G)$ of the n vertices of G such that:

(i) $\big||S| - cn\big| \leq \varepsilon n$

(ii) for every vertex $u \in V$, the fraction of u's neighbors that are in S is at least $c - \varepsilon$ and at most $c + \varepsilon$.

It turns out that in random regular graphs any (c, ε)-degree-balanced cut possesses the properties needed in the Partition Lemma, as summarized in the following lemma.

Lemma B.4.2. *For all constants $c, \varepsilon, d > 0$ satisfying $c > 1/2 + \varepsilon$ and $d \geq \max\{(c - 1/2 - \varepsilon)^{-2}, 4 \cdot \varepsilon^{-2}\}$ the following holds. Let n be odd and $G \sim \mathcal{G}(n, d)$. Then, asymptotically almost surely as $n \to \infty$, for any (c, ε)-degree-balanced cut (S, T) of G it holds that*

(i) *the graph G is $\left(\kappa, d\left(\kappa - 2\sqrt{\frac{1-\kappa}{\kappa d}}\right)\right)$-max-degree-robust for all constants $\kappa \in [0, 1]$,*

(ii) *the graph G is $(6/\sqrt{d}, \beta, \ell)$-odd-cycle-robust, for any constants β and ℓ,*

(iii) *the graph $G[T]$ is an α-expander, where $\alpha = \frac{1-c-2\varepsilon}{2(1-c-\varepsilon)}$, and*

(iv) *the graph $G \setminus U$ has a perfect matching for any $U \subseteq T$ of odd cardinality.*

Deferring the proof of this lemma to Section B.4.2, let us first show that (c, ε)-degree-balanced cuts always exist in regular graphs of large enough degree.

Lemma B.4.3. *For all $c \in [0, 1], \varepsilon > 0$ there is a $d_0 \in O\left(\frac{c}{\varepsilon^2} \log^2\left(\frac{c}{\varepsilon^2}\right)\right)$ such that the following holds. For every $d > d_0$, every d-regular graph G has a (c, ε)-degree-balanced cut.*

Proof. Independently include every vertex $v \in V(G)$ in S with probability c. Let A_u denote the bad event that $\big||N(u,S)| - cd\big| \geq \varepsilon d$. By the Chernoff bound (Theorem B.2.6), we have

$$\Pr[A_u] \leq 2\exp(-\varepsilon^2 d/3c)\ . \tag{B.8}$$

Note that the event A_u depends only on A_v for v within distance 2 of u in G, and there are at most d^2 many such v's. We want to apply the Lovász local lemma (Lemma B.2.8) to the events $\{A_v \mid v \in V(G)\}$ and $x_v = x$ for some parameter x. The local lemma conditions then require $\Pr[A_u] \leq x(1-x)^{d^2}$ and this right hand side is maximized at $x = \frac{1}{d^2+1}$ where, using the bound $1 - x = 1 - 1/(d^2+1) \geq e^{-1/d^2}$, it becomes

$$x \cdot (1-x)^{d^2} = \frac{1}{d^2+1} \cdot \left(1 - \frac{1}{d^2+1}\right)^{d^2} \geq \frac{1}{d^2+1} \cdot \frac{1}{e}.$$

For large enough $d = \Omega(\frac{c}{\varepsilon^2}\log(\frac{c}{\varepsilon^2}))$, this is much larger than $\Pr[A_u] \leq 2\exp(-\varepsilon^2 d/3c)$ so by Lemma B.2.8 we conclude that $\Pr[\wedge_{v\in V(G)}\bar{A}_v] > (1-x)^n \geq \exp(-\frac{n}{d^2})$. All that remains is to argue that there is a positive probability that both this happens as well as the size of S being close to cn. In particular if $\Pr\left[\big||S| - cn\big| \geq \varepsilon d\right] < \Pr[\wedge_{v\in V(G)}\bar{A}_v]$, the lemma follows.

By the Chernoff bound (Theorem B.2.6), the cardinality of S is in $[cn \pm \varepsilon n]$ except with probability at most $2\exp(-\varepsilon^2 n/3c)$. Hence it is sufficient that $2\exp(-\frac{\varepsilon^2 n}{3c}) < \exp(-\frac{n}{d^2})$, and for $d \gg \sqrt{c}/\varepsilon$ this clearly holds. This concludes the proof. □

With Lemmas B.4.3 and B.4.2 at hand, proving the Partition Lemma simply boils down to choosing appropriate values for the different constants.

Proof of Lemma B.3.4. Fix $c = 3/4$, $\varepsilon = \kappa = 1/16$ and $\ell = 7$. Let (S,T) be the (c,ε)-degree-balanced cut as guaranteed to exist in G by Lemma B.4.3. The cut (S,T) satisfies all the properties of Lemma B.4.2. Hence all that remains is to verify that the constants were chosen appropriately.

(i) $G[T]$ is $(1/3, d/32)$-max-degree-robust: we have that $|T| \geq (1-c-\varepsilon)n = 3n/16$. Thus if the graph G is $(1/16, d/32)$-max-degree-robust, the statement follows. Observe that for our choice of κ and d large enough (e.g. $d \geq 2^{16}$ suffices) it holds that

$$d\left(\kappa - 2\sqrt{\frac{1-\kappa}{\kappa d}}\right) = d\left(\frac{1}{16} - 2\sqrt{\frac{15}{d}}\right) \geq d/32.$$

(ii) $G[T]$ is $(1/2, 1/12, 25)$-odd-cycle-robust: as we may assume that $d \geq 144$, this property is satisfied.

(iii) $G[T]$ is a 1/3-expander: the expansion α guaranteed by Lemma B.4.2 is

$$\alpha = \frac{1-c-2\varepsilon}{2(1-c-\varepsilon)} = \frac{1/8}{3/8} = 1/3 \ .$$

The statement follows. □

All that remains is to prove Lemma B.4.2. In the following section we recall some results from spectral graph theory needed for the proof of Lemma B.4.2 which is then given in Section B.4.2.

B.4.1 Spectral Bounds

Let us establish some notation and recall some results from spectral graph theory.

We denote the adjacency matrix of a graph G by A_G and by L_G its Laplacian $L_G = D_G - A_G$ (where D_G is the diagonal matrix containing the degrees of the vertices of G). For a matrix $A \in \mathbb{R}^{n \times n}$, denote by $\lambda_1(A) \leq \lambda_2(A) \leq \cdots \leq \lambda_n(A)$ the eigenvalues of A in non-decreasing order.

The *edge expansion* of a graph G on n vertices is

$$\Phi(G) = \min_{\substack{U \subseteq V(G) \\ |U| \leq n/2}} \frac{|E(U, V(G) \setminus U)|}{|U|} \ . \tag{B.9}$$

It is well-known that if the second smallest eigenvalue of the Laplacian is large, then the graph is a good expander. Note that the following theorem does not require that G is regular.

Theorem B.4.4 ([Moh89]). *For all graphs G it holds that $\frac{\lambda_2(L_G)}{2} \leq \Phi(G)$.*

Corollary B.4.5. *All graphs G have vertex expansion $\frac{\lambda_2(L_G)}{2\Delta(G)}$.*

Proof. As every vertex has at most $\Delta(G)$ neighbors, the neighborhood of every set U, satisfying $|U| \leq n/2$, is of size at least $\Phi(G)/\Delta(G)$. The statement follows from Theorem B.4.4. □

Recall that regular random graphs are very good spectral expanders. For the sake of conciseness, let $\lambda = \max\{|\lambda_1(A_G)|, |\lambda_{n-1}(A_G)|\}$.

Theorem B.4.6 ([Fri08]). *Fix $d \geq 3$ and let nd be even. Then, for $G \sim \mathcal{G}(n, d)$ it holds asymptotically almost surely as $n \to \infty$ that $\lambda \leq 2\sqrt{d-1} + o(1)$.*

Another well-known result from spectral graph theory is that the smallest eigenvalue of the adjacency matrix puts a limit on the maximum size of an independent set.

Theorem B.4.7 (Hoffman's bound)**.** *Let* G *be a* d*-regular graph on* n *vertices. If* $S \subseteq V(G)$ *is an independent set of* G*, then*

$$|S| \le -\frac{n \cdot \lambda_1(A_G)}{d - \lambda_1(A_G)} \ .$$

Corollary B.4.8. *Let* G *be a* d*-regular graph on* n *vertices. For any set* $S \subseteq V(G)$ *it holds that if* $|S| > -\frac{2 \cdot n \cdot \lambda_1(A_G)}{d - \lambda_1(A_G)}$*, then* $G[S]$ *is not bipartite.*

Proof. For the sake of contradiction suppose that there is an $S \subseteq V(G)$ such that $G[S]$ is bipartite and $|S| > -\frac{2 \cdot n \cdot \lambda_1(A_G)}{d - \lambda_1(A_G)}$. Let us denote the partition by $S = A \dot{\cup} B$. W.l.o.g., assume that $|A| \ge |S|/2$ and apply Theorem B.4.7 to A to conclude that $-\frac{n \cdot \lambda_1(A_G)}{d - \lambda_1(A_G)} < |A| \le -\frac{n \cdot \lambda_1(A_G)}{d - \lambda_1(A_G)}$. □

Let us recall the mixing lemma; it states that between linearly sized sets of vertices there are about as many edges as expected in a random regular graph.

Lemma B.4.9 (Expander Mixing Lemma [HLW06])**.** *Let* G *be a* d*-regular graph on* n *vertices. Then for all* $S, T \subseteq V(G)$*:*

$$\left| |E(S,T)| - \frac{d|S||T|}{n} \right| \le \lambda \sqrt{|S||T|} \ .$$

We also rely on the following theorem that relates the spectrum of the Laplacian and the existence of a perfect matching.

Theorem B.4.10 ([BH05])**.** *Let* G *be a graph on* n *vertices. If* n *is even and* $\lambda_n(L_G) \le 2\lambda_2(L_G)$*, then* G *has a perfect matching.*

The following statements consider large induced subgraphs $H \subseteq G$. Proposition B.4.13 states that if we have good control of the degrees in H, then we have good control of the spectrum of the Laplacian of H in terms of the spectrum of the adjacency matrix of G. The proof uses Weyl's theorem and Cauchy's interlacing theorem, so let us first state these.

Theorem B.4.11 (Weyl)**.** *Let* $A, B \in \mathbb{R}^{n \times n}$ *be Hermitian. Then, for all* $k \in [n]$*,*

$$\lambda_k(A) + \lambda_1(B) \le \lambda_k(A + B) \le \lambda_k(A) + \lambda_n(B) \ .$$

Theorem B.4.12 (Interlacing Theorem)**.** *Suppose* $A \in \mathbb{R}^{n \times n}$ *is symmetric. Let* $B \in \mathbb{R}^{m \times m}$*, with* $m < n$*, be a principal submatrix. Then, for all* $k \in [m]$*,*

$$\lambda_k(A) \le \lambda_k(B) \le \lambda_{k+n-m}(A) \ .$$

Proposition B.4.13. *Let G be a graph on n vertices and H be an induced subgraph of G with m vertices. Then, for all $k \in [m]$,*

$$\delta(H) - \lambda_{n-k+1}(A_G) \le \lambda_k(L_H) \le \Delta(H) - \lambda_{m-k+1}(A_G) \ .$$

Proof. By Theorem B.4.12, applied to $-A_G$ and $-A_H$, we see that for all $k \in [m]$

$$\lambda_k(-A_G) \le \lambda_k(-A_H) \le \lambda_{k+n-m}(-A_G) \ . \tag{B.10}$$

Note that $\lambda_1(D_H) = \delta(H)$ and $\lambda_m(D_H) = \Delta(H)$. Applying Theorem B.4.11 to D_H and $-A_H$, we conclude that, for all $k \in [m]$

$$\begin{aligned}
\lambda_k(-A_G) + \delta(H) &\le \lambda_k(-A_H) + \lambda_1(D_H) && \text{(B.11)}\\
&\le \lambda_k(D_H - A_H) && \text{(B.12)}\\
&\le \lambda_k(-A_H) + \lambda_m(D_H) \le \lambda_{k+n-m}(-A_G) + \Delta(H) \ . && \text{(B.13)}
\end{aligned}$$

As $\lambda_k(-A_G) = -\lambda_{n-k+1}(A_G)$ and $\lambda_{k+n-m}(-A_G) = -\lambda_{m-k+1}(A_G)$, the statement follows. □

Before commencing with the proof of Lemma B.4.2, let us state two results that are of non-spectral nature. The following is a theorem by Bollobás which captures the distribution of short cycles in random regular graphs.

Theorem B.4.14 ([Bol01], Corollary 2.19)**.** *Let $d \ge 2$ and $k \ge 3$ be fixed natural numbers and denote by $Y_i = Y_i(G)$ the number of i-cycles in a graph $G \sim \mathcal{G}(n, d)$. Then $Y_3, Y_4, \dots Y_k$ are asymptotically independent Poisson random variables with means $\lambda_3, \lambda_4, \dots, \lambda_k$, where $\lambda_i = (d-1)^i/(2i)$.*

Let us also record a simple observation that establishes that shortest odd cycles contain no shortcut.

Lemma B.4.15. *Let G be any graph and suppose that C is a shortest odd cycle in G. Then there is no path p connecting two vertices u, v on C such that both paths of C connecting u to v are longer than p.*

Proof. Suppose such a path p exists. Let $q \subseteq p$ be a subpath of p such that

(i) q only shares its endpoints w_0, w_1 with C, and

(ii) the two paths a_0, a_1 from w_0 to w_1 on C are longer than q.

Note that such a subpath q exists as the two paths connecting u to v on C are both longer than p: if no such path q exists, then each potential q can be replaced by a part of C, thereby obtaining a walk from u to v on C of length at most $|p|$; a contradiction.

But note that such a q gives rise to a shorter odd cycle: either $a_0 \cup q$ or $a_1 \cup q$ is an odd cycle, of length less than C. This is in contradiction to the initial assumption that C is a shortest odd cycle. The statement follows. □

B.4.2 Proof of Lemma B.4.2

Recall that by Theorem B.4.6, with high probability all but the largest eigenvalue of the adjacency matrix of G are bounded in magnitude by $2\sqrt{d-1}+o(1)$. In the following we assume that n is large enough such that the $o(1)$ term is small. Let us argue each property separately.

(i) Let $U \subseteq V(G)$ be any set of size κn. Apply the mixing lemma (Lemma B.4.9) to the graph G to conclude that

$$|E(U, V(G) \setminus U)| \le \kappa n \cdot d\Big((1-\kappa) + 2\sqrt{\frac{1-\kappa}{\kappa d}} + o(1)\Big) \ .$$

As G is a d-regular graph, we conclude that the average degree in $G[U]$ is at least $d(\kappa - 2\sqrt{\frac{1-\kappa}{\kappa d}} - o(1))$. By the observation that if the average degree is at least t, then there is a vertex of degree at least $\lceil t \rceil$, the statement follows for n large enough.

(ii) Recall that a sum of independent Poisson variables $X_1, \ldots, X_k$ with means $\mu_1, \ldots, \mu_k$ is again a Poisson variable with mean $\sum_{i \in [k]} \mu_i$. Hence the number of cycles in G of length at most ℓ is, according to Theorem B.4.14, a Poisson random variable Y with mean

$$\mu = \sum_{i=3}^{\ell} \frac{(d-1)^i}{2i} \le d^\ell/6 \ , \tag{B.14}$$

where we used that $d^{\ell-1} + (d-1)^\ell \le d^\ell$. Theorem B.2.7 then tells us that for any $\gamma > 0$, independent of n, it holds that

$$\Pr[Y \ge \gamma \log n] \le e^{-\mu}\left(\frac{e\mu}{\gamma \log n}\right)^{\gamma \log n} < \frac{1}{n} \ , \tag{B.15}$$

where the strict inequality holds for n large enough. Hence we may assume that $Y < \gamma \log n$. Let $S \subseteq V(G)$ be a set of vertices that contains one vertex from each cycle of length at most ℓ. By assumption $|S| < \gamma \log n$ and the shortest cycle in $G \setminus S$ is of length at least ℓ.

We also know that all but the largest eigenvalue of the adjacency matrix of G are bounded in magnitude by $2\sqrt{d-1}+o(1)$. Apply Corollary B.4.8 to conclude that no subset $W \subseteq V(G)$ of size at least $|W| \geq 5\frac{n}{\sqrt{d}}$ induces a bipartite subgraph, in other words any such $G[W]$ contains an odd cycle.

Let $U \subseteq V(G)$ be of size at least $6\frac{n}{\sqrt{d}}$. For n large, it holds that $G[U \setminus S]$ is of size at least $5\frac{n}{\sqrt{d}}$ and thus contains an odd cycle of length at least ℓ. Let C denote such a cycle. What remains is to show that there is a cycle in $G[U]$ that is simultaneously of length at least ℓ as well as bounded in length by $3D^{\varnothing}_{\beta/2}\log n$.

Towards contradiction suppose that $|C| \geq 3D^{\varnothing}_{\beta/2}\log n$ and that C is a shortest odd cycle in $G[U \setminus S]$. Arbitrarily split the cycle C into four paths A_1, B_1, A_2 and B_2, such that B_1 and B_2 separate A_1 from A_2 on C, both A_is are of size at least $\frac{1}{4}D^{\varnothing}_{\beta/2}\log n$, and both B_is are of size at least $\frac{9}{8}D^{\varnothing}_{\beta/2}\log n$.

We may assume that $\gamma \leq \frac{\beta}{9}D^{\varnothing}_{\beta/2}$ so that for n large we can apply Corollary B.2.5 to $G[U]$, S, A_1 and A_2 to conclude that in $G[U \setminus S]$ there is a path p connecting A_1 to A_2 of length at most $D^{\varnothing}_{\beta/2}\log n$. This contradicts Lemma B.4.15 as both paths of C connecting A_1 to A_2 are of length at least $\frac{9}{8}D^{\varnothing}_{\beta/2}\log n$.

We conclude that there is an odd cycle of length at most $3D^{\varnothing}_{\beta/2}\log n$ in $G[U \setminus S]$. As there are no cycles of length at most ℓ in $G[U \setminus S]$ we see that this cycle is also of length at least ℓ, as required.

(iii) Applying Proposition B.4.13 to G and $G[T]$, we see that

$$\lambda_2(L_{G[T]}) \geq \delta(G[T]) - \lambda_{n-1}(A_G) \ . \tag{B.16}$$

Every vertex $v \in T$ has degree at least $(1-c-\varepsilon)d$ in $G[T]$. Furthermore, as $\lambda_{n-1}(A_G)$ is bounded by $2\sqrt{d-1}+o(1)$ and we assumed that $d \geq 4/\varepsilon^2$, we obtain that $\lambda_2(L_{G[T]}) \geq (1-c-2\varepsilon)d$. Applying Corollary B.4.5, we conclude that $G[T]$ has vertex expansion at least $\frac{1-c-2\varepsilon}{2(1-c-\varepsilon)}$.

(iv) Let $U \subseteq T$ of odd cardinality be as in the statement, and denote by m the number of vertices in $G \setminus U$. By Theorem B.4.10, it is sufficient to establish the bound $\lambda_m(L_{G\setminus U}) \leq 2\lambda_2(L_{G\setminus U})$ on the eigenvalues of the Laplacian of $G \setminus U$. Applying Proposition B.4.13 to $G \setminus U$, we can bound these eigenvalues in terms of the eigenvalues of the adjacency matrix of G, obtaining

$$\lambda_m(L_{G\setminus U}) \leq d - \lambda_1(A_G) \text{ and} \tag{B.17}$$
$$\lambda_2(L_{G\setminus U}) \geq (c-\varepsilon)d - \lambda_{n-1}(A_G) \ . \tag{B.18}$$

As $\lambda_1(A_G)$ and $\lambda_{n-1}(A_G)$ are both bounded in absolute value by $2\sqrt{d-1} + o(1)$ we thus conclude

$$2\lambda_2(L_{G\setminus U}) - \lambda_m(L_{G\setminus U}) \geq (2(c-\varepsilon)-1)d - 2\sqrt{d-1} - o(1).$$

Since $c > 1/2 + \varepsilon$ and we assumed that $d \geq (c - 1/2 - \varepsilon)^{-2}$, we have that $\lambda_m(L_{G\setminus U}) \leq 2\lambda_1(L_{G\setminus U})$ as desired.

B.5 Embedding Theorem

In this section we prove our embedding theorem (Theorem B.3.3). Before starting with the proof, let us establish some notation and recall some facts from graph theory.

B.5.1 Further Graph Theory Preliminaries

In a graph $G = (V, E)$ on n vertices a vertex set $S \subseteq V$ is a *balanced separator in* G if there is a partition $V = A \mathbin{\dot\cup} B \mathbin{\dot\cup} S$ of the vertex set of G such that $|A|, |B| \leq 2n/3$, and G has no edges between A and B.

Large vertex expansion implies that balanced separators are large: the next lemma makes this well-known connection precise.

Lemma B.5.1. *Let* G *be an* α*-expander on* n *vertices, and let* S *be a balanced separator in* G*. Then* $|S| \geq \frac{\alpha n}{3(1+\alpha)}$.

Proof. Let S be a balanced separator in G of size $|S| = s$, separating A and B, with $|A| = a$, $|B| = b$. Without loss of generality assume that $a \leq b \leq 2n/3$. Clearly, $a + s \geq n/3$. Further, $N(A) \subseteq S$, and since $a \leq n/2$, by expansion, we get that $s \geq \alpha a$. In other words, $s/\alpha \geq a$, which when substituted into $a + s \geq n/3$ yields $s(1 + 1/\alpha) \geq n/3$. □

We also require the following lemma on vertex-disjoint paths in expanders.

Lemma B.5.2 ([FK19]). *Let* $G = (V, E)$ *be an* α*-expander and let* $A, B \subseteq V$ *be two vertex sets of sizes* $|A|, |B| \geq t$ *for some* $t > 0$*. Then* G *contains at least* $\frac{t\alpha}{1+\alpha}$ *vertex-disjoint paths between* A *and* B*.*

B.5.2 Proof of Theorem B.3.3

We now proceed with the proof of Theorem B.3.3, restated here for convenience.

Theorem B.3.3 (Embedding Theorem)**.** *For $\alpha > 0$ there are $\epsilon, n_0 > 0$ such that the following holds. Let G be an α-expander on $n > n_0$ vertices, let $k \geq 6$, and let H be a graph on at most $\epsilon n/k \log n$ vertices and edges. If G is $(1-4/k, 550\Delta(H)/\alpha^2)$-max-degree-robust, then G contains H as a topological minor. Furthermore, if G is also $(1-2/k, \beta, 1+2/\beta)$-odd-cycle-robust, for $\beta = \frac{\alpha}{3(1+\alpha)}$, then one can choose the parities of the lengths of all the edge embeddings in the minor.*

When embedding a high degree vertex $x \in V(H)$ into G, we want to find a vertex $v \in V(G)$ of high degree such that many neighbors are connected to large, disjoint sets of vertices. These large sets are very useful as they guarantee that there are many vertices to which we can connect a vertex embedding. The following definition makes this intuition precise.

Definition B.5.3 (Cross)**.** An (r, s)-*cross* in a graph $G = (V, E)$ is a tuple $(v, \mathcal{U})$, where $v \in V$ is a vertex and $\mathcal{U} \subseteq 2^V$ consists of r pairwise disjoint vertex sets $U \subseteq V \setminus \{v\}$, each of size $|U| = s$, such that $N(v) \cap U \neq \emptyset$ and the graph $G[U]$ is connected. We refer to v as the *center* of the cross and to $\mathcal{U}$ as the *branches* of the cross.

The following lemma shows that crosses always exist in expanders with sufficiently large maximum degree.

Lemma B.5.4. *For all $\beta > 0$ and $\gamma = \frac{\beta}{3(1+\beta)}$ the following holds. Let G be a β-expander on n vertices that is $(1-2/k, (1+1/\beta)r)$-max-degree-robust, for some $k \geq 3$ and $r > 0$ such that $r \leq \frac{\gamma^3 n}{k(1+\gamma)}$. Then G contains an (r, s)-cross, for all s that satisfy $r \cdot s \leq \frac{\gamma^2 n}{k(1+\gamma)}$.*

The proof is an adaptation of a proof by Krivelevich and Nenadov [KN19] and is deferred to Section B.5.3. We also have the following lemma which is what allows us to choose the path length parities in the "furthermore" part of Theorem B.3.3. It states that if there is an odd cycle in the graph, then there is an odd and even path between any vertex u and a large enough set A of vertices. Note that this does not necessarily hold if A is too small: the vertex u may have degree 1 and A may be the single neighbor of u. Similarly a lower bound on the length of the odd cycle is needed.

Lemma B.5.5. *For all $\beta > 0$ the following holds. Let G be a β-expander on n vertices that contains an odd cycle of length $\ell \geq 1 + 2/\beta$. Then, for all $u \in V(G)$ and $A \subseteq V(G)$, of size $|A| \geq (D^{\sigma}_{\beta} \log n + 1)(1 + 2/\beta)$, there is a vertex $v \in A$*

such that u *and* v *are connected by both an odd and an even path, each of length at most* $(15D^{\varnothing}_{\beta/2}/\beta)\log n + \ell$.

We defer the proof of Lemma B.5.5 to Section B.5.4.

We now prove Theorem B.3.3 with the assumption of odd-cycle-robustness. Furthermore, the proof makes all paths of odd length, though it is immediate that one can choose the parities. To get the theorem without the assumption of odd-cycle-robustness, one just has to replace the application of Lemma B.5.5 by any shortest path (which, by Lemma B.2.3 is short).

The main idea is due to Krivelevich and Nenadov [KN19] (see also [Kri19]). In contrast to their work we cannot directly embed the vertices into the graph but rather take a detour by embedding appropriately sized crosses for each vertex and then connect branches of crosses that correspond to embeddings of adjacent vertices. The reason for this difference is that the present theorem deals with topological minors rather than plain graph minors (the difference is that in topological minors vertices are connected by vertex disjoint paths while in graph minors subgraphs are connected).

In order for this to work we need to make some further changes to the embedding process used. In their work, three sets of vertices are maintained throughout the process: one set A of "discarded" vertices, one set B of vertices used in the embedding, and the remaining set C of vertices. A key invariant which is maintained is that the set of discarded vertices expand poorly into the set of remaining vertices, which together with expansion implies that not too many vertices can be discarded. In our case, some of the discarded vertices may in fact have good expansion into C, but we can maintain the property that there are not too many such vertices. The details are worked out in what follows. If the verbal description is ambiguous, there is an algorithmic description in Section B.8 (Algorithm 4).

Formally, the algorithm maintains a partition $A \mathbin{\dot\cup} A' \mathbin{\dot\cup} \dot{\bigcup}_{B \in \mathcal{B}} B \mathbin{\dot\cup} C$ of the vertices of G. The sets A, B, and C play the same roles as in the informal description above, and A' is an additional set of discarded vertices which may have large expansion into C. When the algorithm terminates, every vertex $v \in V(H)$ (edge $e \in E(H)$, respectively) has a vertex embedding $B_v \in \mathcal{B}$ (an edge embedding $B_e \in \mathcal{B}$) giving a topological minor of H in G. Initially, all sets except $C = V(G)$ are empty.

Let $\beta = \frac{\alpha}{3(1+\alpha)}$ be the constant from Lemma B.5.1 for the lower bound on the size of a balanced separator in an α-expander. At several points in the algorithm we want to ensure that $G[C]$ is a β-expander. This is achieved by removing any subset $U \subseteq C$ of size $|U| \leq |C|/2$ with small neighborhood $|N(U, C \setminus U)| < \beta|U|$ from C and adding it to A (i.e., letting $C \leftarrow C \setminus U$ and $A \leftarrow A \cup U$). Clearly once there are no sets $U \subseteq C$ left as above, $G[C]$ is a β-expander.

Throughout the algorithm the following invariants are maintained:

(i) C never increases in size and $|C| \geq n(1 - 2/k)$,

(ii) $G[C]$ is a β-expander (by restoring expansion as described above whenever needed),

(iii) $N(A, C) < \beta|A|$, and

(iv) $|A'| < \beta|A|/2$.

The algorithm maintains the set $I \subseteq V(H)$ to keep track of the vertices already embedded.

Let $r(d) = d(1 + 4/\beta) - 1 < 25d/\alpha$ and $s = \left(18D^{\varnothing}_{\beta/2}/\beta\right) \log n$. In what follows we assume ε is sufficiently small as a function of β.

Fix a vertex $x \in V(H) \setminus I$ not already embedded and apply Lemma B.5.4 to $G[C]$ to obtain a $(r(\deg_H(x)), s)$-cross B_x. Remove B_x from C and add it to $\mathcal{B}$ as the vertex embedding of x (set $C \leftarrow C \setminus B_x$ and $\mathcal{B} \leftarrow \mathcal{B} \cup \{B_x\}$), and restore β-expansion in $G[C]$.

Let us check that all the conditions of Lemma B.5.4 are satisfied. First, we need that $G[C]$ is $(1-2/k, (1+3(1+\beta)/\beta)r(\deg_H(x)))$-max-degree-robust. We have $1 + 3(1 + \beta)/\beta \leq 22/\alpha$ and thus

$$r(\deg_H(x))(1 + 3(1 + \beta)/\beta) \leq r(\Delta(H))\frac{22}{\alpha} < \frac{550}{\alpha^2}\Delta(H).$$

Furthermore since G is $(1-4/k, 550\Delta(H)/\alpha^2)$-max-degree-robust and $|C| \geq (1 - 2/k)n$, $G[C]$ is $(1 - 2/k, 550\Delta(H)/\alpha^2)$-max-degree-robust. Second we need to check that

$$r(\deg_H(x)) \leq \frac{\gamma^3|C|}{k(1+\gamma)} \qquad \text{and} \qquad r(\deg_H(x)) \cdot s \leq \frac{\gamma^2|C|}{k(1+\gamma)} \quad ,$$

where $\gamma = \frac{\beta}{3(1+\beta)}$. Since $|C| \geq (1 - 2/k)n$ and $\Delta(H) \leq |V(H)| \leq \frac{\varepsilon n}{k \log n}$ the first bound clearly holds for n large enough, and provided ε is sufficiently small as a function of α the second bound also holds. Thus we can indeed apply Lemma B.5.4 on $G[C]$ with the desired choice of r and s.

After embedding x, we need to connect the embedding B_x to the embeddings of the neighbors $N_H(x) \cap I = \{y_1, \dots, y_v\}$ that are already embedded. Suppose, for now, that the vertex embeddings have branches $U_x \in B_x$ and $U_{y_i} \in B_{y_i}$ that are β-expanding into C (i.e. $|N(U_x, C)|, |N(U_{y_i}, C)| \geq \beta s$), and such that neither of the two branches are already used to connect x, resp. y_i, to a neighbor.

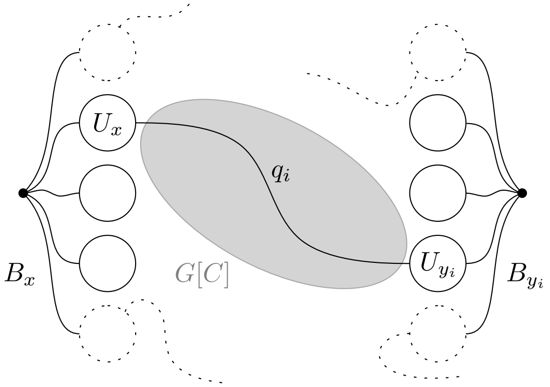


Figure B.1: The vertex embedding B_x is connected to B_{y_i} by the path q_i which connects the two branches U_x and U_{y_i}. The dotted branches have an edge embedding adjacent and can thus not be used to connect B_{y_i} to B_x.

By the assumption on odd-cycle-robustness, we see that $G[C]$ is non-bipartite and contains an odd cycle c of length

$$1 + 2/\beta \leq |c| \leq 3D^{\varnothing}_{\beta/2} \log n \ . \tag{B.19}$$

As each branch is rather large, of size s, we can apply Lemma B.5.5 to $G[C]$, $N(U_x, C)$ and $N(U_{y_i}, C)$ to conclude that in $G[C]$ there is an odd path q_i connecting U_x to U_{y_i} of length $\left(18D^{\varnothing}_{\beta/2}/\beta\right) \log n \leq s$. Remove q_i from C, add it to $\mathcal{B}$ as the edge embedding $B_{\{x,y_i\}}$ and restore β-expansion in $G[C]$. This process is illustrated in Figure B.1 and can be found as pseudo code in Algorithm 4.

If all branches of a vertex embedding B_z have either too few neighbors in C or are already adjacent to an edge embedding (i.e., have already been used to embed some other edge), then we want to remove the embedding of z. This has to be done in a careful manner in order not to break the invariants. First, move all branches that are not used to connect z to a neighbor to A. Note that each such branch U satisfies $|N(U, C)| < \beta|U|$. Next, move the remaining branches along with the adjacent edge embeddings to A'. Last, the center of B_z is moved to A' and z is removed from I. Note that at most $2(\deg_H(z) - 1)s$ many vertices are moved to A': at most $\deg_H(z) - 1$ many branches of size s and as many edge embeddings, each again of size at most s. On the other hand at least

$$\left(r(\deg_H(z)) - (\deg_H(z) - 1)\right) \cdot s = \deg_H(z) \cdot 4s/\beta \tag{B.20}$$

many vertices are moved to A. Hence the invariant $|A'| < \beta|A|/2$ is maintained.

The algorithm terminates the first time either $I = V(H)$ or $|A| \geq n/k$. This completes the description of the algorithm.

It remains to argue that it cannot happen that $|A| \geq n/k$, in other words that when the algorithm terminates, all of H is embedded in G. To this end, observe that the size of $\cup_{B \in \mathcal{B}} B$ is upper bounded by

$$\begin{aligned} s \cdot \left(|E(H)| + \sum_{v \in V(H)} r(\deg_H(v)) \right) &< s \cdot \left(|E(H)| + (4/\beta + 1) \sum_{v \in V(H)} \deg_H(v) \right) \\ &\leq s \cdot |E(H)| \cdot \frac{11}{\beta} \\ &\leq s \cdot \frac{\varepsilon n}{k \log n} \cdot \frac{11}{\beta} \\ &\leq \beta n/2k \ . \end{aligned}$$

Furthermore, while $|A| \leq n/k$ we have that

$$|A'| < \beta|A|/2 \leq \beta n/2k \ .$$

Note that this also holds the first time $|A|$ becomes larger than n/k. This shows, in particular, that the invariant $|C| \geq n(1 - 2/k)$ is maintained throughout the execution of the algorithm.

For the sake of contradiction, suppose that the algorithm terminates because of $|A| \geq n/k$. Note that $|N(A)| \leq |A'| + |\cup_{B \in \mathcal{B}} B| + |N(A, C)| < \beta(|A| + n/k)$. We do a case distinction, depending on the size of A. In both cases we derive contradiction and thus show that the algorithm only terminates after having embedded all of H into G.

Case 1: $n/k \leq |A| \leq n/2$. By expansion and using $\beta < \alpha/3$ we have

$$\alpha|A| \leq |N(A)| < \beta(|A| + n/k) < \frac{\alpha}{3}(|A| + n/k) \ ,$$

which together with $|A| \geq n/k$ yields the desired contradiction.

Case 2: $|A| > n/2$. Note that the first time $|A| \geq n/k$, it also holds that $|A| \leq n(1 + 1/k)/2$ as the sets added to A are of size at most $|C|/2 \leq (n - |A|)/2$. Hence we get that

$$|N(A)| < \beta\big(n/k + |A|\big) < \beta n\big(1/k + (1 + 1/k)/2\big) \leq \frac{\alpha n}{3(1+\alpha)} \ ,$$

using that $k \geq 3$. Note that $N(A)$ is a balanced separator, separating A from $V(G) \setminus A \setminus N(A)$. But this is a contradiction, since Lemma B.5.1 states that any balanced separator of G has size at least $\frac{\alpha n}{3(1+\alpha)}$.

B.5.3 Crosses in Expanders

Let us now turn to the proof of Lemma B.5.4, restated here for convenience.

Lemma B.5.4. *For all $\beta > 0$ and $\gamma = \frac{\beta}{3(1+\beta)}$ the following holds. Let G be a β-expander on n vertices that is $(1-2/k, (1+1/\beta)r)$-max-degree-robust, for some $k \geq 3$ and $r > 0$ such that $r \leq \frac{\gamma^3 n}{k(1+\gamma)}$. Then G contains an (r, s)-cross, for all s that satisfy $r \cdot s \leq \frac{\gamma^2 n}{k(1+\gamma)}$.*

The proof follows a similar algorithm as the proof of Theorem B.3.3. In this case we can in fact more or less use the original argument of Krivelevich and Nenadov [KN19] without any extensions.

Proof. The high-level idea of the proof is as follows. First, using the embedding argument of Krivelevich and Nenadov, we find some number $r' > r$ pairwise disjoint sets $B_1, \ldots, B_{r'}$ of vertices of G and a final set C disjoint from all B_is such that (i) each B_i is a connected subgraph of G on s vertices, (ii) the B_is have many neighbors in C, and (iii) $G[C]$ is expanding. Having these subsets, we can then choose a representative $u_i \in N(B_i, C)$ of each B_i, take a vertex $v \in C$ of high degree (which exists by the max-degree-robustness of G), and apply Lemma B.5.2 to find vertex-disjoint paths connecting $N(v)$ to the u_is. This establishes the existence of a cross with v as the center and the B_is together with the respective paths as branches. See Figure B.2 for an illustration.

Let us proceed with the details. In case there is some ambiguity in the verbal description there is also a pseudo code description in Section B.8 of what follows.

Fix r, set $r' = r(1 + 1/\gamma)$ and choose $s \in \mathbb{N}$ maximal such that $s \leq \frac{\gamma n}{k \cdot r'}$. Note that $s \geq 1/\gamma$ and if the statement holds for this maximal s, then it also holds for smaller values of s, as one can always shrink the branches to the appropriate size.

Let us describe an algorithm to identify the sets $\mathcal{B} = \{B_i \subseteq V(G) \mid i \in [r']\}$. The algorithm maintains a partition $A \,\dot\cup\, \dot\bigcup_{B \in \mathcal{B}} B \,\dot\cup\, C$ of the vertices of G. Initially, all sets except $C = V(G)$ are empty. After running the procedure, the set $\mathcal{B}$ contains r' pairwise vertex-disjoint sets such that for each $B_i \in \mathcal{B}$ it holds that $|B_i| = s$ and the induced subgraph $G[B_i]$ is a single connected component. Further, for all subfamilies $\mathcal{F} \subseteq \mathcal{B}$ it holds that $\left|\bigcup_{F \in \mathcal{F}} N(F, C)\right| \geq \gamma s |\mathcal{F}|$. Throughout the execution of the algorithm the following invariants are maintained

(i) C never increases in size and $|C| \geq n(1 - 2/k)$,

(ii) $G[C]$ is a γ-expander (by restoring expansion whenever needed),

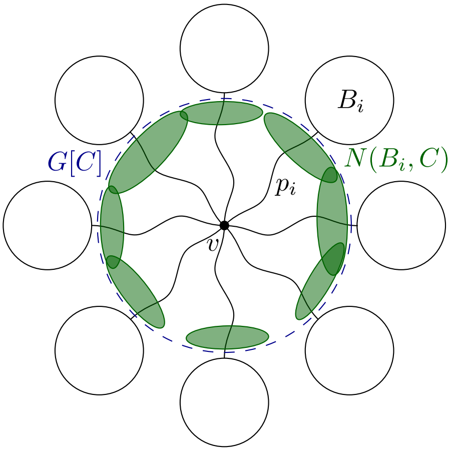


Figure B.2: A cross with center v and branches $\{V(p_i) \cup B_i \mid i \in [r]\}$.

(iii) $N(A, C) < \beta|A|$, and

(iv) $\left|\dot{\bigcup}_{B \in \mathcal{B}} B\right| \leq r' \cdot s \leq \gamma n/k$.

The algorithm terminates if $\mathcal{B}$ contains r' vertex sets as described, or if the size of A reaches $|A| \geq n/k$. The latter case can only occur if there is a small balanced separator in G. But G is a β-expander, so we know from Lemma B.5.1 that there are no small balanced separators and hence when the algorithm terminates, $\mathcal{B}$ must contain r' sets as described above.

Like in the main algorithm used in the proof of Theorem B.3.3, we want to ensure that $G[C]$ is a γ-expander throughout the algorithm, which is achieved by removing any subset $U \subseteq C$ of size $|U| \leq |C|/2$ with small neighborhood $|N(U, C \setminus U)| < \gamma|U|$ from $C = C \setminus U$ and adding it to $A = A \cup U$.

Repeat the following while there are less than r' sets in $\mathcal{B}$. Choose a set of vertices $U \subseteq C$ of size $|U| = s$ such that $G[U]$ is a single connected component. Remove this set from $C = C \setminus U$, add it to $\mathcal{B} = \mathcal{B} \cup \{U\}$ and restore expansion in $G[C]$. After expansion is restored, let $\mathcal{F} \subseteq \mathcal{B}$ be a maximal (possibly empty) family such that $\left|\bigcup_{F \in \mathcal{F}} N(F, C)\right| < \gamma s|\mathcal{F}|$. Remove $\mathcal{F}$ from $\mathcal{B} = \mathcal{B} \setminus \mathcal{F}$, and add these sets to $A = A \cup_{F \in \mathcal{F}} F$.

As mentioned before, the algorithm terminates once there are either r' sets in $\mathcal{B}$ or the set A is large $|A| \geq n/k$. This completes the description of the algorithm. Let us argue that the latter cannot happen – for the sake of contradiction, suppose the algorithm terminates because $|A| \geq n/k$. Note that we have $|N(A)| \leq | \cup_{B \in \mathcal{B}} B| + |N(A, C)| < \gamma(|A| + n/k)$. We do a case distincion on the size of $|A|$.

Case 1: $n/k \leq |A| \leq n/2$. By expansion, $\beta|A| \leq N(A) < \gamma(|A| + n/k) < \frac{\beta}{3}(|A| + n/k)$. As $|A| \geq n/k$ this is a contradiction.

Case 2: $|A| \geq n/2$. Note that the first time $|A| \geq n/k$, it also holds that $|A| \leq n(1 + 1/k)/2$ as the sets added to A are of size at most $|C|/2 \leq (n - |A|)/2$. Hence we get (using $k \geq 3$) that

$$|N(A)| \leq \gamma(n/k + |A|) \leq \gamma n = \frac{\beta n}{3(1+\beta)}.$$

Note that $N(A)$ is a balanced separator, separating A from $V(G) \setminus A$. But this is a contradiction, since Lemma B.5.1 states that any balanced separator of G has size at least $\frac{\beta n}{3(1+\beta)}$.

It remains to obtain an (r, s)-cross from the sets B_i and the remaining part C. Choose a vertex $v \in C$ of degree at least $\deg_{G[C]}(v) \geq r'$. Such a vertex v exists, as $|C| \geq (1 - 2/k)n$ is large (first invariant) and the statement assumes that there is a vertex of degree r' in every induced subgraph of size at least $(1 - 2/k)n$. Let T be a transversal of the family $\{N(B, C) \mid B \in \mathcal{B}\}$. Note that such a transversal T exists by Hall's marriage theorem, using that $s \geq 1/\beta$ and that every subset of $\mathcal{B}$ is β-expanding into C.

Apply Lemma B.5.2 to $G[C]$ and the vertex sets $N(v)$ and T to conclude that there are pairwise vertex-disjoint paths $\{p_i \mid i \in [r]\}$ each connecting $N(v)$ to a set $N(B_i, C)$, for some $B_i \in \mathcal{B}$. We let the (r, s)-cross have center v and branches $\{V(p_i) \cup B_i \mid i \in [r]\}$. Let us verify that this is indeed a valid (r, s)-cross.

Each path p_i connects $N(v)$ to $N(B_i, C)$ and we thus have that, as required, each branch intersects $N(v)$ and that the branches are connected, where we use that the sets $B_i \in \mathcal{B}$ are by definition connected. We also need to verify that the branches are pairwise vertex-disjoint. To this end recall that the sets $B_i \in \mathcal{B}$ are pairwise disjoint and, furthermore, each such set is disjoint from C. As the pairwise vertex-disjoint paths p_i live in $G[C]$, these paths do not intersect $\cup_{i \in [r]} B_i$ and we may thus conclude that the branches are pairwise vertex-disjoint. Finally, we also need to check that each branch is of size s: each set B_i is of size s and thus each branch is of size at least s. Shrinking the branches to the appropriate size recovers the statement. □

B.5.4 Odd and Even Paths

In this section we prove Lemma B.5.5.

Lemma B.5.5. *For all $\beta > 0$ the following holds. Let G be a β-expander on n vertices that contains an odd cycle of length $\ell \geq 1 + 2/\beta$. Then, for all $u \in V(G)$*

and $A \subseteq V(G)$, *of size* $|A| \geq (D^{\emptyset}_{\beta} \log n + 1)(1 + 2/\beta)$, *there is a vertex* $v \in A$ *such that* u *and* v *are connected by both an odd and an even path, each of length at most* $(15D^{\emptyset}_{\beta/2}/\beta) \log n + \ell$.

The lemma is a corollary of a more general statement about short paths in α-expanders. The lemma states that if sets S, T, where $|S| \gtrsim |T|/\alpha$, are connected by $|T|$ many short vertex-disjoint paths, then for any large set U there is again a set of short vertex-disjoint paths that does not only connect every vertex of T to S but also a vertex from U to S.

In order to state the lemma, let us introduce some notation. For a graph G and vertex sets $S, T \subseteq V(G)$, denote by $L^{G}_{\text{disj}}(T, S)$ the minimum total length of connecting all vertices of T to S by pairwise vertex-disjoint paths;

$$L^{G}_{\text{disj}}(T, S) = \min_{\{p_t | t \in T\}} \sum_{t \in T} |p_t| \tag{B.21}$$

where $\{p_t \mid t \in T\}$ ranges over all sets of pairwise vertex-disjoint paths such that p_t connects t to S (note the paths $\{p_t\}$ are not allowed to intersect even in S). If no such set of paths exists, the value of the minimum is taken to be ∞. If the graph G is clear from context, we omit the superscript.

A similar lemma (though without the essential upper bound on the path lengths) has appeared in e.g. [FK19].

Lemma B.5.6. *Let* G *be a* β*-expander on* n *vertices and* $S, T \subseteq V(G)$ *satisfy* $|S| \geq |T|(1 + 2/\beta)$. *Then every set* $U \subseteq V(G)$, *of size* $|U| \geq (L_{disj}(T, S) + |T|)(1 + 2/\beta)$, *contains a vertex* $u \in U$ *such that* $L_{disj}(T \cup \{u\}, S) \leq 7(L_{disj}(T, S) + |T|)/\beta + 2D^{\emptyset}_{\beta/2} \log n$.

Lemma B.5.5 follows by a single application of Lemma B.5.6.

Proof of Lemma B.5.5. Let C denote an odd cycle of length $\ell \geq 1 + 2/\beta$, as guaranteed to exist, and denote by p a shortest path connecting u to C. By Lemma B.2.3, we know that $|p| \leq D^{\emptyset}_{\beta} \log n$. Apply Lemma B.5.6 to $S = C$, $T = \{u\}$, $p_u = p$, and $U = A$. We conclude that there is a $v \in A$ and two vertex-disjoint paths p'_u, p'_v connecting u and v to C, of total length at most $(15D^{\emptyset}_{\beta/2}/\beta) \log n$. We can join these paths into a path between u and v by walking along C in either of the two directions. Since C has odd length this results in one odd and one even length path connecting u to v, each of length at most $\ell + (15D^{\emptyset}_{\beta/2}/\beta) \log n$, as required. □

Proof of Lemma B.5.6. Denote by $\mathcal{P} = \{p_t \mid t \in T\}$ a set of pairwise vertex-disjoint paths of smallest total length, where the path p_t connects t to S. Let $V(\mathcal{P}) = \cup_{p \in \mathcal{P}} V(p)$ denote all the vertices in the paths in $\mathcal{P}$. Clearly,

$|V(\mathcal{P})| = L_{\text{disj}}(T, S) + |T|$. Set $m = |V(\mathcal{P})|(1 + 2/\beta)$ and $r = \lceil \frac{\log m}{\log(1+\beta/2)} \rceil$. Note that $n \geq |U| \geq m$ and hence $r \leq \frac{1}{2} D^{\varnothing}_{\beta/2} \log n$.

If $|S| \geq m$, apply Corollary B.2.5 to $V(\mathcal{P})$, $S \setminus V(\mathcal{P})$ and $U \setminus V(\mathcal{P})$ to conclude that there is a path p of length $D^{\varnothing}_{\beta/2} \log n$ connecting $S \setminus V(\mathcal{P})$ to $U \setminus V(\mathcal{P})$ in $G \setminus V(\mathcal{P})$. The set $\mathcal{P} \cup \{p\}$ clearly satisfies the conclusion of the lemma.

Otherwise, if $|S| < m$, we want to get into a position where we can again apply Corollary B.2.5. To this end, we define a sequence of sets of vertices $S = S_0 \subseteq S_1 \subseteq \ldots \subseteq S_\ell \subseteq V(G)$ that are in some sense well-connected to S. We formalize this property after explaining how to obtain these sets.

The set S_{i+1} is defined in terms of S_i using the following process. Let w^i_t be the last vertex on the path p_t (viewed as a path from S to t) that is in S_i and $W_i = \{w^i_t \mid t \in T\}$. Suppose $|S_i| < m$ and there is a path of length at most r connecting $S_i \setminus W_i$ to $V(\mathcal{P}) \setminus S_i$ in the graph $G \setminus W_i$. Denote by q_i a minimal such path, denote by w the endpoint of q_i in $V(\mathcal{P}) \setminus S_i$, and let $t_i \in T$ be such that $w \in V(p_{t_i})$. Then, define $S_{i+1} = S_i \cup q_i \cup p_{t_i}[w^{t_i}_i, w]$. Otherwise, if $|S_i| \geq m$ or there is no such q_i, set $\ell = i$ and stop the process. There is an illustration of this process in Figure B.3.

The following claim formalizes the well-connectedness property of S_ℓ.

Claim B.5.7. *For every vertex $s^\star \in S_\ell \setminus W_\ell$ it holds that $L^{G[S_\ell \cup V(\mathcal{P})]}_{disj}(T \cup \{s^\star\}, S) \leq L^{G}_{disj}(T, S) + |S_\ell|$ and furthermore the paths achieving this bound are the same as the paths in $\mathcal{P}$ outside $S_\ell \setminus W_\ell$.*

Proof. Proof by induction on $i \in \{0, \ldots, \ell\}$. The base case $i = 0$ clearly holds – we have for all $s^\star \in S_0 \setminus W_0 \subseteq S$ that $L^{G[S_0 \cup V(\mathcal{P})]}_{\text{disj}}(T \cup \{s^\star\}, S) = L^{G}_{\text{disj}}(T, S)$.

Suppose the statement is true for some $i \in \{0, \ldots \ell - 1\}$ and let us prove that is then true for $i + 1$ as well. By the inductive hypothesis, $L^{G[S_i \cup V(\mathcal{P})]}_{\text{disj}}(T \cup \{s_i\}, S) \leq L^{G}_{\text{disj}}(T, S) + |S_i|$, and this bound can be achieved by a set of paths $\mathcal{P}'$ which follow $\mathcal{P}$ outside $S_i \setminus W_i$.

Fix an arbitrary $s^\star \in S_{i+1} \setminus W_{i+1}$. By the induction hypothesis the claim holds for $s^\star \in S_i \setminus W_i$, so we may assume [5] that either $s^\star \in q_i$, or $s^\star \in p_{t_i}[w^{t_i}_i, w^{t_i}_{i+1}]$. If $s^\star \in q_i$ (excluding its endpoint $w^{t_i}_{i+1}$) then we simply extend the path in $\mathcal{P}'$ ending in s_i with the subpath of q_i from $s^\star$ to s_i, increasing the total length of $\mathcal{P}'$ by at most $|q_i|$. On the other hand if $s^\star \in p_{t_i}[w^{t_i}_i, w^{t_i}_{i+1}]$ then we reroute the path from t_i in $\mathcal{P}'$ to s_i via q_i and then use the now unused part of p_{t_i} to connect $s^\star$ to S, again increasing the total length of $\mathcal{P}'$ by at most $|q_i|$. There is an illustration of the two cases in Figure B.3.

[5] Here we are using that $W_i = (W_{i-1} \setminus \{w^{t_i}_{i-1}\}) \cup \{w^{t_i}_i\}$.

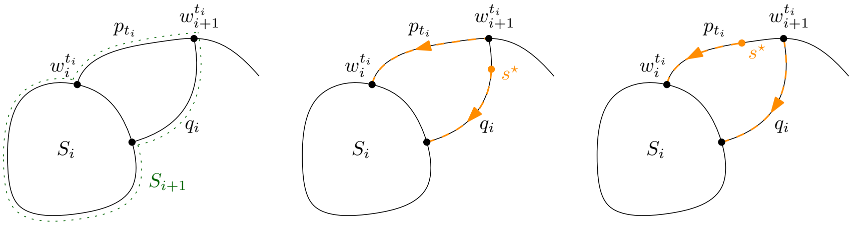


Figure B.3: Given the set S_i, the first figure depicts the process of obtaining the set S_{i+1}. The following figures indicate how to route the paths, as in the proof of Claim B.5.7, depending on where $s^\star$ is located.

In either case, we can connect T and $s^\star$ to S via vertex-disjoint paths of length at most

$$L_{\mathrm{disj}}^{G[S_{i+1}\cup V(\mathcal{P})]}(T\cup\{s^\star\},S)\le L_{\mathrm{disj}}^{G}(T,S)+|S_i|+|q_i|\le L_{\mathrm{disj}}^{G}(T,S)+|S_{i+1}|,$$

as desired. □

It is easy to see that $|S_\ell|\le 2m+r$: the number of vertices added by $p_{t_i}[w_i^{t_i},w_{i+1}^{t_i}]$ is always upper bounded by $|V(\mathcal{P})|\le m$. Suppose there is a path $p^\star$ of length $|p^\star|\le D^{\o}_{\beta/2}\log n+r$ connecting some vertex $s^\star\in S_\ell\setminus W_\ell$ to $u^\star\in U\setminus V(\mathcal{P}_\ell)$ in $G\setminus V(\mathcal{P}_\ell)$. We can then "compose" the paths to conclude that

$$\begin{aligned}
L_{\mathrm{disj}}^{G}(T\cup\{u^\star\},S)\le L_{\mathrm{disj}}^{G[S_\ell]}(W_\ell\cup\{s^\star\},S)+&\\
L_{\mathrm{disj}}^{G\setminus(S_\ell\setminus(W_\ell\cup\{s^\star\}))}(W_\ell\cup\{s^\star\},T\cup\{u^\star\})& \qquad\text{(B.22)}\\
\le|S_\ell|+L_{\mathrm{disj}}^{G}(T,S)+|p^\star|& \qquad\text{(B.23)}\\
\le|S_\ell|+L_{\mathrm{disj}}^{G}(T,S)+D^{\o}_{\beta/2}\log n+r& \qquad\text{(B.24)}\\
\le 2m+r+L_{\mathrm{disj}}^{G}(T,S)+D^{\o}_{\beta/2}\log n+r& \qquad\text{(B.25)}\\
\le 7(L_{\mathrm{disj}}^{G}(T,S)+|T|)/\beta+2D^{\o}_{\beta/2}\log n\ ,& \qquad\text{(B.26)}
\end{aligned}$$

as claimed in the statement.

It remains to establish that such a path $p^\star$ exists. If $|S_\ell|\ge m$, apply Corollary B.2.5 to $V(\mathcal{P})$, $S_\ell\setminus W_\ell$ and $U\setminus V(\mathcal{P})$ to conclude that there is a path $p^\star$ of length at most $|p^\star|\le D^{\o}_{\beta/2}\log n$ that connects $S_\ell\setminus W_\ell$ to $U\setminus V(\mathcal{P})$ in $G\setminus V(\mathcal{P})$.

Otherwise, by construction, $S_\ell\setminus W_\ell$ cannot reach $V(\mathcal{P})\setminus W_\ell$ within r steps in $G\setminus W_\ell$. Hence, to argue that in $G\setminus V(\mathcal{P})$ the ball of radius r

around $S_\ell \setminus W_\ell$ is large, we do not need to apply Lemma B.2.4 to $V(\mathcal{P})$ and $S_\ell \setminus W_\ell$ but in fact can apply it to W_ℓ and $S_\ell \setminus W_\ell$, where we use that $|S_\ell \setminus W_\ell| \geq |S| - |T| \geq 2|T|/\beta$. This enables us to grow $S_\ell \setminus W_\ell$ into a set $S^\star = B_r^{G \setminus W_\ell}(S_\ell \setminus W_\ell) = B_r^{G \setminus V(\mathcal{P})}(S_\ell \setminus W_\ell)$ of size at least m. Now we are in a position to apply Corollary B.2.5 to $V(\mathcal{P})$, $S^\star$ and $U \setminus V(\mathcal{P})$ to conclude that there is a path of length at most $D^{\varnothing}_{\beta/2} \log n$ that connects $S^\star$ to $U \setminus V(\mathcal{P})$ in $G \setminus V(\mathcal{P})$. Taking an additional r steps in $G[S^\star]$, one can reach $S_\ell \setminus W_\ell$, as required. This concludes the proof of the lemma. □

B.6 Concluding Remarks

We have established average-case lower bounds for refuting the perfect matching formula and more generally the $\mathrm{Card}(G, \vec{t})$ formula in random d-regular graphs on an odd number of vertices. Let us conclude by discussing some further loose ends and mention some open problems.

B.6.1 Polynomial Calculus Space Lower Bounds

The space of a PC refutation π is the amount of memory needed to verify π. The PC space of a formula $\mathcal{F}$ is then the minimum space required for any PC refutation π of $\mathcal{F}$. As this is rather tangential to the rest of the paper we refer to [FLM+13] for formal definitions. For convenience, let us restate our result on PC space.

Theorem B.1.3. *For all $\alpha > 0$ there is a d_0 such that the following holds. Let G be a bounded degree α-expander on n vertices of average degree at least d_0. Then over any field $\mathbb{F}$ it holds that $PC_{\mathbb{F}}$ requires space $\Omega(n/\log n)$ to refute the Tseitin formula defined on G.*

The proof idea is to take the worst-case Tseitin lower bounds from Filmus et al. [FLM+13] for which PC requires $\Omega(n)$ space and embed these into a vertex expander of large enough average degree. The only compication that arises is that these formulas are defined over multigraphs – the multigraph H is obtained from an appropriate[6] constant degree graph G by doubling each edge. An inspection of the proof of Theorem B.3.3 reveals that H may be a multigraph and we can thus implement our proof strategy.

Proof Sketch. Consider the worst-case instance H from Filmus et al. [FLM+13] on $\varepsilon n/\log n$ vertices, for some small enough $\varepsilon > 0$. Apply Theorem B.3.3 to H and G. This gives a topological embedding of H in G, with no control

[6] See the proof of Theorem 8 in [FLM+13].

of the parities of the length of the paths. Consider a restriction ρ that sets the variables outside the embedding of H such that no axiom is falsified (see, e.g., [PRST16]). By appropriately substituting the variables on each path of the topological embedding we obtain that the worst-case instance $\tau(H)$ is an affine restriction of $\tau(G)$. As an affine restriction only reduces the amount of space needed to verify a proof, we see that $\tau(G)$ requires PC space $\Omega(n/\log n)$. □

B.6.2 Paths in Expanders

The arguments used in the proof of Theorem B.3.3 can be adapted to make partial progress on a question by Friedman and Krivelevich [FK19]. They asked, given a positive integer q, whether it is possible to guarantee the existence of a cycle whose length is divisible by q in every α-expander.

We can show that for all primes q satisfying $1/\text{poly}(\alpha) \ll q \ll \sqrt{n/\log n}$, this indeed holds. In fact, for all $a \in \mathbb{Z}_q$, we can show that there is a cycle of length $a \bmod q$.

The idea is to embed a cycle C_{q^2} of length q^2 into G such that between any two vertices there are two paths whose length difference is non-zero modulo q. If we can ensure this, as all $0 \neq b \in \mathbb{Z}_q$ are generators, we can choose one path between all embedded vertices such that the length of the cycle is $a \bmod q$ for any $a \in \mathbb{Z}_q$.

In order to obtain paths of different length modulo q, let us embed a cycle c_e (of length $\gg 1/\text{poly}(\alpha)$) for each edge $e = \{u, v\}$. We then want to connect the vertex embeddings B_u, B_v to c_e such that the two resulting paths are of different length modulo q. Note that once a vertex is connected to the cycle, there are only about $2/q$ vertices in c_e such that both paths are of equal length modulo q. As q is rather large and thus there are few such "bad" vertices, when an edge embedding has to be moved to the sets A, A', we can ensure that the set A' remains relatively small compared to A.

B.6.3 Open Problems

The main concrete problem left open is to reduce the degree of the hard graphs: the embedding approach taken in the worst-case to average-case reduction results in very large degree d; while Theorem B.1.1 does not give an explicit estimate on d_0 one can trace through the proofs and get an estimate somewhere around 15 000. The main bottleneck that prevents us from reducing this is the Partition Lemma and in particular the dependence of d_0 on c and ϵ in Lemma B.4.3. If this part could be significantly improved or circumvented we believe that the degree of the graph could be significantly reduced, although it would still be relatively large (at least a few hundred). It would be interesting to see what happens for very small degrees such as a 4-regular graph (recall that since n is odd, d must be even) – is $\mathrm{PM}(G)$ hard with high probability even for these graphs?

Another interesting question is the proof complexity of perfect matching in Polynomial Calculus over $\mathbb{F}_2$ (or any other field of characteristic 2). While $\mathrm{PC}_{\mathbb{F}_2}$ can refute the perfect matching formula on an odd number of vertices for parity reasons, the situation is less clear when the number of vertices is even. Are there graphs G that do not admit perfect matchings but $\mathrm{PC}_{\mathbb{F}_2}$ requires exponential size refutations?

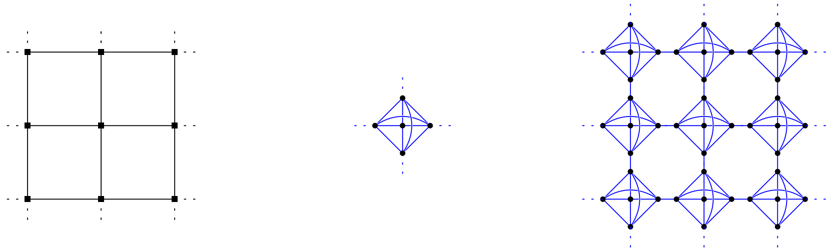
We establish Theorem B.1.1 for random graphs and not for all spectral expanders. Our proof mostly uses the expansion properties of random graphs, except for two places: in the Partition Lemma to argue that every subgraph contains a not too short odd cycle, and for the contiguity argument in Section B.3.2. However, we believe that it should be possible to circumvent these uses of randomness and establish the lower bounds for all spectral expanders.

Theorem B.1.1 only gives lower bounds for $\mathrm{Card}(G, b)$ when $b = \vec{t}$ is a constant vector (and G is regular). It would be nice to characterize more generally for which vectors b the formula is hard. In the analogous setting for Tseitin formulas, the precise charges of the vertices do not matter, as long as the sum of charges is odd the formula remains hard to refute on a random graph [BGIP01; Gri01]. In the $\mathrm{Card}(G, b)$ case however this is not the case. For instance, if the vector of target degrees b violates any of the *inequalities* of the Erdős-Gallai characterization of degree sequences then SoS (in fact even Sherali-Adams) can easily refute $\mathrm{Card}(G, b)$.[7]

In the case when G is the complete graph this in fact gives a complete characterization of the easy and hard vectors b but for sparse graphs the situation is less clear. Is there a nice characterization of vectors b for which $\mathrm{Card}(G, b)$ is hard for SoS with high probability over a random d-regular G?

More broadly, another open problem is to prove SoS lower bounds for random CSPs that do not support pairwise uniform distributions (cf. the brief discussion on CSPs in Section B.1.2). Viewed this way, our results establish hardness of random monotone 1-in-k-SAT instances with two occurrences per variable, for some large constant k. Reducing k corresponds to the aforementioned problem of reducing the degree, but some other natural questions are to look at other CSPs such as 1-in-k-SAT with negated

[7] To see this, note that for any $S \subseteq V(G)$, SoS of degree 1 can derive $\sum_{v \in S} b_v = \sum_{v \in S} \sum_{e \ni v} x_e \leq |E(S, S)| + \sum_{e \in E(S, \overline{S})} x_e \leq |S|(|S| - 1) + \sum_{v \notin S} \sum_{e \in E(S,v)} x_e$, and also that $\sum_{e \in E(S,v)} x_e \leq \min(b_v, |S|)$ for every $v \notin S$. Combining these, Sherali-Adams can derive $\sum_{v \in S} b_v \leq |S|(|S| - 1) + \sum_{v \notin S} \min(b_v, |S|)$ which in particular means it can derive all the inequalities of the Erdős-Gallai theorem.



(a) The graph G.

(b) The lift of a single vertex.

(c) The lifted graph H.

Figure B.4: An illustration of the blow-up construction, starting from a 4-regular graph.

literals, or to understand the hardness as a function of the density of the instances.

B.7 Worst-Case Lower Bounds

In this section we describe a general reduction from the Tseitin formula to the Perfect Matching formula as it appeared in [BGIP01] for Polynomial Calculus. We then observe that this reduction also works for the SoS and bounded depth Frege proof systems.

Starting from a graph G such that the Tseitin formula $\tau(G)$ is hard for a proof system P, we want to craft a graph H so that $\mathrm{PM}(H)$ is hard for P. To simplify the presentation, let us assume that G is d-regular. As we are interested in unsatisfiable instances, i.e., when G has an odd number of vertices, we may assume that d is even.

The graph H is a “blow-up” (or “lift”) of G: each vertex in $V(G)$ is lifted to a clique of $d + 1$ vertices and each lifted edge connects a single pair of vertices from the corresponding cliques. If we denote the lifted vertices of $v \in V(G)$ by $\mathrm{lift}(v) = \{(v, \star), (v, 1), \ldots, (v, d)\}$, we add for each edge $\{u, v\} \in E(G)$, where v is the ith neighbor of u and u is the jth neighbor of v, an edge $\{(u, i), (v, j)\}$. An illustration of the construction of H can be found in Figure B.4.

For intuition, let us describe how we would obtain a satisfying assignment to the Perfect Matching Formula from a hypothetical satisfying assignment to the Tseitin Formula. Set the lifted edges to the same value as they are set to in the Tseitin Formula. Now observe that each charge is odd and hence there is an even number of vertices left that are not matched yet in each $\mathrm{lift}(v)$, for $v \in V(G)$. As the vertices in $\mathrm{lift}(v)$ form a clique, we can select a perfect matching on these unmatched vertices to obtain a satisfying assignment to the Perfect Matching Formula.

Buss et al. [BGIP01] showed that Polynomial Calculus can simulate this reduction.

Theorem B.7.1 ([BGIP01])**.** *There are graphs* G *on an odd number of vertices* n *and maximum degree* $\Delta(G) = 5$ *such that Polynomial Calculus over any field of characteristic different from* 2 *requires degree* $\Theta(n)$ *to refute* $\mathrm{PM}(G)$*.*

What remains is to check that this reduction also gives Perfect Matching worst-case lower bounds for Sum-of-Squares and bounded depth Frege. This straightforward verification is carried out in the following two sections.

B.7.1 Sum-of-Squares

A nice property of the Sum-of-Squares system is that if the variables for a formula $\mathcal{Q}$ can be expressed as well-behaved low-degree polynomials in the variables of another formula $\mathcal{P}$ for which a pseudo-expectation exists, then a pseudo-expectation also exists for $\mathcal{Q}$. This property is well-known but let us state and quickly prove the exact version we need.

Claim B.7.2. *Let* $\mathcal{P} \subseteq \mathbb{R}[x_1, \ldots, x_n]$ *and* $\mathcal{Q} \subseteq \mathbb{R}[y_1, \ldots, y_m]$ *be two systems of polynomial equations. Let* $\widetilde{\mathbb{E}}_{\mathcal{Q}}$ *be a degree* D *pseudo-expectation for* $\mathcal{Q}$*. Suppose there is a function* $f : \{x_1, \ldots, x_m\} \to \mathbb{R}[y_1, \ldots y_n]$*, mapping the* x *variables to polynomials in* y *of degree at most* t*. Extend* f *to polynomials by applying the function to each variable individually. If* f *satisfies that* $\widetilde{\mathbb{E}}_{\mathcal{Q}}[f(r \cdot p)] = 0$*, for all* $p \in \mathcal{P}$ *and* $r \in \mathbb{R}[x_1, \ldots x_m]$ *of degree* $\deg(r \cdot p) \leq D/t$*, then* $\widetilde{\mathbb{E}}_{\mathcal{Q}} \circ f$ *is a degree* D/t *pseudo-expectation for* $\mathcal{P}$*.*

Proof. As f only maps variables we have that $\widetilde{\mathbb{E}}_{\mathcal{Q}}[f(1)] = \widetilde{\mathbb{E}}_{\mathcal{Q}}[1] = 1$. Also, we need to check that $\widetilde{\mathbb{E}}_{\mathcal{Q}}[f(s^2)] \geq 0$ for $s \in \mathbb{R}[x_1, \ldots, x_m]$ of degree $\deg(s) \leq D/2t$. As we apply f individually to each variable, we can write $\widetilde{\mathbb{E}}_{\mathcal{Q}}[f(s^2)] = \widetilde{\mathbb{E}}_{\mathcal{Q}}[\big(\sum_{t \in s} f(t)\big)^2] \geq 0$, as $\widetilde{\mathbb{E}}_{\mathcal{Q}}$ is a degree D pseudo-expectation. □

In order to apply Claim B.7.2 with $\mathcal{Q} = \tau(G)$ and $\mathcal{P} = \mathrm{PM}(H)$, we need to express each variable from the Perfect Matching formula as a low degree polynomial in the Tseitin variables.

Let us recall some notation. For a vertex $v \in V(G)$, let Y_v be the set of Tseitin variables corresponding to edges incident to v and denote by A_v all boolean assignments to Y_v that satisfy the vertex axiom of v, i.e., assignments that set an odd number of edges to true. For a Tseitin variable y_e, where $e \in E(G)$, let $\mathrm{lift}(y_e) \in E(H)$ denote the lifted edge variable.

With this notation at hand, let us define the function f to use in Claim B.7.2. Variables that correspond to lifted edges, $x_e = \mathrm{lift}(y_{e'})$ for some $e' \in E(G)$, are set to 1 if and only if $y_{e'}$ is set to 1 and the variables in Y_v are set according to some assignment in A_v:

$$f(x_e) = \sum_{\substack{\alpha \in A_v \\ \alpha(y_{e'})=1}} 1\{Y_v = \alpha\} \ . \tag{B.27}$$

Note that this is a polynomial of degree $\deg(v) = d$ in the y_e's. For each assignment $\alpha \in A_v$, set the variables in $\text{lift}(Y_v)$ according to α and fix a matching m_α on the vertices in $\text{lift}(v)$ not matched by α. For any edge $e \subseteq \text{lift}(v)$, let

$$f(x_e) = \sum_{\substack{\alpha \in A_v \\ e \in m_\alpha}} 1\{Y_v = \alpha\} \ . \tag{B.28}$$

If we apply f individually to each variable, we claim that for $i \in \{1, \ldots, d\}$ and $v \in V(G)$ the polynomial $f(q^{\text{PM}}_{(v,i)})$ is equal to the Tseitin axiom q^τ_v:

$$\begin{aligned} f(q^{\text{PM}}_{(v,i)}) &= \sum_{e \ni (v,i)} f(x_e) - 1 && \text{(B.29)} \\ &= \sum_{\substack{\alpha \in A_v \\ \alpha(y_{e'})=1}} 1\{Y_v = \alpha\} + \sum_{\substack{\alpha \in A_v \\ \alpha(y_{e'})=0}} 1\{Y_v = \alpha\} - 1 && \text{(B.30)} \\ &= q^\tau_v \ , && \text{(B.31)} \end{aligned}$$

using that the m_α are matchings. The axioms $q^{\text{PM}}_{(v,\star)}$ are handled similarly:

$$f(q^{\text{PM}}_{(v,\star)}) = \sum_{e \ni (v,\star)} f(x_e) - 1 = \sum_{\alpha \in A_v} 1\{Y_v = \alpha\} - 1 = q^\tau_v \ . \tag{B.32}$$

As $\widetilde{\mathbb{E}}_{\tau(G)}$ maps all axioms multiplied by a low degree polynomial to 0, the same holds for $\widetilde{\mathbb{E}}_{\tau(G)} \circ f$ and we can thus apply Claim B.7.2.

We conclude that if there is a degree D pseudo-expectation $\widetilde{\mathbb{E}}_{\tau(G)}$ for the Tseitin Formula $\tau(G)$, then there is a degree D/d pseudo-expectation $\widetilde{\mathbb{E}}_{\text{PM}(H)}$ for the Perfect Matching formula over the lifted graph H. Using Grigoriev's Tseitin lower bound [Gri01] we obtain the following Theorem.

Theorem B.7.3. *There are graphs G on an odd number of vertices n and maximum degree $\Delta(G) = 5$ for which SoS requires degree $\Theta(n)$ to refute* $\text{PM}(G)$.

B.7.2 Bounded Depth Frege

In this section we intend to prove the following theorem.

Theorem B.7.4. *There is a constant $c > 0$ such that the following holds. Suppose $D \le \frac{c \log n}{\log \log n}$. Then there are graphs G on an odd number of vertices n and maximum degree $\Delta(G) = 5$ such that any depth-D Frege refutation of* $\mathrm{PM}(G)$ *requires size* $\exp(\Omega(n^{c/D}))$.

As in the previous section we use a function f, mapping Perfect Matching variables to low depth formulas in the Tseitin Variables, to argue that we can transform a refutation of $\mathrm{PM}(H)$ into a refutation of the Tseitin formula $\tau(G)$. Assuming that this can be done, we use the following recent result of Håstad about the Tseitin formula over the grid to obtain Theorem B.7.4.

Theorem B.7.5 ([Hås20]). *Suppose that $D \le \frac{\log n}{59 \log \log n}$, then any depth-$D$ Frege refutation of the Tseitin formula on the $n \times n$ grid requires size* $\exp(\Omega(n^{1/58(D+1)}))$.

In the previous section f mapped to polynomials. As we are now working with formulas we need to translate the polynomials to formulas. This is straightforward; reusing notation from the previous section, let

$$f(x_e) = \bigvee_{\substack{\alpha \in A_v \\ \alpha(y_{e'})=1}} 1\{Y_v = \alpha\} \ , \tag{B.33}$$

if $x_e = \mathrm{lift}(y_{e'})$ is a lifted edge. Else let

$$f(x_e) = \bigvee_{\substack{\alpha \in A_v \\ e \in m_\alpha}} 1\{Y_v = \alpha\} \ . \tag{B.34}$$

Suppose there is a depth-D Frege refutation π of the Perfect Matching formula $\mathrm{PM}(H)$. Replace each occurrence of a Perfect Matching variable x_e by $f(x_e)$ to obtain a depth-$(D+2)$ refutation π'. We claim that π' can be massaged into a refutation of the Tseitin formula $\tau(G)$ of size $O_d(\mathrm{Size}(\pi))$.

To this end we need to argue that f maps Perfect Matching axioms to Tseitin Axioms or tautologies that are derivable in small size and depth. Analoguous to SoS observe that for all $v \in V(G)$ and $i \in \{1, \ldots, d\}$,

$$f\Big(\bigvee_{e \ni (v,i)} x_e\Big) = \bigvee_{e \ni (v,i)} f(x_e) \tag{B.35}$$

$$= \bigvee_{\substack{\alpha \in A_v \\ \alpha(y_{e'})=1}} 1\{Y_v = \alpha\} \vee \bigvee_{\substack{\alpha \in A_v \\ \alpha(y_{e'})=0}} 1\{Y_v = \alpha\} \ . \tag{B.36}$$

Note that that the final formula is equal to the axiom q_v^τ, up to a reordering of the terms. As the axioms are over d variables, which is constant in our case, this formula can be derived from q_v^τ in constant depth and size. The axiom for the vertex $(v, \star)$ is handled in a similar manner.

Last we need to show that the axioms $\bar{x}_e \vee \bar{x}_{e'}$, for edges $e \neq e' \in E(H)$ satisfying $e \cap e' \neq \emptyset$, are mapped to a tautology derivable in small size and depth. If we let $\{(v, i)\} = e \cap e'$ we can write

$$f(\bar{x}_e \vee \bar{x}_{e'}) = \Big(\neg \bigvee_{\beta \in B} 1\{Y_v = \beta\}\Big) \vee \Big(\neg \bigvee_{\gamma \in C} 1\{Y_v = \gamma\}\Big) \ , \tag{B.37}$$

for disjoint subsets $B, C \subseteq A_v$. Observe that this formula is a tautology and defined on d variables. Thus it is derivable in constant depth and size dependent on d, which is constant in our case.

B.8 Embedding Algorithm

Algorithm 1 Restores β-expansion of $G[C]$.

```
procedure FixExpansion($G, C, A, \beta$)
    while $G[C]$ is not a $\beta$-expander do
        $U \leftarrow_{\text{any}}$ subset of $C$ s.t. $|U| \leq |C|/2$ and $|N(U, C \setminus U)| < \beta|U|$
        $C \leftarrow C \setminus U$
        $A \leftarrow A \cup U$
```

Algorithm 2 Finds an (r, s)-cross in an β-expander G as in the proof of Lemma B.5.4.

```
Require: Conditions of Lemma B.5.4.
procedure EmbedVertex($G, r, s, \beta, k$)
    $\gamma \leftarrow \frac{\beta}{3(1+\beta)}$
    $s \leftarrow \max\{1/\gamma, s\}$
    $r' \leftarrow (1 + 1/\gamma)r$
    $A, \mathcal{B} \leftarrow \emptyset; C \leftarrow V(G)$
    while $|\mathcal{B}| < r'$ do
        $U \leftarrow_{\text{any}}$ subset of $C$ s.t. $|U| = s$ and $G[U]$ is connected
        $\mathcal{B} \leftarrow \mathcal{B} \cup \{U\}; C \leftarrow C \setminus U$
        FixExpansion($G, C, A, \gamma$)
        $\mathcal{F} \subseteq \mathcal{B}$ maximal such that $|\cup_{F \in \mathcal{F}} N(F, C)| < \gamma s|\mathcal{F}|$
        $\mathcal{B} \leftarrow \mathcal{B} \setminus \mathcal{F}; A \leftarrow A \cup_{F \in \mathcal{F}} F$

    $v \leftarrow_{\text{any}} C$ such that $\deg_{G[C]}(v) \geq r'$
    $F \leftarrow$ a transversal of $\{N(B, C) \mid B \in \mathcal{B}\}$
    $\{p_i \mid i \in [r]\} \leftarrow$ from Lemma B.5.2 applied to $G[C]$, $v$ and $F$
    return $\{v\} \cup \{V(p_i) \cup B_i \mid i \in [r]\}$ ▷ Shrink branches appropriately
```

Algorithm 3 Remove the embedding of vertex x.

```
procedure UnEmbedVertex($A, A', B, H, I, x$)
    $(v, \mathcal{U}) \leftarrow B_x$            ▷ $v$ is the center and $\mathcal{U}$ are the branches of $B_x$
    $B \leftarrow B \setminus B_x$; $I \leftarrow I \setminus x$
    $W \leftarrow \emptyset$
    for all $e \in E(H)$ such that $x \in e$ and $e$ is embedded do
        let $U \leftarrow \mathcal{U}$ be the branch adjacent to $B_e$
        $\mathcal{U} \leftarrow \mathcal{U} \setminus U$
        $B \leftarrow B \setminus B_e$
        $W \leftarrow W \cup U \cup B_e$
    $A \leftarrow A \cup \mathcal{U}$    ▷ First add to $A$, then to $A'$ to maintain the invariant
    $A' \leftarrow A' \cup W \cup \{v\}$
```

Algorithm 4 Embeds H in an α-expander G as in the proof of Theorem B.3.3.

```
procedure EmbedGraph($H, G, \alpha$)
    $\beta \leftarrow \alpha/3(1+\alpha)$
    $A, A', B \leftarrow \emptyset$; $C \leftarrow V(G)$
    $I \leftarrow \emptyset$
    while $I \neq V(H)$ do
        $x \leftarrow_{\text{any}} V(H) \setminus I$
        $B_x \leftarrow$ EmbedVertex($G[C], \deg_H(x), s, \beta, k$)
        $C \leftarrow C \setminus B_x$; $B \leftarrow B \cup B_x$; $I \leftarrow I \cup x$
        FixExpansion($G, C, A, \beta$)

        $\mathcal{U}_{\text{free}}(K) \leftarrow$ branches of cross $K$ not used for edge embeddings
        for all $\{x, y\} \in E(H)$ such that $y \in I$ do
            try
                $U_z \leftarrow_{\text{any}} \mathcal{U}_{\text{free}}(B_z)$ s.t. $|N(U_z, C)| \geq \beta |U_z|$ for $z \in \{x, y\}$
            catch no such $U_z$ for $z \in \{x, y\}$
                UnEmbedVertex($A, A', B, H, I, z$); continue
            $B_{xy} \leftarrow$ odd path from Lemma B.5.5
            $C \leftarrow C \setminus B_{xy}$; $B \leftarrow B \cup B_{xy}$
            FixExpansion($G, C, A, \beta$)
    return $B$
```

References

[AGK20] J. Abascal, V. Guruswami and P. K. Kothari, ”Strongly refuting all semi-random boolean csps”, *CoRR*, vol. abs/2009.08032, 2020. arXiv: 2009.08032. [Online]. Available: https://arxiv.org/abs/2009.08032 (cit. on p. 112)

[AR01] M. Alekhnovich and A. A. Razborov, ”Lower bounds for polynomial calculus: Non-binomial case”, in *Proceedings 42nd IEEE Symposium on Foundations of Computer Science*, 2001, pp. 190–199 (cit. on pp. 112, 113, 117)

[ABRW04] M. Alekhnovich, E. Ben-Sasson, A. A. Razborov and A. Wigderson, ”Pseudorandom generators in propositional proof complexity”, *SIAM Journal on Computing*, vol. 34, no. 1, pp. 67–88, 2004. DOI: 10.1137/S0097539701389944 (cit. on p. 112)

[AS00] N. Alon and J. H. Spencer, *The Probabilistic Method*, 2nd. Wiley Publishing, 2000, ISBN: 1119061954 (cit. on p. 128)

[AH19] A. Atserias and T. Hakoniemi, ”Size-Degree Trade-Offs for Sums-of-Squares and Positivstellensatz Proofs”, in *34th Computational Complexity Conference (CCC 2019)*, A. Shpilka, Ed., ser. Leibniz International Proceedings in Informatics (LIPIcs), vol. 137, Dagstuhl, Germany: Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 2019, 24:1–24:20, ISBN: 978-3-95977-116-0. DOI: 10.4230/LIPIcs.CCC.2019.24. [Online]. Available: http://drops.dagstuhl.de/opus/volltexte/2019/10846 (cit. on pp. 112, 115)

[BHK+16] B. Barak, S. B. Hopkins, J. Kelner, P. Kothari, A. Moitra and A. Potechin, ”A nearly tight sum-of-squares lower bound for the planted clique problem”, in *2016 IEEE 57th Annual Symposium on Foundations of Computer Science (FOCS)*, 2016, pp. 428–437 (cit. on p. 112)

[BS14] B. Barak and D. Steurer, ”Sum-of-squares proofs and the quest toward optimal algorithms”, in *Proceedings of the International Congress of Mathematicians (ICM)*, vol. IV, Aug. 2014, pp. 509–533. [Online]. Available: http://www.icm2014.org/download/Proceedings_Volume_IV.pdf (cit. on p. 112)

[Ber57] C. Berge, ”Two theorems in graph theory”, *Proceedings of the National Academy of Sciences of the United States of America*, vol. 43, no. 9, pp. 842–844, 1957, ISSN: 00278424. [Online]. Available: http://www.jstor.org/stable/89875 (visited on 20/07/2022) (cit. on p. 114)

[Ber18] C. Berkholz, ”The relation between polynomial calculus, Sherali-Adams, and sum-of-squares proofs”, in *Proceedings of the 35th Symposium on Theoretical Aspects of Computer Science (STACS '18)*, ser. Leibniz International Proceedings in Informatics (LIPIcs), vol. 96, Feb. 2018, 11:1–11:14 (cit. on p. 112)

[Bla37] A. Blake, ”Canonical expressions in Boolean algebra”, Ph.D. dissertation, University of Chicago, 1937 (cit. on p. 112)

[Bol01] B. Bollobás, *Random Graphs*, 2nd ed., ser. Cambridge Studies in Advanced Mathematics. Cambridge University Press, 2001. DOI: 10.1017/CBO9780511814068 (cit. on p. 137)

[BBH+17] G. Braun, J. Brown-Cohen, A. Huq, S. Pokutta, P. Raghavendra, A. Roy, B. Weitz and D. Zink, ”The matching problem has no small symmetric SDP”, en, *Math. Program.*, vol. 165, no. 2, pp. 643–662, Oct. 2017 (cit. on p. 119)

[BFSU96] A. Z. Broder, A. M. Frieze, S. Suen and E. Upfal, ”An efficient algorithm for the vertex-disjoint paths problem in random graphs”, in *Proceedings of the Seventh Annual ACM-SIAM Symposium on Discrete Algorithms*, ser. SODA '96, Atlanta, Georgia, USA: Society for Industrial and Applied Mathematics, 1996, pp. 261–268, ISBN: 0898713668 (cit. on p. 118)

[BH05] A. E. Brouwer and W. H. Haemers, ”Eigenvalues and perfect matchings”, *Linear Algebra and its Applications*, vol. 395, pp. 155–162, 2005, ISSN: 0024-3795. DOI: https://doi.org/10.1016/j.laa.2004.08.014. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0024379504003453 (cit. on p. 136)

[BN20] S. Buss and J. Nordström, ”Proof complexity and SAT solving”, Chapter to appear in the 2nd edition of *Handbook of Satisfiability.*, 2020 (cit. on p. 115)

[BGIP01] S. R. Buss, D. Grigoriev, R. Impagliazzo and T. Pitassi, ”Linear gaps between degrees for the polynomial calculus modulo distinct primes”, *Journal of Computer and System Sciences*, vol. 62, no. 2, pp. 267–289, Mar. 2001, Preliminary version in *CCC '99* (cit. on pp. 112–114, 119, 121, 154–156)

[CLRS16] S. O. Chan, J. R. Lee, P. Raghavendra and D. Steurer, ”Approximate constraint satisfaction requires large lp relaxations”, *J. ACM*, vol. 63, no. 4, Oct. 2016, ISSN: 0004-5411. DOI: 10.1145/2811255. [Online]. Available: https://doi.org/10.1145/2811255 (cit. on p. 119)

[CN19] J. Chuzhoy and R. Nimavat, *Large minors in expanders*, 2019. arXiv: 1901.09349 [cs.DS] (cit. on p. 116)

[CEI96] M. Clegg, J. Edmonds and R. Impagliazzo, ”Using the Groebner basis algorithm to find proofs of unsatisfiability”, in *Proceedings of the 28th Annual ACM Symposium on Theory of Computing (STOC ’96)*, May 1996, pp. 174–183 (cit. on pp. 112, 115)

[CR79] S. A. Cook and R. A. Reckhow, ”The relative efficiency of propositional proof systems”, *The Journal of Symbolic Logic*, vol. 44, no. 1, pp. 36–50, 1979, ISSN: 00224812. [Online]. Available: http://www.jstor.org/stable/2273702 (cit. on pp. 112, 123)

[DMO+19] Y. Deshpande, A. Montanari, R. O’Donnell, T. Schramm and S. Sen, ”The threshold for SDP-refutation of random regular NAE-3SAT”, in *Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms*, ser. SODA ’19, San Diego, California: Society for Industrial and Applied Mathematics, 2019, pp. 2305–2321 (cit. on p. 117)

[DKN20] N. Draganić, M. Krivelevich and R. Nenadov, *Rolling backwards can move you forward: On embedding problems in sparse expanders*, 2020. arXiv: 2007.08332 [math.CO] (cit. on p. 118)

[Edm65] J. Edmonds, ”Paths, trees, and flowers”, *Canadian Journal of Mathematics*, vol. 17, pp. 449–467, 1965. DOI: 10.4153/CJM-1965-045-4 (cit. on p. 114)

[FLM+13] Y. Filmus, M. Lauria, M. Mikša, J. Nordström and M. Vinyals, ”Towards an understanding of polynomial calculus: New separations and lower bounds (Extended abstract)”, in *Proceedings of the 40th International Colloquium on Automata, Languages and Programming (ICALP ’13)*, ser. Lecture Notes in Computer Science, vol. 7965, Springer, Jul. 2013, pp. 437–448 (cit. on pp. 116, 152)

[Fri08] J. Friedman, *A Proof of Alon’s Second Eigenvalue Conjecture and Related Problems*. Providence, R.I.: American Mathematical Society, 2008, ISBN: 9780821842805 0821842803 (cit. on p. 135)

[FK19] L. Friedman and M. Krivelevich, *Cycle lengths in expanding graphs*, 2019. arXiv: 1912.11011 [math.CO] (cit. on pp. 140, 149, 153)

[GIRS19] N. Galesi, D. Itsykson, A. Riazanov and A. Sofronova, "Bounded-depth frege complexity of tseitin formulas for all graphs", in *44th International Symposium on Mathematical Foundations of Computer Science (MFCS 2019)*, 2019, 49:1–15, ISBN: 978-3-95977-117-7. DOI: 10.4230/LIPIcs.MFCS.2019.49. [Online]. Available: http://drops.dagstuhl.de/opus/volltexte/2019/10993 (cit. on p. 117)

[GKT19] N. Galesi, L. Kołodziejczyk and N. Thapen, "Polynomial calculus space and resolution width", in *Proceedings of the 60th Annual IEEE Symposium on Foundations of Computer Science (FOCS '19)*, Nov. 2019, pp. 1325–1337 (cit. on p. 116)

[GL10] N. Galesi and M. Lauria, "Optimality of size-degree trade-offs for polynomial calculus", *ACM Transactions on Computational Logic*, vol. 12, no. 1, 4:1–4:22, Nov. 2010 (cit. on p. 112)

[GGR+09] J. Geelen, B. Gerards, B. Reed, P. Seymour and A. Vetta, "On the odd-minor variant of Hadwiger's conjecture", *Journal of Combinatorial Theory, Series B*, vol. 99, no. 1, pp. 20–29, 2009, ISSN: 0095-8956. DOI: https://doi.org/10.1016/j.jctb.2008.03.006. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0095895608000488 (cit. on p. 118)

[GI19] L. Glinskih and D. Itsykson, "On Tseitin Formulas, Read-Once Branching Programs and Treewidth", in *Computer Science – Theory and Applications*, R. van Bevern and G. Kucherov, Eds., Cham: Springer International Publishing, 2019, pp. 143–155, ISBN: 978-3-030-19955-5 (cit. on p. 117)

[Gri01] D. Grigoriev, "Linear lower bound on degrees of positivstellensatz calculus proofs for the parity", *Theoretical Computer Science*, vol. 259, no. 1, pp. 613–622, 2001, ISSN: 0304-3975. DOI: https://doi.org/10.1016/S0304-3975(00)00157-2. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0304397500001572 (cit. on pp. 112–114, 119, 154, 157)

[GHP02] D. Grigoriev, E. A. Hirsch and D. V. Pasechnik, "Complexity of semi-algebraic proofs", in *STACS 2002*, Berlin, Heidelberg: Springer Berlin Heidelberg, 2002, pp. 419–430 (cit. on pp. 112, 113)

[Hås20] J. Håstad, ”On small-depth Frege proofs for Tseitin for grids”, *J. ACM*, vol. 68, no. 1, Nov. 2020, ISSN: 0004-5411. DOI: 10.1145/3425606. [Online]. Available: https://doi.org/10.1145/3425606 (cit. on pp. 119, 158)

[HLW06] S. Hoory, N. Linial and A. Wigderson, ”Expander graphs and their applications”, *BULL. AMER. MATH. SOC.*, vol. 43, no. 4, pp. 439–561, 2006 (cit. on p. 136)

[IPS99] R. Impagliazzo, P. Pudlák and J. Sgall, ”Lower bounds for the polynomial calculus and the Gröbner basis algorithm”, *Computational Complexity*, vol. 8, no. 2, pp. 127–144, 1999 (cit. on pp. 112, 115)

[IRSS19] D. Itsykson, A. Riazanov, D. Sagunov and P. Smirnov, ”Almost tight lower bounds on regular resolution refutations of tseitin formulas for all constant-degree graphs”, *Electron. Colloquium Comput. Complex.*, vol. 26, p. 178, 2019. [Online]. Available: https://eccc.weizmann.ac.il/report/2019/178 (cit. on p. 117)

[JŁR00] S. Janson, T. Łuczak and A. Ruciński, *Theory of random graphs*. New York; Chichester: John Wiley & Sons, 2000 (cit. on p. 132)

[KR96] J. Kleinberg and R. Rubinfeld, ”Short paths in expander graphs”, in *Proceedings of the 37th Annual Symposium on Foundations of Computer Science*, ser. FOCS ’96, USA: IEEE Computer Society, 1996, p. 86 (cit. on pp. 116–118)

[KMR17] P. K. Kothari, R. Meka and P. Raghavendra, ”Approximating rectangles by juntas and weakly-exponential lower bounds for LP relaxations of CSPs”, ser. STOC 2017, Montreal, Canada: Association for Computing Machinery, 2017, pp. 590–603, ISBN: 9781450345286. DOI: 10.1145/3055399.3055438. [Online]. Available: https://doi.org/10.1145/3055399.3055438 (cit. on p. 119)

[KMOW17] P. K. Kothari, R. Mori, R. O’Donnell and D. Witmer, ”Sum of squares lower bounds for refuting any csp”, in *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing*, ser. STOC 2017, Montreal, Canada: Association for Computing Machinery, 2017, pp. 132–145, ISBN: 9781450345286. DOI: 10.1145/3055399.3055485. [Online]. Available: https://doi.org/10.1145/3055399.3055485 (cit. on pp. 112, 117)

[Kra19] J. Krajíček, *Proof Complexity*, ser. Encyclopedia of Mathematics and Its Applications. Cambridge University Press, Mar. 2019, vol. 170 (cit. on p. 112)

[Kri19] M. Krivelevich, "Expanders – how to find them, and what to find in them", in *Surveys in Combinatorics 2019*, ser. London Mathematical Society Lecture Note Series. Cambridge University Press, 2019, pp. 115–142. DOI: 10.1017/9781108649094.005 (cit. on pp. 116, 120, 127, 142)

[KN19] M. Krivelevich and R. Nenadov, "Complete Minors in Graphs Without Sparse Cuts", *International Mathematics Research Notices*, May 2019, rnz086, ISSN: 1073-7928. DOI: 10.1093/imrn/rnz086. eprint: https://academic.oup.com/imrn/article-pdf/doi/10.1093/imrn/rnz086/28672004/rnz086.pdf. [Online]. Available: https://doi.org/10.1093/imrn/rnz086 (cit. on pp. 116, 118, 120, 141, 142, 146)

[Las01] J. B. Lasserre, "An explicit exact sdp relaxation for nonlinear 0-1 programs", in *International Conference on Integer Programming and Combinatorial Optimization*, Springer, 2001, pp. 293–303 (cit. on p. 112)

[LRS15] J. R. Lee, P. Raghavendra and D. Steurer, "Lower bounds on the size of semidefinite programming relaxations", in *Proceedings of the Forty-Seventh Annual ACM on Symposium on Theory of Computing, STOC 2015, Portland, OR, USA, June 14-17, 2015*, R. A. Servedio and R. Rubinfeld, Eds., ACM, 2015, pp. 567–576. DOI: 10.1145/2746539.2746599. [Online]. Available: https://doi.org/10.1145/2746539.2746599 (cit. on p. 119)

[LS91] L. Lovász and A. Schrijver, "Cones of matrices and set-functions and 0-1 optimization", *SIAM Journal on Optimization*, vol. 1, no. 2, pp. 166–190, 1991 (cit. on p. 114)

[Mar06] K. Markström, "Locality and hard SAT-instances", *Journal on Satisfiability, Boolean Modeling and Computation*, vol. 2, no. 1-4, pp. 221–227, 2006 (cit. on p. 115)

[MPW15] R. Meka, A. Potechin and A. Wigderson, "Sum-of-squares lower bounds for planted clique", in *Proceedings of the 47th Annual ACM Symposium on Theory of Computing (STOC '15)*, Jun. 2015, pp. 87–96 (cit. on p. 112)

[MN15] M. Mikša and J. 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, pp. 467–487 (cit. on pp. 112, 113)

[MU05] M. Mitzenmacher and E. Upfal, *Probability and Computing: Randomized Algorithms and Probabilistic Analysis*. USA: Cambridge University Press, 2005, ISBN: 0521835402 (cit. on p. 128)

[Moh89] B. Mohar, "Isoperimetric numbers of graphs", *Journal of Combinatorial Theory, Series B*, vol. 47, no. 3, pp. 274–291, 1989, ISSN: 0095-8956. DOI: https://doi.org/10.1016/0095-8956(89)90029-4. [Online]. Available: http://www.sciencedirect.com/science/article/pii/0095895689900294 (cit. on p. 135)

[Par00] P. A. Parrilo, "Structured semidefinite programs and semialgebraic geometry methods in robustness and optimization", Ph.D. dissertation, California Institute of Technology, 2000 (cit. on p. 112)

[PRST16] T. Pitassi, B. Rossman, R. A. Servedio and L.-Y. Tan, "Polylogarithmic Frege depth lower bounds via an expander switching lemma", in *Proceedings of the Forty-Eighth Annual ACM Symposium on Theory of Computing*, ser. STOC '16, Cambridge, MA, USA: Association for Computing Machinery, 2016, pp. 644–657, ISBN: 9781450341325. DOI: 10.1145/2897518.2897637. [Online]. Available: https://doi.org/10.1145/2897518.2897637 (cit. on pp. 115, 117, 153)

[Pot17] A. Potechin, "Sum of squares lower bounds from symmetry and a good story", *CoRR*, vol. abs/1711.11469, 2017. arXiv: 1711.11469. [Online]. Available: http://arxiv.org/abs/1711.11469 (cit. on p. 114)

[Pot20] ——, "Sum of squares bounds for the ordering principle", in *35th Computational Complexity Conference, CCC 2020, July 28-31, 2020, Saarbrücken, Germany (Virtual Conference)*, S. Saraf, Ed., ser. LIPIcs, vol. 169, Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2020, 38:1–38:37. DOI: 10.4230/LIPIcs.CCC.2020.38. [Online]. Available: https://doi.org/10.4230/LIPIcs.CCC.2020.38 (cit. on p. 112)

[Raz98] A. A. Razborov, ”Lower bounds for the polynomial calculus”, *Computational Complexity*, vol. 7, no. 4, pp. 291–324, Dec. 1998 (cit. on pp. 112, 113)

[Raz02] ——, ”Proof complexity of pigeonhole principles”, in *5th International Conference on Developments in Language Theory, (DLT ’01), Revised Papers*, ser. Lecture Notes in Computer Science, vol. 2295, Springer, Jul. 2002, pp. 100–116 (cit. on p. 113)

[Raz17] ——, ”On the width of semialgebraic proofs and algorithms”, *Math. Oper. Res.*, vol. 42, no. 4, pp. 1106–1134, Nov. 2017, ISSN: 0364-765X. DOI: 10.1287/moor.2016.0840. [Online]. Available: https://doi.org/10.1287/moor.2016.0840 (cit. on p. 114)

[Rii93] S. Riis, ”Independence in bounded arithmetic”, Ph.D. dissertation, University of Oxford, 1993 (cit. on p. 113)

[RS86] N. Robertson and P. Seymour, ”Graph minors. v. excluding a planar graph”, *Journal of Combinatorial Theory, Series B*, vol. 41, no. 1, pp. 92–114, 1986, ISSN: 0095-8956. DOI: https://doi.org/10.1016/0095-8956(86)90030-4. [Online]. Available: https://www.sciencedirect.com/science/article/pii/0095895686900304 (cit. on p. 117)

[Rot17] T. Rothvoss, ”The matching polytope has exponential extension complexity”, *J. ACM*, vol. 64, no. 6, Sep. 2017, ISSN: 0004-5411. DOI: 10.1145/3127497. [Online]. Available: https://doi.org/10.1145/3127497 (cit. on p. 119)

[Sch08] G. Schoenebeck, ”Linear level Lasserre lower bounds for certain k-CSPs”, in *Proceedings of the 49th Annual IEEE Symposium on Foundations of Computer Science (FOCS ’08)*, Oct. 2008, pp. 593–602 (cit. on p. 112)

[Sey16] P. Seymour, ”Hadwiger’s conjecture”, in *Open Problems in Mathematics*, J. F. Nash Jr. and M. T. Rassias, Eds. Cham: Springer International Publishing, 2016, pp. 417–437, ISBN: 978-3-319-32162-2. DOI: 10.1007/978-3-319-32162-2_13. [Online]. Available: https://doi.org/10.1007/978-3-319-32162-2_13 (cit. on p. 118)

[Sho87] N. Shor, ”Class of global minimum bounds of polynomial functions”, *Cybernetics*, vol. 23, pp. 731–734, 1987 (cit. on p. 112)

[UF96] A. Urquhart and X. Fu, ”Simplified lower bounds for propositional proofs”, *Notre Dame Journal of Formal Logic*, vol. 37, no. 4, pp. 523–544, 1996 (cit. on p. 123)

[Yan88] M. Yannakakis, ”Expressing combinatorial optimization problems by linear programs”, in *Proceedings of the Twentieth Annual ACM Symposium on Theory of Computing*, ser. STOC ’88, Chicago, Illinois, USA: Association for Computing Machinery, 1988, pp. 223–228, ISBN: 0897912640. DOI: 10.1145/62212.62232. [Online]. Available: https://doi.org/10.1145/62212.62232 (cit. on p. 119)

Paper C

The Minimum Circuit Size Problem is Hard for Sum of Squares

Per Austrin and Kilian Risse

Abstract

We prove lower bounds for the Minimum Circuit Size Problem (MCSP) in the Sum-of-Squares (SoS) proof system. Our main result is that for every Boolean function $f : \{0,1\}^n \to \{0,1\}$, SoS requires degree $\Omega(s^{1-\epsilon})$ to prove that f does not have circuits of size s (for any $s > \text{poly}(n)$).

We also show that for any $0 < \alpha < 1$ there are Boolean functions with circuit complexity larger than 2^{n^α} but SoS requires size $2^{2^{\Omega(n^\alpha)}}$ to prove this. In addition we prove analogous results on the minimum *monotone* circuit size for monotone Boolean slice functions.

C.1 Introduction

Even before the dawn of complexity theory, there was an interest in the minimum circuit size problem (MCSP): given the truth table of a Boolean function $f : \{0,1\}^n \to \{0,1\}$ and a parameter s, the MCSP problem asks whether there is a Boolean circuit of size at most s computing f. Despite many years of research, we do not know whether this problem is NP-hard. It clearly is in NP: given a circuit of size at most s (described by $O(s \log s)$ bits) we can easily check in time $O(s \cdot 2^n)$ whether this circuit indeed computes f.

Determining the hardness of MCSP itself turns out to be a difficult problem. Kabanets and Cai [KC00] showed that NP-hardness of the MCSP problem implies breakthrough circuit lower bounds. These lower bounds are not implausible but well out of reach of current techniques. In a similar vein Murray and Williams [MW15] showed that NP-hardness of MCSP implies that EXP ≠ ZPP and more recently Hirahara [Hir18] proved that NP-hardness of MCSP implies a worst-case to average-case reduction for problems in NP (for an appropriate MCSP version).

On the other hand if one could show that MCSP is in P/poly, this would imply even stronger (though less realistic) results: Kabanets and Cai [KC00] also showed that if MCSP is in P/poly, then crypto-secure one way functions can be inverted on a considerable fraction of their range.

To summarize it seems unlikely that MCSP is in P, but showing that it is NP-hard implies very strong consequences. As these results seem out of reach for current techniques, it might be a more fruitful avenue to try to at least rule out that certain (families of) algorithms solve the MCSP problem efficiently.

This can be achieved very elegantly in proof complexity: show that some proof system capturing your algorithm requires long proofs to refute the claim that a complex function has a small circuit. This will then rule out that the algorithm in question can efficiently solve the MCSP problem. This will not only show that this specific algorithm requires long running time but would also show that any algorithm captured by this proof system requires long running time to solve the MCSP problem. Hence by this line of reasoning we manage to rule out entire classes of algorithms to solve the MCSP problem efficiently.

This paper focuses on the Sum of Squares proof system (SoS). This proof system provides certificates of unsatisfiability of systems of polynomial equations $\mathcal{P} = \{p_1 = 0, \ldots, p_m = 0\}$ over $\mathbb{R}$. A key complexity measure is the degree of a refutation, which is the maximum degree of a monomial occurring in the refutation of $\mathcal{P}$. All Boolean system $\mathcal{P}$ over n variables have an SoS refutation of degree n and we are interested in the minimum degree that SoS requires to refute $\mathcal{P}$. An SoS refutation of degree d has size $O(n^d)$ and can be found in $n^{O(d)}$ time using semidefinite programming and this is often a useful heuristic bound for the complexity of an SoS refutation. The actual size complexity of SoS can sometimes be significantly smaller than n^d, but it is in general not believed that the shortest refutation can be found efficiently. Hence it is in general of interest to understand both the degree and the size needed to refute any given system.

SoS is a very powerful proof system and captures many state of the art algorithms that are based on spectral methods. A classic algorithm captured by SoS is Goemans and Williamson's Max-Cut algorithm [GW95], but also approximate graph coloring algorithms [KMS98], and algorithms solving constraint satisfaction problems [AOW15; RRS17] are captured by SoS. On the other hand SoS has real difficulty arguing about integers and in particular parities. Indeed, Grigoriev [Gri01] showed that SoS requires degree $\Omega(n)$ to refute a random *xor* constraint satisfaction problem of the appropriate (constant) density. After this initial lower bound it took a few years to develop good lower bounds methods for SoS, but in recent years there has been a flurry of strong SoS degree lower bounds [MPW15; BHK+16; KMOW17].

In order to rule out that algorithms captured by SoS can solve MCSP efficiently, we need to encode the claim that a given function has a small circuit as a propositional formula. We work with the encoding suggested by Razborov [Raz98], which encodes this claim that the function $f : \{0, 1\}^n \to \{0, 1\}$ has a circuit of size s by a propositional formula $\mathrm{Circuit}_s(f)$ over $O(s^2 + s \cdot 2^n) = O(s \cdot 2^n)$ variables as follows. We have $\Theta(s^2)$ *structure variables* to encode all possible size s circuits, and for every assignment $\alpha \in \{0, 1\}^n$ we then have an additional $\Theta(s)$ *evaluation variables* that simulate the evaluation of the circuit on each input, and constraints forcing the circuit to output the correct value on each input α.

Apart from its intrinsic interest, variants of the MCSP problem are also (conjectured) sources of hard instances for various proof system. In other words, it is believed that even strong proof systems cannot refute variants of the claim that a complex function has a small circuit and hence has been studied through the lens of proof complexity before: Razborov [Raz98; Raz04] has shown that the pigeonhole principle reduces to the MCSP problem and hence there exist functions f such that $\mathrm{Circuit}_s(f)$ require large refutations in weak proof systems such as resolution or polynomial calculus which are not able to solve the pigeonhole principle. A slightly different line of works [ABRW04; Raz15] introduced pseudorandom generators to proof complexity and show strong lower bounds for such formulas. Razborov [Raz15] outlines a general connection between pseudo-random generator lower bounds and MCSP lower bounds.

To date we have no unconditional degree lower bounds for semi-algebraic proof systems for the $\mathrm{Circuit}_s(f)$ formula and this has also been stated [Raz21; Raz22] as an explicit open problem.

C.1.1 Our Results

Our first result gives a lower bound on the degree needed to refute $\mathrm{Circuit}_s(f)$ in SoS. This lower bound is very general and in fact applies to *every* Boolean function $f : \{0,1\}^n \to \{0,1\}$.

Theorem C.1.1. *For all $\varepsilon > 0$ there is a $d = d(\varepsilon)$ such that the following holds. For $n \in \mathbb{N}$, all $s \geq n^d$ and any Boolean function $f : \{0,1\}^n \to \{0,1\}$ on n bits, SoS requires degree $\Omega_\varepsilon(s^{1-\varepsilon})$ to refute $\mathrm{Circuit}_s(f)$.*

Furthermore, the lower bound of $\Omega_\varepsilon(s^{1-\varepsilon})$ on the degree is essentially tight: if f does not have a circuit of size s then there exists an SoS refutation of this in degree $O(s)$.

Proposition C.1.2. *Let $s \in \mathrm{N}$ and $f : \{0,1\}^n \to \{0,1\}$ be a Boolean function on n bits that requires circuits of size larger than s to be computed. Then there is a degree $O(s)$ SoS refutation of $\mathrm{Circuit}_s(f)$.*

We also prove a result about the minimum size (number of monomials) required for SoS to refute $\mathrm{Circuit}_s(f)$. This result holds for all functions that "almost" have a circuit of size s, in the sense that they have an errorless heuristic circuit (see the survey [BT06]) of size $s/2$ and extremely small error probability with respect to the uniform distribution. Formally, we let $\mathcal{F}_n(s,t)$ denote the class of Boolean functions that consists of all functions $f : \{0,1\}^n \to \{0,1\}$ for which there is a Boolean circuit $C_f : \{0,1\}^n \to \{0,1,\perp\}$ of size at most s such that

1. if $C_f(\alpha) \neq \perp$, then $C_f(\alpha) = f(\alpha)$, and
2. $C_f(\alpha) = \perp$ on at most t inputs.

In other words the circuit C_f computes f correctly on all except t inputs. Note that technically the output of the circuit C_f is two bits with the first one indicating whether the output is $\perp$ or the value of the second bit. We believe that above presentation is more intuitive and hope that the slight abuse of notation causes no confusion. With the class of functions $\mathcal{F}_n(s,t)$ at hand we can state our main SoS size lower bound.

Theorem C.1.3. *For all $\varepsilon > 0$ there is a $d = d(\varepsilon)$ such that the following holds. Let $n \in \mathrm{N}$ and $s \in \mathbb{N}$ such that $s \geq n^d$. If $t \geq s$ and $f \in \mathcal{F}_n(s/2, t)$, then it holds that SoS requires size $\exp\left(\Omega_\varepsilon(s^{2-\varepsilon}/t)\right)$ to refute $\mathrm{Circuit}_s(f)$.*

This yields non-trivial size lower bounds for t as large as $s^{2-\varepsilon}/\omega(1)$. Furthermore, note that once $t \gg s \log s$ there are functions that require such large circuits. For example setting $s = 2^{n^{0.99}}$ and $t = s^{1.5}$, the theorem shows that there are functions f that do not have circuits of size s, but SoS requires size $2^{2^{\Omega(n^{0.99})}}$ to prove this.

It is natural to wonder whether SoS fares better in the monotone setting. In other words, whether SoS can refute the claim that a complex monotone function has a small monotone circuit. The following two theorems show that this is not the case for the set $\mathcal{M}_n(\ell)$ of monotone ℓ-slice functions. Recall that $\mathcal{M}_n(\ell)$ consist of all Boolean functions f on n bits such that $f(\alpha) = 0$ for all α with Hamming weight less than ℓ, and $f(\alpha) = 1$ for all α with Hamming weight greater than ℓ (note that any such f is monotone).

We define a variant $\text{Circuit}^{\text{mon}}_s(f)$ of the $\text{Circuit}_s(f)$ formula, which instead encodes the claim that f has a monotone circuit of size s, and prove the following theorem.

Theorem C.1.4. *For all $\varepsilon > 0$ there is a $d = d(\varepsilon)$ such that the following holds. For all $n, \ell \in \mathbb{N}$, all $s \geq n^d$ and any monotone slice function $f \in \mathcal{M}_n(\ell)$ SoS requires degree $\Omega_\varepsilon(s^{1-\varepsilon})$ to refute* $\text{Circuit}^{\text{mon}}_s(f)$.

As in the non-monotone case, we can also obtain size lower bounds for the monotone-MCSP. Akin to the general size lower bound we consider monotone Boolean slice functions that have good monotone errorless heuristic circuits. Let $\mathcal{M}_n(\ell, s, t) \subseteq \mathcal{M}_n(\ell)$ be the class of monotone Boolean ℓ-slice functions $f : \{0,1\}^n \to \{0,1\}$ for which there is a monotone Boolean circuit $C^{\text{mon}}_f : \{0,1\}^n \to \{0,1,\perp\}$ such that for ℓ-slice inputs $\alpha \in \binom{[n]}{\ell}$ it holds that

1. if $C^{\text{mon}}_f(\alpha) \neq \perp$, then $C^{\text{mon}}_f(\alpha) = f(\alpha)$, and
2. $C^{\text{mon}}_f(\alpha) = \perp$ on at most t inputs.

Theorem C.1.5. *For all $\varepsilon > 0$ there is a $d = d(\varepsilon)$ such that the following holds. For $n, \ell \in \mathbb{N}$, all $s \geq n^d$ and $t \geq s$ and monotone function $f \in \mathcal{M}_n(\ell, s/10, t)$ SoS requires size* $\exp\left(\Omega_\varepsilon(s^{2-\varepsilon}/t)\right)$ *to refute* $\text{Circuit}^{\text{mon}}_s(f)$.

C.1.2 Overview of Proof Techniques

Degree Lower Bound: The main idea that drives our result is a reduction from an expanding *xor* constraint satisfaction problem to the $\text{Circuit}_s(f)$ formula. The reduction is achieved through a careful restriction of the $\text{Circuit}_s(f)$ formula, such that each input $\alpha \in \{0,1\}^n$ to the circuit specifies an *xor* constraint over some new set of variables Y. This will then result in an

XOR-CSP instance with 2^n constraints over the variables Y. All that SoS has to prove is that there is no satisfying assignment to this XOR-CSP instance. By ensuring that the constraint-variable incidence graph is sufficiently expanding, SoS requires large degree to refute the restricted formula (see Theorem C.2.5). At the same time, we need the constraint graph to be very explicit so that it can be encoded into a small circuit. For this we utilize a construction of unbalanced expanders by Guruswami et al. [GUV09] (see Theorem C.2.2). This reduction then immediately yields Theorem C.1.1.

This lower bound may also be viewed as implementing the general program sketched by Razborov [Raz15] relating pseudorandom generators in proof complexity to the MCSP problem. However, we prefer to describe it as a direct reduction to the MCSP problem.

Size Lower Bound: In order to obtain size lower bounds, we would like to apply the degree-size tradeoff due to Atserias and Hakoniemi [AH19] to Theorem C.1.1. Unfortunately the formula is over too many variables to be able to conclude a meaningful size lower bound: it is defined over roughly $\Omega(2^n \cdot s)$ variables.

Instead of applying Theorem C.1.1, we restrict our attention to functions with all except the at most t $\perp$-outputs computed by the corresponding errorrless heuristic circuit. If we choose t small enough, then we are able to heavily restrict $\mathrm{Circuit}_s(f)$ and significantly reduce the number of variables to the point where the Atserias-Hakoniemi degree-size tradeoff is applicable.

Monotone Circuits: We prove these theorems by adapting the proofs for the non-monotone setting. The idea is to work over the ℓth slice and disregard all other inputs. The key feature that makes this work is the fact that the monotone circuit complexity of a slice function is essentially the same as the (ordinary) circuit complexity (see Lemma C.2.4). This lets us convert all subcircuits used in the reduction to small monotone circuits (if we only work on the slice).

The size lower bound goes along the same lines as the proof of Theorem C.1.3.

C.1.3 Organization

In Section C.2, we provide the necessary background material. In Section C.3 we set up the general framework for our lower bounds with some preliminary definitions and lemmas. Then in Section C.4 we prove the main degree Theorem C.1.1 and size Theorem C.1.3 lower bounds. We prove the monotone lower bounds Theorem C.1.4 and Theorem C.1.5 in Section C.5.

In Section C.6 we explain how SoS of degree $O(s)$ can refute $\mathrm{Circuit}_s(f)$ (provided f does not have a circuit of size s). Finally in Section C.8 we give some concluding remarks.

C.2 Preliminaries

All logarithms are in base 2. For integers $n \geq 1$ we write $[n] = \{1, 2, \ldots, n\}$ and for a set U we denote the power set of U by 2^U. Further, for a set $V \subseteq U$ we let $\overline{V}$ be the complement of V with respect to U, that is, $\overline{V} = U \setminus V$. We write $\binom{[n]}{\ell} \subseteq \{0,1\}^n$ for the set of binary strings with Hamming weight ℓ. For a string $\alpha \in \{0,1\}^n$ we let $|\alpha| = \sum_{i \in [n]} \alpha_i$.

We sometimes want to supress dependencies on constants and write $f(n, \varepsilon) \in O_\varepsilon\big(g(n, \varepsilon)\big)$, respectively $f(n, \varepsilon) \in \Omega_\varepsilon\big(g(n, \varepsilon)\big)$, to mean that there exists a function $c(\varepsilon) > 0$ such that there is an n_0 and for all $n \geq n_0$ it holds that $f(n, \varepsilon) \leq c(\varepsilon) \cdot g(n, \varepsilon)$, respectively $f(n, \varepsilon) \geq c(\varepsilon) \cdot g(n, \varepsilon)$.

Definition C.2.1. A sequence of bipartite graphs $\{G_n = (U_n, V_n, E_n)\}_{n \in \mathbb{N}}$ with $\deg(u) = d$ for all $u \in U_n$ is *explicit* if there is an algorithm that given (n, u, j), where $n \in \mathbb{N}, u \in U_n$ and $j \in [d]$, computes the jth neighbor of vertex u in the graph G_n in time $\mathrm{poly}(\log n + \log |U| + \log d)$.

From now on it is understood that whenever we talk about an explicit graph we actually mean to say that there is a sequence of explicit graphs with above properties.

A bipartite graph $G = (U, V, E)$ is an (r, d, c)-expander if every vertex $u \in U$ has degree $\deg(u) = d$ and every set $W \subseteq U$ of size $|W| \leq r$ satisfies $|N(W)| \geq c \cdot |W|$. A key ingredient in our proofs is the following result on the existence of strong explicit expanders.

Theorem C.2.2 ([GUV09]). *For all constants $\gamma > 0$, every $M \in \mathbb{N}$, $r \leq M$, and $\varepsilon > 0$, there is an $N \leq d^2 \cdot r^{1+\gamma}$ and an explicit $(r, d, (1-\varepsilon)d)$-expander $G = (U, V, E)$, with $|U| = M$, $|V| = N$, and $d = O\big(((\log M)(\log r)/\varepsilon)^{1+1/\gamma}\big)$.*

For our purposes it is more relevant to compute the neighbor relation $\mathrm{Neigh}(u, v)$ indicating whether $(u, v) \in E$ rather than the neighbor function as in Definition C.2.1, but this is an immediate consequence of being able to compute the neighbor function.

Claim C.2.3. *If $G = (U, V, E)$ is explicit then the neighbor relation $\mathrm{Neigh} : U \times V \to \{0,1\}$ is computable by a circuit of size $d \cdot \big(\mathrm{poly}(\log n + \log |U| + \log d) + 2 \log |V| + 1\big)$.*

A slice function is a Boolean function f such that there is a $\ell \in [n]$ with $f(\alpha) = 0$ whenever $|\alpha| < \ell$, and $f(\alpha) = 1$ whenever $|\alpha| > \ell$. Note that all slice functions are monotone.

The circuit complexity $\mathrm{Size}_{\mathrm{circ}}(f)$ of a Boolean function f is the size of the smallest circuit over the basis $\vee, \wedge$, and $\neg$ (with fan-in 2). Similarly the monotone circuit complexity $\mathrm{Size}^{\mathrm{mon}}_{\mathrm{circ}}(f)$ of a monotone Boolean function f is the size of the smallest circuit over the basis $\vee$, and $\wedge$. We have the following useful inequality between these measures.

Lemma C.2.4 ([Ber82])**.** *If g is any slice function on n bits, then $\mathrm{Size}^{\mathrm{mon}}_{\mathrm{circ}}(g) \leq 2\,\mathrm{Size}_{\mathrm{circ}}(g) + O(n^2 \log n)$.*

C.2.1 Sum of Squares

Let $\mathcal{P} = \{p_1 = 0, \ldots, p_m = 0\}$ be a system of polynomial equations over the set of variables $X = \{x_1, \ldots, x_n, \bar{x}_1, \ldots, \bar{x}_n\}$. Each p_i is called an axiom, and throughout the paper we always assume that $\mathcal{P}$ includes all axioms $x_i^2 - x_i$ and $\bar{x}_i^2 - \bar{x}_i$, ensuring that the variables are Boolean, as well as the axioms $1 - x_i - \bar{x}_i$, making sure that the "bar" variables are in fact the negation of the "non-bar" variables.

Sum-of-Squares (SoS) is a static semi-algebraic proof system. An SoS proof of $f \geq 0$ from $\mathcal{P}$ is a sequence of polynomials $\pi = (t_1, \ldots, t_m; s_1, \ldots, s_a)$ such that

$$\sum_{i \in [m]} t_i p_i + \sum_{i \in [a]} s_i^2 = f \quad . \tag{C.1}$$

The *degree* of a proof π is

$$\mathrm{Deg}(\pi) = \max\{\max_{i \in [m]} \deg(t_i) + \deg(p_i), \max_{i \in [a]} 2\deg(s_i)\} \quad . \tag{C.2}$$

An *SoS refutation of $\mathcal{P}$* is an SoS proof of $-1 \geq 0$ from $\mathcal{P}$, and the SoS degree to refute $\mathcal{P}$ is the minimum degree of any SoS refutation of $\mathcal{P}$: if we let π range over all SoS refutations of $\mathcal{P}$, we can write $\mathrm{Deg}_{\mathrm{SoS}}(\mathcal{P}) = \min_\pi \mathrm{Deg}(\pi)$.

The size of an SoS refutation π, $\mathrm{Size}(\pi)$, is the sum of the number of monomials in each polynomial in π and the size of refuting $\mathcal{P}$ is the minimum size over all refutations $\mathrm{Size}_{\mathrm{SoS}}(\mathcal{P}) = \min_\pi \mathrm{Size}(\pi)$.

Let us recall some well-known results about SoS. Given a bipartite graph $G = (U, V, E)$, and $b \in \{0,1\}^{|U|}$ we denote by $\Phi(G, b)$ the following XOR-CSP instance defined over G: for each $v \in V$ there is a Boolean variable x_v, and for every vertex $u \in U$ there is a constraint $\oplus_{v \in N(u)} x_v = b_u$. We endoce this in the obvious way as a system of polynomial equations:

$$\{\prod_{v \in N(u)} (1 - 2 \cdot x_v) = 1 - 2 \cdot b_u \mid u \in U\} \quad ,$$

along with the Boolean axioms and the negation axioms for the x variables. The first theorem we need to recall is the classic lower bounds for XOR-CSPs by Grigoriev.

Theorem C.2.5 ([Gri01]). *For $n \in \mathbb{N}$, all $k = k(n)$ and $r = r(n)$ the following holds. Let $G = (U, V, E)$ be an $(r, k, 2)$-expander with $|V| = n$. Then for every $b \in \{0,1\}^{|U|}$ SoS requires degree $\Omega(r)$ to refute the claim that there is a satisfying assignment to $\Phi(G, b)$.*

We also need to recall the size-degree tradeoff by Atserias and Hakoniemi.

Theorem C.2.6 ([AH19]). *Let $\mathcal{P}$ be a system of polynomial equations over n Boolean variables and degree at most k. If d is the minimum degree SoS requires to refute $\mathcal{P}$, then the minimum size of an SoS refutation of $\mathcal{P}$ is at least $\exp(\Omega((d-k)^2/n))$.*

C.2.2 Restrictions

Let $\mathcal{P} = \{p_1 = 0, \dots, p_m = 0\}$ be a system of polynomial equations over the set of Boolean variables $X = \{x_1, \dots, x_n, \bar{x}_1, \dots, \bar{x}_n\}$. For a map $\rho : \{x_1, \dots, x_n\} \to \{0, 1, x_1, \dots, x_n, \bar{x}_1, \dots, \bar{x}_n\}$ denote by $\mathcal{P}\lceil_\rho$ the system of polynomial equations $\mathcal{P}$ restricted by ρ, i.e.,

$$\begin{aligned}\mathcal{P}\lceil_\rho = \{&p_1(\rho(x_1), \dots, \rho(x_n)) = 0,\\ &p_2(\rho(x_1), \dots, \rho(x_n)) = 0,\\ &\vdots\\ &p_m(\rho(x_1), \dots, \rho(x_n)) = 0\}\ ,\end{aligned}$$

where it is understood that $\rho(\bar{x}_i) = \overline{\rho(x_i)}$, with the convention $\bar{\bar{x}}_i = x_i$, $\bar{0} = 1$ and vice versa. Throughout the paper all our restrictions set the bar variables to the negation of the non-bar variables. As such it makes sense to treat the pair of variables $(x_i, \bar{x}_i)$ as one variable and we say that $\mathcal{P}$ has n *unset* variables.

We say that a system of polynomial equations $\mathcal{P}'$ is an *affine restriction of* $\mathcal{P}$ if there is a map $\rho : \{x_1, \dots, x_n\} \to \{0, 1, x_1, \dots, x_n, \bar{x}_1, \dots, \bar{x}_n\}$ such that $\mathcal{P}' = \mathcal{P}\lceil_\rho$, where we ignore polynomial equations of the form $0 = 0$. The following well-known lemma states that a system of polynomial equations $\mathcal{P}$ is at least as hard as any of its affine restrictions.

Lemma C.2.7. *Let $\mathcal{P}, \mathcal{P}'$ be systems of polynomial equations such that $\mathcal{P}'$ is an affine restriction of $\mathcal{P}$. Then,*

(i) $\mathrm{Deg}_{\mathrm{SoS}}(\mathcal{P}) \geq \mathrm{Deg}_{\mathrm{SoS}}(\mathcal{P}')$, *and*

(ii) $\mathrm{Size}_{\mathrm{SoS}}(\mathcal{P}) \geq \mathrm{Size}_{\mathrm{SoS}}(\mathcal{P}')$.

The lemma is easy to verify by considering an SoS refutation of $\mathcal{P}$ and hitting it with the appropriate affine restriction. The restricted proof is now a refutation of $\mathcal{P}'$ and it can be seen that the degree/size of the restricted refutation is at most the degree/size of the original refutation.

We also consider more general restrictions: restrictions $\rho : \{x_1, \ldots, x_n\} \rightarrow \mathbb{R}[x]_{\leq k}$ that map variables to polynomials of degree at most k. For bar variables, we let $\rho(\bar{x}_i) = 1 - \rho(x_i)$. For such general restrictions we have the following well-known lemma.

Lemma C.2.8. *Let $\mathcal{P}$ be a system of polynomial equations and let ρ be a restriction mapping variables to polynomials of degree at most k. Then, $\mathrm{Deg}_{\mathrm{SoS}}(\mathcal{P}) \geq \mathrm{Deg}_{\mathrm{SoS}}(\mathcal{P}\restriction_{\rho})/k$.*

This lemma can again be verified by considering a refutation of $\mathcal{P}$. Substitute each variable x_i in the proof by $\rho(x_i)$. This results in a refutation of $\mathcal{P}\restriction_{\rho}$, whose degree is at most a factor k larger than the degree of the refutation of $\mathcal{P}$.

C.2.3 The Circuit Size Formula

The formula $\mathrm{Circuit}_s(f)$ encodes the claim that the function f, given as a truthtable $f \in \{0,1\}^{2^n}$, can be computed by a circuit of size s over n Boolean inputs $x_1, \ldots, x_n$. The encoding is not essential but for concreteness let us fix one encoding of this claim. We deviate from the encoding used by Razborov [Raz98; Raz04] and do not present the formula as a propositional formula but rather as a system of polynomial equations. In order to encode below constraints as a constant width CNF formula, as done by Razborov, one needs to introduce extension variables. Despite this difference it is not difficult to see that our lower bound also works against the CNF encoding. In Section C.7 we directly show that a low degree SoS refutation of the CNF encoding gives rise to a low degree SoS refutation of the encoding used in this paper. Thus a lower bound for our encoding implies a lower bound for the CNF encoding. As the presentation is simpler in the polynomial encoding, we present it as follows.

We also need to define the monotone version of $\mathrm{Circuit}_s(f)$ denoted by $\mathrm{Circuit}_s^{\mathrm{mon}}(f)$. The later is a restriction of the former with the $\mathrm{isNeg}(\nu)$ (see below) variable, for all $\nu \in [s]$, set to 0. This forces the circuit to only contain $\wedge$ and $\vee$ gates, i.e., the circuit is monotone.

All variables introduced in the following are Boolean variables and we implicitly add the Boolean axiom $y(1-y) = 0$ for each variable y and further implicitly introduce the "bar variable" $\bar{y}$ along with the negation axiom $y = 1 - \bar{y}$ (and the corresponding Boolean axiom) ensuring that $\bar{y}$ is always the negation of y.

Let us first describe the *structure variables* which are used to describe the circuit that supposedly computes the function f.

We view the s gates as being indexed from 1 to s in topological order with gate s being the output. For each gate $v \in [s]$ there are three variables $\mathrm{isNeg}(v), \mathrm{isOr}(v), \mathrm{isAnd}(v)$ indicating the operation computed at v. Similarly for a gate $v \in [s]$ and a wire $a \in \{1, 2\}$ we have variables $\mathrm{isFromConst}(v, a), \mathrm{isFromInput}(v, a), \mathrm{isFromGate}(v, a)$ indicating whether the input wire a of v is connected to a constant, a variable or a gate.

Further, we have the variables $\mathrm{constantValue}(v, a)$, $\mathrm{isInput}(v, a, i)$ and $\mathrm{isGate}(v, a, u)$, for $a \in \{1, 2\}$, $i \in [n]$ and $u < v$, specifying the constant value, input x_i or gate u, the input wire a of v is connected to (assuming a is connected to the corresponding kind).

The second set of variables are the *evaluation variables*, which describe what value is computed at each v on input $\alpha = \alpha_1, \ldots, \alpha_n$ (i.e., we have $x_i = \alpha_i$).

For each gate $v \in [s]$ and assignment $\alpha \in \{0, 1\}^n$ we have a Boolean variable $\mathrm{out}_\alpha(v)$ indicating the value computed at gate v on input α. The Boolean variable $\mathrm{in}_\alpha(v, a)$ indicates the value brought to the vertex $v \in [s]$ on wire $a \in \{1, 2\}$ on input α.

Note that there is a total of $3s + 6s + 2s + 2sn + 2\binom{s}{2} = \Theta(s^2 + sn)$ structure variables, and a total of $3s2^n$ evaluation variables, for a total of $\Theta(s^2 + s2^n)$ variables in $\mathrm{Circuit}_s(f)$.

The formula consists of the following axioms. Let us first describe the axioms on the structure of the circuit. In the following section we refer to this set of axioms as the *structure axioms*. The first axioms ensure that every wire is connected to a single kind

$$\mathrm{isFromConst}(v, a) + \mathrm{isFromInput}(v, a) + \mathrm{isFromGate}(v, a) = 1 \quad \forall\, v \in [s] \ , \tag{C.3}$$

and similarly the next axioms make sure that each gate is of precisely one kind

$$\mathrm{isNeg}(v) + \mathrm{isOr}(v) + \mathrm{isAnd}(v) = 1 \quad \forall\, v \in [s] \ . \tag{C.4}$$

The final structure axioms ensure that the variables, which indicate to what input or gate a fixed wire is connected to, always sum to one (except for gate 1 which cannot have any inputs from other gates)

$$\sum_{i=1}^{n} \mathrm{isInput}(v, a, i) = 1 \quad \forall v \in [s], \text{ and} \tag{C.5}$$

$$\sum_{u=1}^{v-1} \mathrm{isGate}(v, a, u) = 1 \quad \forall v \in [s] \setminus \{1\} \ . \tag{C.6}$$

We further strengthen our encoding by adding the axioms

$$\mathrm{isInput}(v, a, i)\mathrm{isInput}(v, a, j) = 0 \quad \forall v \in [s],\ i < j \in [n], \text{ and} \tag{C.7}$$

$$\mathrm{isGate}(v, a, u)\mathrm{isGate}(v, a, u') = 0 \quad \forall v \in [s] \setminus \{1\},\ u < u' < v \ . \tag{C.8}$$

Note that Axioms C.8 and C.7 are implied by Axioms C.5. We add these axioms in order to argue that a short refutation of the CNF encoding of this principle leads to a short refutation of the present encoding.

The second group of axioms are the *evaluation axioms* and they ensure that the evaluation variables indeed compute the intended values. We start by making sure that the wires carry the value intended by the structure axioms. If a wire is connected to a constant, then the evaluation variable associated with that wire should always be equal to the constant

$$\mathrm{isFromConst}(v, a) \cdot \big(\mathrm{in}_\alpha(v, a) - \mathrm{constantValue}(v, a)\big) = 0 \ , \tag{C.9}$$

and similarly in case if a wire is connected to an input or a gate

$$\mathrm{isFromInput}(v, a) \cdot \mathrm{isInput}(v, a, i) \cdot \big(\mathrm{in}_\alpha(v, a) - \alpha_i\big) = 0 \ , \tag{C.10}$$

$$\mathrm{isFromGate}(v, a) \cdot \mathrm{isGate}(v, a, u) \cdot \big(\mathrm{in}_\alpha(v, a) - \mathrm{out}_\alpha(u)\big) = 0 \ . \tag{C.11}$$

The final set of evaluation axioms makes sure that the output evaluation variable of a gate is correctly related to the input evaluation variables:

$$\mathrm{isNeg}(v) \cdot \mathrm{out}_\alpha(v) = \mathrm{isNeg}(v) \cdot \overline{\mathrm{in}_\alpha(v, 1)} \ , \tag{C.12}$$

$$\mathrm{isOr}(v) \cdot \mathrm{out}_\alpha(v) = \mathrm{isOr}(v) \cdot \big(1 - \overline{\mathrm{in}_\alpha(v, 1)} \cdot \overline{\mathrm{in}_\alpha(v, 2)}\big) \ , \tag{C.13}$$

$$\mathrm{isAnd}(v) \cdot \mathrm{out}_\alpha(v) = \mathrm{isAnd}(v) \cdot \mathrm{in}_\alpha(v, 1) \cdot \mathrm{in}_\alpha(v, 2) \ . \tag{C.14}$$

Last but not least we have the axioms that ensure that the circuit outputs the function specified by the truthtable

$$\mathrm{out}_\alpha(s) = f(\alpha) \ . \tag{C.15}$$

C.3 On Circuits and Restrictions

Let $G = (U, V, E)$ be a bipartite graph with $U = \{0,1\}^n$ and $V = [m]$. As in the XOR-CSP setup (Section C.2.1) we think of vertices in U as constraints and vertices in V as variables. More specifically, we think of each vertex $\alpha \in U$ as an *xor* constraint over the variables in the neighborhood $\oplus_{i \in N(\alpha)} v_i = b_\alpha$, for a constraint vector $b \in \{0,1\}^U$. Given an assignment $\beta \in \{0,1\}^m$ to the variables V, we let $f_{G,\beta} : U \to \{0,1\}$ be the function defined by $f_{G,\beta}(\alpha) = \oplus_{i \in N(\alpha)} v_i$. In other words, viewing $f_{G,\beta}$ as a vector in $\{0,1\}^U$, it is the unique constraint vector such that the XOR-CSP instance, defined over G, is satisfied by the assignment β. Let us denote the set of all such constraint vectors that give rise to a satisfiable XOR-CSP instance by

$$\mathcal{F}_G = \{f_{G,\beta} \mid \beta \in \{0,1\}^m\} \enspace .$$

In order for SoS to refute an XOR-CSP instance defined over G, it must prove that the given constraint vector is not in the set $\mathcal{F}_G$.

On the other hand in order for SoS to refute the formula $\mathrm{Circuit}_s(f)$ it needs to show that there is no circuit of size at most s computing f. That is, SoS needs to show that f is not in the set

$$\mathcal{C}_\emptyset = \{T : \{0,1\}^n \to \{0,1\} \text{ such that } \mathrm{Circuit}_s(T) \text{ is satisfiable}\} \enspace .$$

More generally, if we restrict $\mathrm{Circuit}_s(f)$ by a restriction ρ, then the proof system must prove that f is not a member of the family of truthtables

$$\mathcal{C}_\rho = \{T : \{0,1\}^n \to \{0,1\} \text{ such that } \mathrm{Circuit}_s(T)\restriction_\rho \text{ is satisfiable}\} \enspace .$$

In the following we show that there is a well-behaved restriction ρ such that $\mathcal{C}_\rho = \mathcal{F}_G$ for some explicit graphs G. In other words, once we consider the formula $\mathrm{Circuit}_s(f)\restriction_\rho$, all that SoS needs to do is to rule out that f is a valid right hand side of an XOR-CSP instance. But we know that if G is a moderate expander, then low degree SoS cannot determine wheter the XOR-CSP instance is satisfiable and hence we obtain our lower bound.

Let us first formalize the properties we require from ρ. We start off by restricting our attention to a certain natural class of affine restrictions. Namely, we do not want that the structure of the circuit depends on evaluation variables.

Definition C.3.1 (natural affine restrictions)**.** An affine restriction ρ to the variables of $\mathrm{Circuit}_s(f)$ is *natural* if there is *no* structure variable y such that $\rho(y)$ is an evaluation variable.

In order to motivate the following definition, let us informally describe the natural restriction ρ and explain the properties of ρ we require.

For now we can think of ρ as a restriction to the structure variables (though for the size lower bounds we also need to restrict the evaluation variables). Some set of m structure variables remains undetermined. Let us denote these variables by $y_1, \dots, y_m$. We intend to choose ρ such that on a given input $\alpha \in \{0,1\}^n$ to the circuit, it is forced to compute $\oplus_{i \in N(\alpha)} y_i$. In other words, given such a restriction ρ, we are *essentially* left with an XOR-CSP problem over G, with right hand side f. There is however a difference in that the encoding is non-standard: the evaluation variables act like extension variables that correspond to the functions computed at each gate of the circuit. In order to argue that the known degree lower bound for the XOR-CSP problem implies a degree lower bound for the problem at hand, we need to get rid of these extension variables. This can be done if the functions computed at the gates are of low degree and this is precisely what we require of such variables.

Recall from Section C.2.2 that a system of polynomial equations $\mathcal{P}$ has n unset variables if there are n tuples of variables $(x, \bar{x})$ such that at least one variable of each tuple occurs in $\mathcal{P}$ and all variables in these tuples are unset, i.e., they are not fixed to a constant.

Definition C.3.2 (k-determined)**.** Let ρ be an affine restriction to the variables of $\mathrm{Circuit}_s(f)$ and suppose that ρ leaves m structural variables $Y = \{y_1, \dots, y_m\}$ unset. Then ρ is k*-determined* if for every $v \in [s]$ and $\alpha \in \{0,1\}^n$ there are functions

$$g^{\mathrm{out}}_{v,\alpha}, g^{\mathrm{in}_1}_{v,\alpha}, g^{\mathrm{in}_2}_{v,\alpha} : \{0,1\}^m \to \{0,1\}$$

depending on at most k variables such that the following holds. For all $T \in \mathcal{C}_\rho$ and all total assignments σ that satisfy $\mathrm{Circuit}_s(T)\lceil_\rho$ it holds that

$$\begin{aligned} \mathrm{out}_\alpha(v)\lceil_{\rho\cup\sigma} &= g^{\mathrm{out}}_{v,\alpha}(\beta) \ , \\ \mathrm{in}_\alpha(v,1)\lceil_{\rho\cup\sigma} &= g^{\mathrm{in}_1}_{v,\alpha}(\beta) \ , \text{ and} \\ \mathrm{in}_\alpha(v,2)\lceil_{\rho\cup\sigma} &= g^{\mathrm{in}_2}_{v,\alpha}(\beta) \ , \end{aligned}$$

where $\beta \subseteq \sigma$ is the assignment to Y.

As the Y variables are Boolean variables, we may assume that all functions $g^{\cdot}_{v,\alpha}(Y)$ are multilinear and thus of degree at most k. We associate each k-determined restriction ρ with a substitution $\tau(\rho)$ that extends ρ by substituting all evaluation variables by the appropriate polynomials of degree at most k in the Y variables. Note that the resulting formula $\mathrm{Circuit}_s(f)\lceil_{\tau(\rho)}$ is defined over the variables Y.

However, Definition C.3.2 is not quite sufficient. For example, there is no guarantee that $\mathcal{C}_\rho$ is non-empty, i.e., that the restriction ρ describes a valid (partial) circuit. More generally, we need the additional guarantee that there are still many viable circuits that the restricted formula can describe: if there is just a single setting of the Y variables such that all structural axioms are satisfied, then the formula may be refuted in constant degree. Hence we need to ensure that there are many viable assignments to the Y variables that satisfy all structure axioms. This leads us to the following definition.

Definition C.3.3 (m-independent)**.** An affine restriction ρ to the variables of the formula $\mathrm{Circuit}_s(f)$ is m-*independent* if ρ leaves exactly m structural variables $Y = \{y_1, \dots, y_m\}$ unset, and for every assignment $\beta \in \{0,1\}^Y$ it holds that $|\mathcal{C}_{\rho\cup\beta}| = 1$.

The following claim shows that under a natural m-independent affine restriction all structure axioms are satisfied (modulo the negation axioms).

Claim C.3.4. *Let ρ be a natural m-independent affine restriction and let $p = 0$ be a structure axiom. Then p under ρ can either be written as a linear combination of the negation axioms, i.e., $p\lceil_\rho = \sum_{i\in[m]} \gamma_i(y_i + \bar{y}_i - 1)$, for $\gamma_i \in \mathbb{R}$, or it holds that $p\lceil_\rho = y_i\bar{y}_i$.*

Proof. Denote by Y the m structural variables that are left unset by ρ. Consider any structural axiom $p\lceil_\rho = 0$. If p is one of the Axioms C.8 and C.7, then it holds that either $p\lceil_\rho$ is equal to 0 or $p\lceil_\rho = y_i\bar{y}_i$ for some structural variable y_i – otherwise ρ would not be affine, natural and m-independent.

As all structure axioms are of degree 1 and ρ is an affine restriction we have that otherwise $p\lceil_\rho$ is of degree 1. Furthermore, as ρ is natural $p\lceil_\rho$ is in fact a polynomial in the Y variables. Thus, as ρ is m-independent, for all assignments $\beta \in \{0,1\}^Y$, it holds that $p\lceil_{\rho\beta} = 0$ (where we extend β to the bar variables as in Section C.2.2).

We conclude that $p\lceil_\rho$ is a degree 1 polynomial that evaluates to 0 on all Boolean assignments to Y that respect the negation axioms. Put differently, the polynomial $p\lceil_\rho$ is equal to 0 modulo the negation axioms of Y.

What remains to argue is that we can write $p\lceil_\rho$ as a *linear combination* of the negation axioms of Y. To this end we claim that the variables y and $\bar{y}$ occur with the same coefficient in $p\lceil_\rho$: suppose otherwise and fix an assignment to all other variables. If the coefficients differ, then either setting y to 1 or 0 causes the polynomial to evaluate to non-zero, in contradiction to the assumption that ρ is m-independent.

Thus we can subtract the negation axiom $y + \bar{y} - 1$ appropriately scaled from $p\lceil_\rho$. This results in a polynomial on fewer variables that still evaluates to 0. We can thus repeat this argument to recover $p\lceil_\rho$ as a linear combination of negation axioms, as required. □

With these definitions at hand we can prove the lemma that drives all our lower bounds.

Lemma C.3.5. *Let* ρ *be a natural* m*-independent* k*-determined affine restriction of* $\mathrm{Circuit}_s(f)$*, and let* Y *and* $g^{\mathrm{out}}_{u,\alpha}$ *be as in Definition C.3.2. If there is an SoS refutation of* $\mathrm{Circuit}_s(f)\lceil_\rho$ *of degree* d*, then there is a degree* $O(d\cdot k)$ *SoS refutation of the system of polynomial equations*

$$\{g^{\mathrm{out}}_{s,\alpha}(Y) = f(\alpha) \mid \alpha \in \{0,1\}^n\} \cup \{y_i^2 = y_i, \bar{y}_i^2 = \bar{y}_i, y_i = 1 - \bar{y}_i \mid i \in [m]\} \ .$$

Proof. The idea is to replace all evaluation variables in a refutation by the corresponding functions $g^{\mathrm{out}}_{v,\alpha}(Y)$, $g^{\mathrm{in}_1}_{v,\alpha}(Y)$, and $g^{\mathrm{in}_2}_{v,\alpha}(Y)$. As all these functions are of degree at most k and ρ is m-independent we get the desired statement.

More precisely, apply $\tau(\rho)$ to a degree d SoS refutation of the formula $\mathrm{Circuit}_s(f)\lceil_\rho$. As noted previously, the resulting formula is over the Y variables. In the following we argue that all axioms of $\mathrm{Circuit}_s(f)\lceil_{\tau(\rho)}$ can be derived from

$$\mathcal{P} = \{g^{\mathrm{out}}_{s,\alpha}(Y) = f(\alpha) \mid \alpha \in \{0,1\}^n\} \cup \{y_i^2 = y_i, \bar{y}_i^2 = \bar{y}_i, y_i = 1 - \bar{y}_i \mid i \in [m]\} \tag{C.16}$$

in degree at most $O(k)$. This is enough to conclude the lemma: the substitution $\tau(\rho)$ increases the degree of the refutation by at most a factor $O(k)$, and all axioms of the restricted formula $\mathrm{Circuit}_s(f)\lceil_{\tau(\rho)}$ can be derived in degree at most $O(k)$ from $\mathcal{P}$. We thus obtain a degree $O(d \cdot k)$ SoS refutation of $\mathcal{P}$.

As ρ is natural and m-independent, by Claim C.3.4, the structure axioms are derivable in constant degree from the negation axioms and Boolean axioms of $\mathcal{P}$. All that remains is to argue that the evalutation axioms can be derived in degree $O(k)$. By the assumption that ρ is k-determined and as the evaluation axioms are on $O(1)$ variables, we see that the axioms are mapped to polynomials that depend on at most $O(k)$ many variables. Because ρ is natural and m-independent, for any $\beta \in \{0,1\}^Y$, the functions $g^{\mathrm{out}}_{v,\alpha}(\beta)$, $g^{\mathrm{in}_1}_{v,\alpha}(\beta)$ and $g^{\mathrm{in}_2}_{v,\alpha}(\beta)$ correspond to a proper evaluation of the variables $\mathrm{out}_\alpha(v)$, $\mathrm{in}_\alpha(v,1)$ and $\mathrm{in}_\alpha(v,2)$ in the circuit $C_{\rho\cup\beta}$ under the input α. A proper evaluation satisfies all evaluation axioms and thus the substituted evaluation axioms are equal to 0 modulo the Boolean axioms and the negation axioms of Y. As all the substituted evaluation axioms are defined over $O(k)$ variables, these derivations can be performed in degree $O(k)$ as required. □

C.4 Lower Bounds for General Circuits

We state the following lemma general enough so that we can apply it for the degree as well as the size lower bound. As explained previously, for the size lower bounds we rely on functions that almost have circuits of size s. Recall that we consider the class of functions $\mathcal{F}_n(s,t)$ that consists of all Boolean functions $f : \{0,1\}^n \to \{0,1\}$ for which there is a Boolean circuit $C_f : \{0,1\}^n \to \{0,1,\perp\}$ of size at most s such that

1. if $C_f(\alpha) \neq \perp$, then $C_f(\alpha) = f(\alpha)$, and
2. $C_f(\alpha) = \perp$ on at most t inputs.

The following lemma establishes the existence of m-independent k-determined affine restrictions that result in XOR-CSP instances over explicit graphs.

Lemma C.4.1. *For all* $k, m, n, t \in \mathbb{N}$ *satisfying* $m \leq 2^n$, *and any explicit bipartite graph* $G = (U, V, E)$ *such that* $|U| = 2^n$, $|V| = m$ *and all* $u \in U$ *are of degree* $\deg(u) \leq k$, *the following holds. There is a constant* $C > 0$, *depending on the explicitness of* G, *such that for all* $s \geq C \cdot m \cdot n^C \cdot k^C$ *and any Boolean function* $f \in \mathcal{F}_n(s/2, t)$ *there is a natural* m*-independent* k*-determined affine restriction* ρ *for the formula* $\mathrm{Circuit}_s(f)$ *such that*

$$g^{\mathrm{out}}_{s,\alpha}(Y) = \begin{cases} f(\alpha), & \textit{if } C_f(\alpha) \neq \perp, \\ \bigoplus_{i \in N(\alpha)} y_i, & \textit{otherwise} \end{cases}$$

for all $\alpha \in \{0,1\}^n$ *and* $g^{\mathrm{out}}_{s,\alpha}$ *and* Y *as in Definition C.3.2.*

Furthermore, the formula $\mathrm{Circuit}_s(f)\lceil_\rho$ *is over* $O\big(t \cdot k + m\big)$ *variables.*

For the degree lower bound (Theorem C.1.1) we will set $t = 2^n$ and use the trivial C_f which always outputs $\perp$, so the reader who wishes a simplified version of the lemma can focus on this special case.

Proof. We consider the formula $\mathrm{Circuit}_s(f)$ and let the first m gates of the formula be denoted by Y. We restrict the formula such that each gate in Y computes an *or* of two constants. The first wire to the gate is fixed to the constant 0, whereas the second wire is only restricted to carry either the constant 0 or 1. In the end these will be the only structural variables that are not restricted to a constant. In the following we think of the gates Y as Boolean variables; as m additional input bits to our circuit.

Further, we restrict another part of the formula such that one part of the circuit described by the formula computes the circuit C_f. Recall that we pretend that the output of C_f is in $\{0,1,\perp\}$, but it actually outputs two bits C_f^1 and C_f^2, where the first bit is 1 on an input α if and only if $C_f^2(\alpha) = f(\alpha)$.

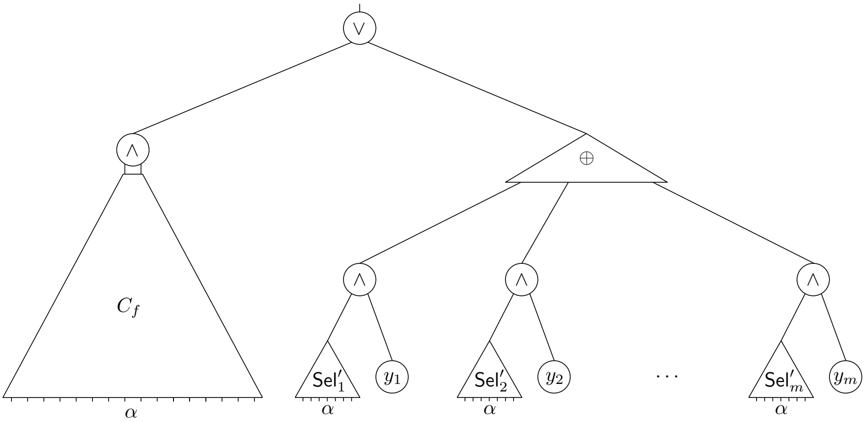


Figure C.1: A schematic depiction of the formula after hitting it with the described restriction.

Finally we also want to hard code the bipartite graph $G(\{0,1\}^n, Y, E)$ into our circuit. Since G is very large this requires G to be explicit. That is, we require small circuits $\mathrm{Sel}_1, \dots, \mathrm{Sel}_m$, where each Sel_i computes, given any $\alpha \in \{0,1\}^n$, whether the vertex $y_i \in Y$ is a neighbor of the vertex α. By Claim C.2.3 these circuits Sel_i are each of size

$$k \cdot \big(\mathrm{poly}(n + \log k) + 2\log m + 1\big) \le \mathrm{poly}(n, k) \enspace .$$

The restriction ρ restricts some structural variables such that a part of the circuit computes $\mathrm{Sel}_1, \dots, \mathrm{Sel}_m$. We connect each output of the Sel_i circuit by an *and* gate to the negation of C_f^1. Denote the resulting circuits by $\mathrm{Sel}'_1, \dots, \mathrm{Sel}'_m$. Observe that the circuits Sel'_i output 0 whenever $C_f^2(\alpha) = f(\alpha)$ and otherwise output Sel_i. We think of these circuits as "selector circuits" which indicate whether on input $\alpha \in \{0,1\}^n$ (to the original variables $x_1, \dots, x_n$ over which the circuit is defined) the variable $y_i \in Y$ appears in the constraint for α.

The output of these selector circuits Sel'_i is connected to the gate y_i by an *and* gate. All these m *and* gates are in turn connect to a circuit computing the *xor* of these gates. Finally, to ensure that the circuit computes $f(\alpha)$ on inputs α such that $C_f(\alpha) \neq \perp$, we connect C_f^1 with C_f^2 by an *and* gate which is then connceted by a *or* gate to the output of the *xor* circuit. This completes the description of the restriction on the structure variables. A depiction of the resulting circuit can be found in Figure C.1.

Note that this implements the intended semantics: for each input $\alpha \in \{0,1\}^n$ the selector circuits output 1 on some variables y_i which are then *xor*'ed, and the restricted circuit outputs

$$\bigoplus_{i \in N(\alpha)} y_i \ , \tag{C.17}$$

unless $C_f(\alpha) \neq \perp$, in which case the output of the circuit is $f(\alpha)$ and all selector circuits output 0. We require that s is larger than the size of the described circuit which is of size $O\big(m \cdot \mathrm{poly}(n, k)\big) + s/2$.

We have the intended semantics of the circuit and need to ensure the furthermore property: that the restricted formula is over few variables. First, since the selector circuits Sel'_i are fixed, all evaluation variables for these subcircuits can be fixed to constants. The same holds for the circuit C_f. Similarly, since the y_i gate always carries the value of the y_i variable, all $2^n \cdot m$ wire variables corresponding to the Y variables can be substituted by the corresponding y_i variable and are thus restricted away.

After these restrictions the only evaluation variables left are those for the evaluation of the $\oplus$ circuit. For α such that $C_f(\alpha) \neq \perp$, the selector circuits are hard-wired to 0 and in particular the inputs to the $\oplus$ circuit is hard-wired to 0, meaning that these evalation variables can be restricted away.

There remains then only the $O(t \cdot m)$ evaluation variables corresponding to the evaluation of the $\oplus$ circuit for inputs α such that $C_f(\alpha) = \perp$. Let us, without loss of generality, use an *xor*-circuit which iteratively *xor*s each variable. Concretely, let it have subcircuits χ_i where $\chi_1 = \mathrm{Sel}'_1 \wedge y_1$ and $\chi_i = \chi_{i-1} \oplus (\mathrm{Sel}'_i \wedge y_i)$ for $i > 1$, and χ_m is the overall output of the $\oplus$ circuit.

The only observation required is that if the circuit $\mathrm{Sel}'_i(\alpha) = 0$, then χ_i gets a 0 as input from index i, independent of the value of y_i. Hence the output wire variable of the circuit χ_i indexed by the input α can be substituted by the output of the circuit χ_{i-1}. Hence for each α such that $C_f(\alpha) = \perp$, we can reduce the number of free wire variables indexed by α to $O(k)$, as each $\oplus$-constraint is over at most k variables. As C_f outputs $\perp$ on at most t inputs, we end up with a restriction leaving only a total of $O(t \cdot k + m)$ remaining variables in the restricted formula.

This completes the description of the restriction. As for a fixed input α at most k selector circuits output 1, we see that every variable $\mathrm{out}_\alpha(u)$ can be computed by a function over the appropriate k variables. Furthermore, each assignment to the remaining structure variables Y gives a valid circuit and this restriction is thus m-independent. □

We are ready to prove the degree lower bound, restated here for convenience.

Theorem C.1.1. *For all $\varepsilon > 0$ there is a $d = d(\varepsilon)$ such that the following holds. For $n \in \mathbb{N}$, all $s \geq n^d$ and any Boolean function $f : \{0,1\}^n \longrightarrow \{0,1\}$ on n bits, SoS requires degree $\Omega_\varepsilon(s^{1-\varepsilon})$ to refute* $\mathrm{Circuit}_s(f)$.

Proof. Let $G = (U, V, E)$ be an explicit bipartite graph as in Theorem C.2.2, with $U = \{0,1\}^n$, $k = O_\gamma\big((n \log r)^{1+1/\gamma}\big)$, and $|V| \leq k^2 r^{1+\gamma}$ for parameters $\gamma > 0$ and $r \leq 2^n$. Apply Lemma C.4.1 with $t = 2^n$ along with $C_f = \bot$ to obtain, for $s \geq m \cdot \mathrm{poly}(n,k)$, a natural m-independent k-determined affine restriction ρ for $\mathrm{Circuit}_s(f)$ such that $g^{\mathrm{out}}_{s,\alpha}(Y) = \oplus_{i \in N(\alpha)} y_i$. In words, the circuit of the restricted formula on input α computes an *xor* of the neighborhood of the vertex α of G.

Apply Lemma C.3.5 to ρ to conclude that if there is an SoS refutation of $\mathrm{Circuit}_s(f)\lceil_\rho$ of degree d, then there is a degree $O(d \cdot k)$ SoS refutation of the system of polynomial equations computing

$$\mathcal{P}_G = \Big\{ \bigoplus_{i \in N(\alpha)} y_i = f(\alpha) : \alpha \in \{0,1\}^n \Big\} \cup \\ \{y_i^2 = y_i, \bar{y}_i^2 = \bar{y}_i, y_i = 1 - \bar{y}_i \mid i \in [m]\} \ . \tag{C.18}$$

As the graph G is a strong expander, we can apply Theorem C.2.5 to get an SoS degree lower bound of $\Omega(r)$ for the XOR-CSP instance $\mathcal{P}_G$ defined over G, which in turn gives us an $\Omega(r/k)$ degree lower bound for the $\mathrm{Circuit}_s(f)\lceil_\rho$ formula and hence also for the unrestricted formula.

Let us fix the parameters. We want to choose r as large as possible. However, the larger we choose r, the larger m may become, since Theorem C.2.2 only guarantees that $m \leq k^2 r^{1+\gamma}$. Let us analyze how large r can be chosen in terms of n and s.

Note that $k = \mathrm{poly}_\gamma(n)$, where we use that $r \leq 2^n$, and we write $\mathrm{poly}_\gamma(n)$ to denote some polynomial in n whose degree and coefficients may depend on γ. Hence we may choose

$$m = \frac{s}{\mathrm{poly}_\gamma(n)} \ , \tag{C.19}$$

according to the requirement on s in Lemma C.4.1. From the guarantees of Theorem C.2.2 we know that $r \geq (m/k^2)^{1/(1+\gamma)}$. Substituting m according to the previous equation we get that

$$r \geq \left(\frac{s}{k^2 \mathrm{poly}_\gamma(n)}\right)^{\frac{1}{1+\gamma}} = \frac{s^{1/(1+\gamma)}}{\mathrm{poly}_\gamma(n)} \ . \tag{C.20}$$

Hence if we choose γ small enough so that $\frac{1}{1+\gamma} > 1 - \varepsilon/2$ and then require s to be large enough such that the final $\mathrm{poly}_\gamma(n)$ is at most $s^{\varepsilon/2}$, we obtain the claimed lower bound. □

In the following we prove the claimed size lower bound.

Theorem C.1.3. *For all* $\varepsilon > 0$ *there is a* $d = d(\varepsilon)$ *such that the following holds. Let* $n \in \mathbb{N}$ *and* $s \in \mathbb{N}$ *such that* $s \geq n^d$. *If* $t \geq s$ *and* $f \in \mathcal{F}_n(s/2, t)$, *then it holds that SoS requires size* $\exp\left(\Omega_\varepsilon(s^{2-\varepsilon}/t)\right)$ *to refute* $\mathrm{Circuit}_s(f)$.

Proof. Apply Lemma C.4.1 with the graphs from Theorem C.2.2 as in the proof of Theorem C.1.1. We get a natural m-independent k-determined affine restriction ρ and the formula $\mathrm{Circuit}_s(f)\lceil_\rho$ over $O(t \cdot k + m)$ variables. To this formula we then apply Lemma C.3.5 to obtain a degree lower bound of $\Omega(r/k)$, akin to the proof of Theorem C.1.1. By setting the parameters as in the aforementioned proof we get the same degree lower bound of $\Omega_\varepsilon(s^{1-\varepsilon})$ for the formula $\mathrm{Circuit}_s(f)\lceil_\rho$. As this formula is over few variables we can apply Theorem C.2.6 to obtain an SoS size lower bound of $\exp\left(\Omega_\varepsilon\left((s^{1-\varepsilon} - 3k)^2/(t \cdot k + m)\right)\right)$ for the restricted formula. As affine restrictions may only decrease the size of a refutation, the same lower bound also holds for the unrestricted formula. We obtain the desired lower bound by choosing s large enough such that $s^\varepsilon \geq k = \mathrm{poly}_\varepsilon(n)$ and by recalling that $t \geq s \geq m$. □

C.5 Lower Bounds for Monotone Circuits

Recall that $\mathcal{M}_n(\ell)$ denotes all Boolean monotone ℓ-slice functions on n bits: all Boolean functions $f : \{0,1\}^n \to \{0,1\}$ that output 0 on all inputs of Hamming weight less than ℓ and 1 on all inputs of Hamming weight larger than ℓ. There is no restriction on the output for inputs of Hamming weight ℓ and we hence have that $|\mathcal{M}_n(\ell)| = 2^{\binom{n}{\ell}}$. Further, recall that $\mathcal{M}_n(\ell, s, t) \subseteq \mathcal{M}_n(\ell)$ is the class of monotone Boolean ℓ-slice functions $f : \{0,1\}^n \to \{0,1\}$ for which there is a monotone Boolean circuit $C_f^{\mathrm{mon}} : \{0,1\}^n \to \{0,1,\perp\}$ such that for ℓ-slice inputs $\alpha \in \binom{[n]}{\ell}$ it holds that

1. if $C_f^{\mathrm{mon}}(\alpha) \neq \perp$, then $C_f^{\mathrm{mon}}(\alpha) = f(\alpha)$, and
2. $C_f^{\mathrm{mon}}(\alpha) = \perp$ on at most t inputs.

It is very convenient to work with slice functions as we have a handle on their monotone circuit complexity: by Lemma C.2.4 their monotone circuit size is the same as their ordinary circuit size up to a polynomial size increase. Hence we do not need to worry whether the functions needed for the reduction have small monotone circuits, as long as we are working on a slice only.

The proof of the monotone lower bound is an adaption of the argument used to prove Lemma C.4.1. The idea is to work over the ℓth slice and disregard all other inputs. By Lemma C.2.4 we can implement our selector circuits by small monotone circuits. We then also need to take care of the negations in the $\oplus$-circuit. We push the negations down until they either hit a gate in Y or a selector circuit. We create a set $\overline{Y}$ gates, which we can think of as the negation of the gates in Y and also create negated selector circuits (on the ℓth slice). By doing so we can now get rid of the last negations by appropriately connecting the appropriate circuits. The following corollary of Lemma C.2.4 will be useful to us.

Claim C.5.1. *Let C be a Boolean circuit on n input bits of size s. Then, for $\ell \in [n]$, there is a monotone Boolean circuit C^{mon} of size $2s + \mathrm{poly}(n)$ computing the ℓ-slice function that is equal to C on the ℓ-slice.*

Proof. Let $\mathcal{T}_{\geq \ell}$ be the threshold function that outputs 1 if and only if the Hamming weight of an input $\alpha \in \{0,1\}^n$ is at least ℓ. Connect the output of C by an *and* gate to a circuit computing $\mathcal{T}_{\geq \ell}$. The output of this circuit is then connected by an *or* gate to the output of a circuit computing $\mathcal{T}_{> \ell}$. Let us denote this new circuit by C'.

The circuit C' clearly outputs 1 whenever the input is of Hamming weight larger than ℓ. Furthermore, on the ℓ-slice it is equal to C because $\mathcal{T}_{\geq \ell}$ outputs 1 while $\mathcal{T}_{> \ell}$ outputs 0. Finally the output is 0 if the Hamming weight is less than ℓ because the output of both threshold functions is 0.

Clearly the size of the circuits computing the threshold functions is $\mathrm{poly}(n)$. We apply Lemma C.2.4 to conclude that there is a monotone circuit C^{mon} computing the same function as C' of size $2s + \mathrm{poly}(n)$. □

Before stating the following lemma we need to adapt some terminology to the monotone setting. Observe that $\mathrm{Circuit}_s^{\mathrm{mon}}(f)$ is a restriction of $\mathrm{Circuit}_s(f)$. Let τ be such that $\mathrm{Circuit}_s(f)\lceil_\tau = \mathrm{Circuit}_s^{\mathrm{mon}}(f)$. This allows us to naturally extend k-determined restrictions to the monotone setting: a restriction ρ is a k-determined restriction for $\mathrm{Circuit}_s^{\mathrm{mon}}(f)$ if the restriction $\rho\tau$ is a k-determined restriction for $\mathrm{Circuit}_s(f)$. Similarly we can extend m-independence to the monotone setting. This will later allow us to use Lemma C.3.5 even though we are working with the monotone formula.

Lemma C.5.2. *For all $k, \ell, m, n, t \in \mathbb{N}$ satisfying $m \leq 2^n$, and any explicit bipartite graph $G = (U, V, E)$ such that $|U| = 2^n$, $|V| = m$ and all $u \in U$ are of degree $\deg(u) \leq k$, the following holds. There is a constant $C > 0$, depending on the explicitness of G, such that for all $s \geq C \cdot m \cdot n^C \cdot k^C$ and any $f \in \mathcal{M}_n(\ell, s/10, t)$ there is a natural m-independent k-determined affine restriction ρ for the formula*

$\mathrm{Circuit}^{\mathrm{mon}}_{s}(f)$ *such that*

$$g^{\mathrm{out}}_{s,\alpha}(Y) = \begin{cases} 1, & \text{if } |\alpha| > \ell, \\ 0, & \text{if } |\alpha| < \ell, \\ f(\alpha), & \text{if } |\alpha| = \ell \text{ and } C^{\mathrm{mon}}_{f}(\alpha) \neq \bot, \\ \oplus_{i \in N(\alpha)} y_i, & \text{otherwise,} \end{cases}$$

for $g^{\mathrm{out}}_{s,\alpha}$ *and* Y *as in Definition C.3.2.*

Furthermore, the formula $\mathrm{Circuit}^{\mathrm{mon}}_{s}(f)\lceil_{\rho}$ *is over* $O(t \cdot k + m)$ *variables.*

Proof. This proof is an adaptation of the argument of the proof Lemma C.4.1. Let us describe the natural m-independent k-determined restriction ρ for the formula $\mathrm{Circuit}^{\mathrm{mon}}_{s}(f)$.

As in the proof of Lemma C.4.1 we have gates that act as Boolean variables. But instead of having a single set Y of variables we now have two sets Y and $\overline{Y}$, each of size m. We think of the variables in $\overline{Y}$ as the negations of the variables in Y and ensure this by applying the appropriate affine restriction for all $\alpha \in \{0,1\}^n$ and $i \in [m]$.

According to Claim C.5.1 we may assume that the circuit C^{mon}_{f} computes a monotone ℓ-slice function in both outputs $C^{\mathrm{mon}}_{f,1}, C^{\mathrm{mon}}_{f,2}$ for a mild increase in size; $|C^{\mathrm{mon}}_{f}| \le s/5 + \mathrm{poly}(n) \le s/4$ for s large enough. Recall that the first output of C^{mon}_{f} indicates whether the second output bit is equal to f on the ℓ-slice. Let $\overline{C}^{\mathrm{mon}}_{f,1}$ be the negation of $C^{\mathrm{mon}}_{f,1}$ on the ℓ-slice. In other words, $\overline{C}^{\mathrm{mon}}_{f,1}(\alpha) = \neg C^{\mathrm{mon}}_{f,1}(\alpha)$ if α has Hamming weight ℓ, and $\overline{C}^{\mathrm{mon}}_{f,1}(\alpha) = C^{\mathrm{mon}}_{f,1}(\alpha)$ otherwise.

The monotone circuit C^{mon}_{f} is of size at most $s/4$ and hence according to Lemma C.2.4 there is a monotone circuit of size $s/2 + \mathrm{poly}(n) \le 5s/8$ computing $\overline{C}^{\mathrm{mon}}_{f,1}(\alpha)$.

We restrict the formula such that a part of the circuit is equivalent to C^{mon}_{f} and another part is equal to $\overline{C}^{\mathrm{mon}}_{f,1}$. Note that the size of these two circuits is at most $7s/8$ by above discussion.

Recall that because $G(\{0,1\}^n, Y, E)$ is explicit, there are circuits $\mathrm{Sel}_1, \mathrm{Sel}_2, \ldots, \mathrm{Sel}_m$, each of size $\mathrm{poly}(n,k)$, where each Sel_i computes, given an input $\alpha \in \{0,1\}^n$, whether the vertex $y_i \in Y$ is a neighbor of the vertex α. Let $\overline{\mathrm{Sel}}_i = \neg\, \mathrm{Sel}_i$ and denote by $\mathrm{Sel}^{\mathrm{mon}}_{i}$ (respectively $\overline{\mathrm{Sel}}^{\mathrm{mon}}_{i}$) the circuit obtained by applying Claim C.5.1 to Sel_i (to $\overline{\mathrm{Sel}}_i$ respectively). By the guarantees of Claim C.5.1 all these $2m$ circuits are of size $\mathrm{poly}(n,k)$.

We restrict the formula such that a part of the circuit computes the functions

$$\mathrm{Sel}^{\mathrm{mon}}_{1}, \ldots, \mathrm{Sel}^{\mathrm{mon}}_{m}, \overline{\mathrm{Sel}}^{\mathrm{mon}}_{1}, \ldots, \overline{\mathrm{Sel}}^{\mathrm{mon}}_{m} \;. \tag{C.21}$$

From these ℓ-slice selector circuits we can then define selector circuits that take C_f^{mon} into account. Namely, we connect $\mathrm{Sel}_i^{\mathrm{mon}}$ by an *and* gate to the output of $\overline{C}_{f,1}^{\mathrm{mon}}$ to obtain the circuit $\mathrm{Sel}_i'^{\mathrm{mon}}$ and similarly connect $\overline{\mathrm{Sel}}_i^{\mathrm{mon}}$ by an *or* gate to $C_{f,1}^{\mathrm{mon}}$ to obtain the circuit $\overline{\mathrm{Sel}}_i'^{\mathrm{mon}}$.

Finally, we also put each variable y_i and $\bar{y}_i$ onto the slice by the same construction used in the proof of Claim C.5.1: connect the variable y_i (respectively $\bar{y}_i$) by an *and* to the threshold circuit $\mathcal{T}_{\geq \ell}$ and connect this circuit in turn by an *or* gate to a $\mathcal{T}_{>\ell}$ threshold circuit to obtain y_i^{mon} (respectively $\bar{y}_i^{\mathrm{mon}}$). It is well-known [Val84; BW06; Gol20] that threshold circuits have montone circuits of size $\mathrm{poly}(n)$ and we can thus restrict the formula such that a part of the circuit computes y_i^{mon} and $\bar{y}_i^{\mathrm{mon}}$.

Finally we connect y_i^{mon} by an *and* gate to the selector circuit $\mathrm{Sel}_i'^{\mathrm{mon}}$. Note that this circuit is equal to an ℓ-slice function. As we will see later this ensures that the whole circuit outputs an ℓ-slice function. We connect the circuits $\bar{y}_i^{\mathrm{mon}}$ similarly: connect $\bar{y}_i^{\mathrm{mon}}$ by an *or* gate to the negated selector circuit $\overline{\mathrm{Sel}}_i'^{\mathrm{mon}}$. Again, the output of this circuit is equal to an ℓ-slice function.

Equally inportant is that these circuits behave well on the ℓ-slice. Indeed it can be checked that the positive circuit, on input $\alpha \in \{0,1\}^n$, outputs $\mathrm{Sel}_i'^{\mathrm{mon}}(\alpha) \wedge y_i$ while the negative circuit outputs $\overline{\mathrm{Sel}}_i'^{\mathrm{mon}}(\alpha) \vee \bar{y}_i$. On the ℓ-slice these functions are the negation of eachother, which we are going to use in the following.

We need to construct a monotone circuit for the *xor* of $\mathrm{Sel}_i'^{\mathrm{mon}}(\alpha) \wedge y_i$ for i from 1 to m, on ℓ-slice inputs α. We take a standard $O(m)$-size $\oplus$-circuit and monotonize it by pushing all negations in it down using De Morgan's law until they reach one of the $\oplus$-circuit's inputs $\mathrm{Sel}_i'^{\mathrm{mon}} \wedge y_i$. Whenever the negation of $\mathrm{Sel}_i'^{\mathrm{mon}}(\alpha) \wedge y_i$ is needed, we do one last step of De Morgan and replace it by $\overline{\mathrm{Sel}}_i'^{\mathrm{mon}}(\alpha) \vee \bar{y}_i$.

To ensure that the circuit outputs $f(\alpha)$ whenever $C_f^{\mathrm{mon}}(\alpha) \neq \perp$, we connect the two outputs of C_f^{mon} by an *and* gate and connect this gate by an *or* gate to the output of the *xor* circuit. This completes the description of the restriction on the structure variables. A depiction of the resulting circuit can be found in Figure C.2. We ensure that s is large enough so that above circuit can be described by the formula.

Note that the constructed circuit always outputs a monotone ℓ-slice function: as the monotonized $\oplus$-circuit is non-constant, we see that if all inputs to the circuit are 0, it outputs 0 and if all inputs are 1, it outputs 1. This, in particular, implies that the circuit outputs 0 (respectively 1) if the input is below (respectively, above) the ℓ-slice and hence the entire circuit computes a monotone ℓ-slice function.

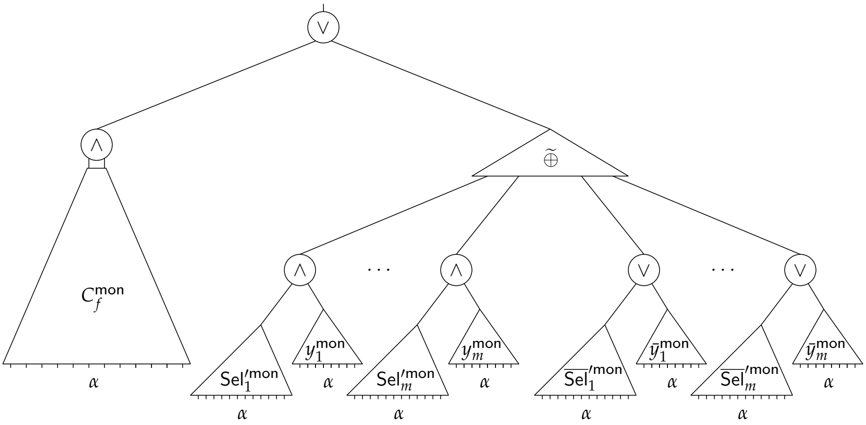


Figure C.2: A depiction of the monotone circuit, where $\widetilde{\oplus}$ is the $\oplus$ circuit with the negations pushed down.

It can be easily checked that the described restriction is m-independent and k-determined. In order to prove the furthermore part, we need to reduce the number of evaluation variables. This can be achieved analogous to the proof of Lemma C.4.1 and we thus omit it here. □

Let us prove our degree lower bound for monotone circuits, restated here for convenience.

Theorem C.1.4. *For all $\varepsilon > 0$ there is a $d = d(\varepsilon)$ such that the following holds. For all $n, \ell \in \mathbb{N}$, all $s \geq n^d$ and any monotone slice function $f \in \mathcal{M}_n(\ell)$ SoS requires degree $\Omega_\varepsilon(s^{1-\varepsilon})$ to refute* $\mathrm{Circuit}_s^{\mathrm{mon}}(f)$.

Proof of Theorem C.1.4. As in the proof of Theorem C.1.1, we use the graphs from Theorem C.2.2, with $U = \{0,1\}^n$, $k = O_\gamma\big((n \log r)^{1+1/\gamma}\big)$, and $|V| \leq k^2 r^{1+\gamma}$ for parameters $\gamma > 0$ and $r \leq 2^n$. We apply Lemma C.5.2 with above graph and $t = 2^n$ along with $C_f^{\mathrm{mon}} = \perp$ to obtain, for $s \geq m \cdot \mathrm{poly}(n,k)$, an appropriate natural m-independent k-determined affine restriction ρ for $\mathrm{Circuit}_s^{\mathrm{mon}}(f)$. In particular ρ satisfies

$$g_{s,\alpha}^{\mathrm{out}}(Y) = \begin{cases} 1, & \text{if } |\alpha| > \ell, \\ 0, & \text{if } |\alpha| < \ell, \\ \oplus_{i \in N(\alpha)} y_i, & \text{otherwise,} \end{cases}$$

for $g_{s,\alpha}^{\mathrm{out}}$ and Y as in definition Definition C.3.2.

Recall that there is a restriction τ such that $\mathrm{Circuit}_s^{\mathrm{mon}}(f) = \mathrm{Circuit}_s(f)\lceil_\tau$ and we can thus apply Lemma C.3.5 with $\tau\rho$ to conclude that if there is an

SoS refutation of $\mathrm{Circuit}_s^{\mathrm{mon}}(f)\lceil_\rho$ in degree d, then there is a degree $d \cdot k$ SoS refutation of the system of polynomial equations computing

$$\{ \bigoplus_{i \in N(\alpha)} y_i = f(\alpha) \mid \alpha \in \binom{[n]}{\ell} \} \ . \tag{C.22}$$

As the graph G is a strong expander, we can apply Theorem C.2.5 to get an SoS degree lower bound of $\Omega(r)$ for above system of equations. By above connection this gives an $\Omega(r/k)$ degree lower bound for the $\mathrm{Circuit}_s^{\mathrm{mon}}(f)\lceil_\rho$ formula and hence also for the unrestricted formula.

Regarding the parameters, as in the proof of Theorem C.1.1 we choose $m = s/\mathrm{poly}_\gamma(n)$. Repeating the calculations from the aforementioned proof we obtain that $r \geq s^{1/(1+\gamma)}/\mathrm{poly}_\gamma(n)$. Thus by choosing γ small enough such that $\frac{1}{1+\gamma} > 1 - \varepsilon/2$ and s large enough such that the final $\mathrm{poly}_\gamma(n) \leq s^{\varepsilon/2}$ we obtain the claimed degree lower bound of $\Omega_\varepsilon(s^{1-\varepsilon})$. □

As in the non-monotone case, we can also obtain size lower bounds for functions that almost have a circuit of size s.

Theorem C.1.5. *For all $\varepsilon > 0$ there is a $d = d(\varepsilon)$ such that the following holds. For $n, \ell \in \mathbb{N}$, all $s \geq n^d$ and $t \geq s$ and monotone function $f \in \mathcal{M}_n(\ell, s/10, t)$ SoS requires size $\exp\left(\Omega_\varepsilon(s^{2-\varepsilon}/t)\right)$ to refute $\mathrm{Circuit}_s^{\mathrm{mon}}(f)$.*

Proof. Analogous to the proof of Theorem C.1.3. □

C.6 Degree Upper Bound

We are given the formula $\mathrm{Circuit}_s(f)$, for a truthtable f that has no circuit of size s. In the following we describe an SoS refutation of $\mathrm{Circuit}_s(f)$ in degree $O(s)$. This shows that our degree lower bounds are essentially tight.

Let us first define a set of monomials, which essentially correspond to circuits of size s. A multilinear monomial m is a *circuit monomial* if for every gate $v \in [s]$ it holds that

1. one of the variables $\mathrm{isNeg}(v), \mathrm{isOr}(v)$ or $\mathrm{isAnd}(v)$ occurs in m,
2. for $a \in \{1, 2\}$ one of the variables $\mathrm{isFromConst}(v, a), \mathrm{isFromInput}(v, a)$ or $\mathrm{isFromGate}(v, a)$ occurs in m,
3. again for $a \in \{1, 2\}$ one (unless empty) variable of each of the two sets of variables

$$\{\mathrm{isInput}(v, a, i) \mid i \in [n]\} \ , \text{ and } \quad \{\mathrm{isGate}(v, a, u) \mid u < v\} \tag{C.23}$$

occurs in m, and

4. no other variables occur in m than the ones described above.

We denote by $\mathcal{M}_s$ the set of circuit monomials. We first show that SoS can derive in degree $O(s)$ the polynomial $\sum_{m \in \mathcal{M}_s} m - 1$, and then in a second step that it can also derive $-\sum_{m \in \mathcal{M}_s} m$ in degree $O(s)$. The sum of these two derivations is clearly an SoS derivation of -1; a refutation of the $\mathrm{Circuit}_s(f)$ formula.

Deriving $\sum_{m \in \mathcal{M}_s} m - 1$. We proceed by induction on s. Note that $\mathcal{M}_0 = \{1\}$ and hence the base case is trivial. Suppose we have an SoS derivation of $\sum_{m \in \mathcal{M}_s} m - 1$. For every monomial $m \in \mathcal{M}_s$ we add the polynomial

$$m \cdot \big(\mathrm{isNeg}(v) + \mathrm{isOr}(v) + \mathrm{isAnd}(v) - 1\big) \tag{C.24}$$

to the derivation (note that the second term is Axiom C.4). This gives us an SoS derivation of $\sum_{m \in \mathcal{M}'_s} m - 1$, where

$$\mathcal{M}'_s = \big\{ m \times \{\mathrm{isNeg}(v), \mathrm{isOr}(v), \mathrm{isAnd}(v)\} \mid m \in \mathcal{M}_s \big\} \ .$$

We can continue in the same manner with Axiom C.3 and Axioms C.5 to finally obtain an SoS derivation of $\sum_{m \in \mathcal{M}_{s+1}} m - 1$. Clearly this derivation requires degree at most $O(s)$, as for each gate there are at most 7 variables in every monomial from $\mathcal{M}_s$.

Deriving $-\sum_{m \in \mathcal{M}_s} m$. Fix a monomial $m \in \mathcal{M}_s$. We describe a degree $O(s)$ SoS derivation of $-m$. Let C be the circuit that corresponds to the monomial m and let $\alpha \in \{0,1\}^n$ be such that $f(\alpha) \neq C(\alpha)$. Let us assume that $f(\alpha) = 0$ but $C(\alpha) = 1$.

We construct a degree $O(s)$ SoS proof of the fact that $C(\alpha) = 1$. That is, we are going to conctruct a polynomial p_s which can be written as

$$p_s = m \cdot \left(\sum_{i=1}^{m} r_i \cdot q_i \right) \ , \tag{C.25}$$

for axioms q_i and some polynomials r_i, such that p_s simplifies to the polynomial $m \cdot (\mathrm{out}_\alpha(s) - 1)$. We construct this proof by structural induction over the circuit: for every gate v we are going to construct an SoS proof p_v of the fact that the circuit rooted at v outputs the bit b_v on input α. In other words, p_v simplifies to the polynomial $m \cdot (\mathrm{out}_\alpha(v) - b_v)$.

Let us explain how to construct an SoS proof p_v. Consider a gate v in the circuit. Depending on the function computed at v and how the wires of v are connected we construct p_v slightly different. As a first step let us construct SoS proofs q_1 and q_2 of the fact that on input α the bit b_a, $a \in \{1, 2\}$, is carried on wire a to the gate v. That is, the polynomial q_a should simplify to $m \cdot \big(\mathrm{in}_\alpha(v, a) - b_a\big)$. In the following we explain how to precisely define q_a depending on what the wire is connected to. Note that not a lot is going on – we are mostly just multilinearizing using the Boolean axioms.

If the wire a is connected to a constant and $m = m_1 \cdot \mathrm{isFromConst}(v, a)$, then the polynomial

$$\begin{aligned} q_a = m \cdot \mathrm{isFromConst}(v,a) \cdot \big(\mathrm{in}_\alpha(v,a) - \mathrm{constantValue}(v,a)\big) + \\ m_1 \cdot \big(\mathrm{in}_\alpha(v,a) - \mathrm{constantValue}(v,a)\big) \cdot \\ \big(\mathrm{isFromConst}(v,a) - \mathrm{isFromConst}(v,a)^2\big) \end{aligned} \tag{C.26}$$

is a valid SoS proof that simplifies to the polynomial $m \cdot \big(\mathrm{in}_\alpha(v, a) - \mathrm{constantValue}(v, a)\big)$. Similarly, if a is connected to an input i and we let $m = m_2 \cdot \mathrm{isFromInput}(v, a) \cdot \mathrm{isInput}(v, a, i)$, then we can choose

$$\begin{aligned} q_a = m \cdot \mathrm{isFromInput}(v,a) \cdot \mathrm{isInput}(v,a,i) \cdot \big(\mathrm{in}_\alpha(v,a) - \alpha_i\big) \\ + m_2 \cdot \mathrm{isFromInput}(v,a)^2 \cdot \big(\mathrm{in}_\alpha(v,a) - \alpha_i\big) \cdot \\ \big(\mathrm{isInput}(v,a,i) - \mathrm{isInput}(v,a,i)^2\big) \\ + m_2 \cdot \mathrm{isInput}(v,a,i) \cdot \big(\mathrm{in}_\alpha(v,a) - \alpha_i\big) \cdot \\ \big(\mathrm{isFromInput}(v,a) - \mathrm{isFromInput}(v,a)^2\big) \ , \end{aligned} \tag{C.27}$$

which simplifies to $m \cdot \big(\mathrm{in}_\alpha(v, a) - \alpha_i\big)$. And similarly, if a is connected to a gate u and we let $m = m_3 \cdot \mathrm{isFromGate}(v, a) \cdot \mathrm{isGate}(v, a, u)$, then

$$\begin{aligned} q_a = m \cdot \mathrm{isFromGate}(v,a) \cdot \mathrm{isGate}(v,a,u) \cdot \big(\mathrm{in}_\alpha(v,a) - \mathrm{out}_\alpha(u)\big) \\ + m_3 \cdot \mathrm{isFromGate}(v,a)^2 \cdot \big(\mathrm{in}_\alpha(v,a) - \mathrm{out}_\alpha(u)\big) \cdot \\ \big(\mathrm{isGate}(v,a,u) - \mathrm{isGate}(v,a,u)^2\big) \\ + m_3 \cdot \mathrm{isGate}(v,a,u) \cdot \big(\mathrm{in}_\alpha(v,a) - \mathrm{out}_\alpha(u)\big) \cdot \\ \big(\mathrm{isFromGate}(v,a) - \mathrm{isFromGate}(v,a)^2\big) \\ + p_u \ , \end{aligned} \tag{C.28}$$

where, by induction, we assume that p_u has already been derived. Note that this polynomial simplifies to $m \cdot \big(\mathrm{in}_\alpha(v, a) - b_u\big)$, which is what we would expect.

Given the two SoS proofs q_1 and q_2 we are ready to construct the SoS proof p_v. As mentioned earlier we do a case distinction over the funtion computed at v. For the sake of readibility we implicitly multilinearize. In other words, below polynomials are correct SoS derivations once they are reduced by the Boolean axioms.

1. v *is a* not *gate.* We choose

$$\begin{aligned} p_v =& m \cdot \mathrm{isNeg}(v) \cdot \left(\mathrm{out}_\alpha(v) - \overline{\mathrm{in}_\alpha(v,1)}\right) \\ &+ m \cdot \left(\overline{\mathrm{in}_\alpha(v,1)} - 1 + \mathrm{in}_\alpha(v,1)\right) \\ &- q_1 \quad , \end{aligned} \tag{C.29}$$

where the first line is Axiom C.12 multiplied by m and the second line is the axiom relating $\mathrm{in}_\alpha(v,1)$ with $\overline{\mathrm{in}_\alpha(v,1)}$, again multiplied by m. The SoS proof p_v simplifies to $m \cdot \left(\mathrm{out}_\alpha(v) - (1 - b_1)\right)$, as one would expected.

2. v *is an* or *gate.* Let

$$\begin{aligned} p_v =& m \cdot \mathrm{isOr}(v) \cdot \left(\mathrm{out}_\alpha(v) - \left(1 - \overline{\mathrm{in}_\alpha(v,1)} \cdot \overline{\mathrm{in}_\alpha(v,2)}\right)\right) \\ &- m \cdot \overline{\mathrm{in}_\alpha(v,1)} \cdot \left(\overline{\mathrm{in}_\alpha(v,2)} - 1 + \mathrm{in}_\alpha(v,2)\right) \\ &+ m \cdot \left(\mathrm{in}_\alpha(v,2) - 1\right) \cdot \left(\overline{\mathrm{in}_\alpha(v,1)} - 1 + \mathrm{in}_\alpha(v,1)\right) \\ &+ \left(1 - \mathrm{in}_\alpha(v,1)\right) \cdot q_2 \\ &+ \left(1 - b_2\right) \cdot q_1 \quad , \end{aligned} \tag{C.30}$$

where the first line is Axiom C.13 multiplied by m and the following two lines are the axioms relating the non-bar variable with the corresponding bar variable appropriately multiplied by a polynomial. By inspection it is not hard to see that p_v simplifies to $m \cdot \left(\mathrm{out}_\alpha(v) + b_1 b_2 - b_1 - b_2\right)$. Note that this polynomial has the intended semantics: $b_v = 0$ if and only if both b_1 and b_2 are 0; otherwise $b_v = 1$.

3. v *is an* and *gate.* We have

$$\begin{aligned} p_v =& m \cdot \mathrm{isAnd}(v) \cdot \left(\mathrm{out}_\alpha(v) - \mathrm{in}_\alpha(v,1) \cdot \mathrm{in}_\alpha(v,2)\right) \\ &+ \mathrm{in}_\alpha(v,1) \cdot q_2 \\ &+ b_2 \cdot q_1 \quad , \end{aligned} \tag{C.31}$$

where the first line is Axiom C.14 multiplied by m. The polynomial p_v simplifies to $m \cdot \left(\mathrm{out}_\alpha(v) - b_1 b_2\right)$.

This completes the description of the SoS derivation of $\mathrm{out}_\alpha(s) = 1$. Observe that the final proof p_s is of degree $O(s)$: in every inductive step we increase the degree of the proof by at most a constant.

So far we only have an SoS proof of $m \cdot \left(\mathrm{out}_\alpha(s) - 1\right)$. What we really want, though, is a derivation of the term $-m$. But that is simple to derive: from the SoS derivation of $m \cdot \big(\mathrm{out}_\alpha(s) - 1\big)$ we simply subtract the polynomial $m \cdot \big(\mathrm{out}_\alpha(s) - f(\alpha)\big)$ (which is Axiom C.15 multiplied by m) from p_s. This completes the SoS proof of $-m$.

C.7 On Encodings of the $\mathrm{Circuit}_s(f)$ Tautology

In this section we show that if there is an SoS refutation of degree d of the $\mathrm{Circuit}_s(f)$ formula encoded as a constant width CNF, then there is an SoS refutation of degree $O(d)$ of the $\mathrm{Circuit}_s(f)$ formula encoded as in Section C.2.3. Let us first discuss a possible constant width CNF encoding of $\mathrm{Circuit}_s(f)$.

The formula is defined over the same variables as introduced in Section C.2.3, but in order to keep the fan-in bounded, we further introduce variables $\mathrm{isInput}^{\leq}(v, a, i)$ and $\mathrm{isGate}^{\leq}(v, a, u)$ that indicate whether the wire a of v is connected to a variable in $x_1, \ldots, x_i$, a gate $1, \ldots, u$ respectively.

Let us group the axioms in the same manner as we did in Section C.2.3. First we have the structure axioms. The first axioms encode that each wire is connected to a single kind

$$\begin{aligned}&\big(\mathrm{isFromConst}(v, a) \lor \mathrm{isFromInput}(v, a) \lor \mathrm{isFromGate}(v, a)\big)\land\\ &\neg\big(\mathrm{isFromConst}(v, a) \land \mathrm{isFromInput}(v, a)\big)\land\\ &\neg\big(\mathrm{isFromInput}(v, a) \land \mathrm{isFromGate}(v, a)\big)\land\\ &\neg\big(\mathrm{isFromConst}(v, a) \land \mathrm{isFromGate}(v, a)\big)\ .\end{aligned} \tag{C.32}$$

The next set of axioms similarly ensure that each gate computes precisely one function

$$\begin{aligned}&\big(\mathrm{isNeg}(v) \lor \mathrm{isOr}(v) \lor \mathrm{isAnd}(v)\big)\land\\ &\neg\big(\mathrm{isNeg}(v) \land \mathrm{isOr}(v)\big) \land \neg\big(\mathrm{isOr}(v) \land \mathrm{isAnd}(v)\big) \land \neg\big(\mathrm{isNeg}(v) \land \mathrm{isAnd}(v)\big)\ .\end{aligned} \tag{C.33}$$

Last, we need to make sure that each wire is connected to a single input or a gate.

$$\begin{aligned}&\mathrm{isInput}^{\leq}(v, a, n) \land \bigwedge_{i \neq j} \neg\big(\mathrm{isInput}(v, a, i) \land \mathrm{isInput}(v, a, j)\big)\land\\ &\bigwedge_{i \in [n]} \Big(\mathrm{isInput}^{\leq}(v, a, i) \equiv \big(\mathrm{isInput}^{\leq}(v, a, i-1) \lor \mathrm{isInput}(v, a, i)\big)\Big)\ ,\\ &\qquad \text{where } \mathrm{isInput}^{\leq}(v, a, 0) \overset{\mathrm{def}}{=} 0\ ,\end{aligned} \tag{C.34}$$

and similarly for $v \in [s] \setminus \{1\}$ we have that

$$\begin{aligned}&\mathrm{isGate}^{\leq}(v, a, v-1) \wedge \bigwedge_{u<u'<v} \neg\big(\mathrm{isGate}(v, a, u) \wedge \mathrm{isGate}(v, a, u')\big) \wedge \\ &\bigwedge_{u \in [v-1]} \Big(\mathrm{isGate}^{\leq}(v, a, u) \equiv \big(\mathrm{isGate}^{\leq}(v, a, u-1) \vee \mathrm{isGate}(v, a, u)\big)\Big) \ , \\ &\qquad \text{where } \mathrm{isGate}^{\leq}(v, a, 0) \overset{\mathrm{def}}{=} 0 \ . \end{aligned} \tag{C.35}$$

Let us take a look at the evaluation axioms. Again, we have axioms that ensure that the wires carry the values intended by the structure variables. If a wire is connected to a constant, then the evaluation variable associated with that wire should be equal to the constant

$$\mathrm{isFromConst}(v, a) \rightarrow \big(\mathrm{in}_\alpha(v, a) \equiv \mathrm{constantValue}(v, a)\big) \ , \tag{C.36}$$

and similarly if a wire is connected to an input or a gate

$$\mathrm{isFromInput}(v, a) \wedge \mathrm{isInput}(v, a, i) \rightarrow \mathrm{in}_\alpha(v, a) \equiv \alpha_i \ , \tag{C.37}$$

$$\mathrm{isFromGate}(v, a) \wedge \mathrm{isGate}(v, a, u) \rightarrow \mathrm{in}_\alpha(v, a) \equiv \mathrm{out}_\alpha(u) \ . \tag{C.38}$$

Last we need to make sure that the gates propagate the value they are supposed to compute.

$$\mathrm{isNeg}(v) \rightarrow \big(\mathrm{out}_\alpha(v) \equiv \neg \mathrm{in}_\alpha(v, 1)\big) \tag{C.39}$$

$$\mathrm{isOr}(v) \rightarrow \big(\mathrm{out}_\alpha(v) \equiv \mathrm{in}_\alpha(v, 1) \vee \mathrm{in}_\alpha(v, 2)\big) \tag{C.40}$$

$$\mathrm{isAnd}(v) \rightarrow \big(\mathrm{out}_\alpha(v) \equiv \mathrm{in}_\alpha(v, 1) \wedge \mathrm{in}_\alpha(v, 2)\big) \ . \tag{C.41}$$

The final axioms ensure that the correct function is computed

$$\mathrm{out}_\alpha(s) \equiv f(\alpha) \ . \tag{C.42}$$

This formula can be rewritten in the usual manner into a 4-CNF. Let us denote this formula by $\mathrm{Circuit}_s^{\mathrm{CNF}}(f)$. Observe that for each axiom p from the polynomial encoding $\mathrm{Circuit}_s(f)$, there is a CNF $F_p \subseteq \mathrm{Circuit}_s^{\mathrm{CNF}}(f)$ over the same variables as p (ignoring the added extension variables) such that $p(\alpha) = 0$ is satisfied by a Boolean assignment α if and only if F_p is satisfied by α (where we extend the assignment to the extension variables in the natural manner).

Recall that SoS operates on polynomials and we thus need to translate the CNF into a system of polynomials. We translate a clause $\vee_{i \in [w]} z_i$ into the polynomial $\prod_{i \in [w]}(1 - z_i) = 0$.

Observe that almost all axioms p of $\mathrm{Circuit}_s(f)$ depend only on a constant number of variables. From such p, using the appropriate Boolean axioms and negation axioms, we can in constant degree derive F_p.

Suppose we have a degree d refutation π of $\mathrm{Circuit}_s^{\mathrm{CNF}}(f)$. For all $v \in [s]$, $a \in \{1,2\}$, $i \in [n]$ and $u < v$, substitute the variable $\mathrm{isInput}^{\leq}(v,a,i)$ by $\sum_{j \leq i} \mathrm{isInput}(v,a,j)$ and $\mathrm{isGate}^{\leq}(v,a,u)$ by $\sum_{w \leq u} \mathrm{isGate}(v,a,w)$ in π. Also substitute the corresponding bar variables by one minus the appropriate sum. This results in a degree d SoS refutation π' of a formula $\mathrm{Circuit}_s^{\widetilde{\mathrm{CNF}}}(f)$.

We claim that in constant degree the axioms of $\mathrm{Circuit}_s^{\widetilde{\mathrm{CNF}}}(f)$ can be derived from $\mathrm{Circuit}_s(f)$. As previously noted, this holds for all axioms but the ones that are over a non-constant number of variables, i.e., what remains is to show that we can derive the substituted Axioms C.35 and C.34 from Axioms C.8, C.7 and C.5.

Let us consider Axiom C.34. With the extension variables substituted and translated into a system of polynomials the axiom consists of the following polynomial equations.

$$1 - \sum_{j \in [n]} \mathrm{isInput}(v,a,j) = 0 \tag{C.43}$$

$$\mathrm{isInput}(v,a,i) \cdot \mathrm{isInput}(v,a,j) = 0, \text{ for } i \neq j \tag{C.44}$$

$$\left(\sum_{j \leq i} \mathrm{isInput}(v,a,j)\right)\left(1 - \sum_{j < i} \mathrm{isInput}(v,a,j)\right) \cdot \overline{\mathrm{isInput}(v,a,i)} = 0, \text{ for } i \in [n] \tag{C.45}$$

$$\left(1 - \sum_{j \leq i} \mathrm{isInput}(v,a,j)\right)\left(\sum_{j < i} \mathrm{isInput}(v,a,j)\right) = 0, \text{ for } i \in [n] \tag{C.46}$$

$$\left(1 - \sum_{j \leq i} \mathrm{isInput}(v,a,j)\right) \mathrm{isInput}(v,a,i) = 0, \text{ for } i \in [n]\ . \tag{C.47}$$

Axiom C.43 is equal to the first Axiom in Axioms C.5 and similarly Axiom C.44 is equal to Axiom C.7. In the following we show that Axioms C.47, C.46 and C.45 can be derived from Axiom C.7, the Boolean axioms and the negation axioms in constant degree.

Consider Axiom C.45. Expand and rewrite modulo the Boolean axioms and the negation axiom to obtain

$$\begin{aligned}\mathrm{isInput}(\nu, a, i)\left(\sum_{j<i}\mathrm{isInput}(\nu, a, j)\right)^2 - \\ 2\sum_{j<j'<i}\mathrm{isInput}(\nu, a, j)\cdot\mathrm{isInput}(\nu, a, j') - \\ \mathrm{isInput}(\nu, a, i)\sum_{j<i}\mathrm{isInput}(\nu, a, j) = 0 \ . \end{aligned} \tag{C.48}$$

Observe that every term t left in this polynomial is of the form $t = t' \cdot \mathrm{isInput}(\nu, a, j) \cdot \mathrm{isInput}(\nu, a, j')$, for some $j \neq j' \in [i]$ and a term t' of degree at most 1. But this means that every term is equal to 0 modulo Axiom C.7 and we thus see that Axiom C.45 can be derived in constant degree from $\mathrm{Circuit}_s(f)$.

Let us consider Axiom C.46. Rewrite modulo the Boolean axiom to obtain

$$\begin{aligned}\mathrm{isInput}(\nu, a, i)\sum_{j<i}\mathrm{isInput}(\nu, a, j) + \\ 2\sum_{j<j'<i}\mathrm{isInput}(\nu, a, j)\cdot\mathrm{isInput}(\nu, a, j') = 0 \ . \end{aligned} \tag{C.49}$$

All terms are of the form of Axiom C.7 and we can thus derive Axiom C.46 from $\mathrm{Circuit}_s(f)$ in constant degree.

Last, we need to consider Axiom C.47. Note that modulo the Boolean axiom we obtain the polynomial equation

$$-\mathrm{isInput}(\nu, a, i)\sum_{j<i}\mathrm{isInput}(\nu, a, j) = 0 \ . \tag{C.50}$$

Also in this polynomial every term is of the form of Axiom C.7 and thus also Axiom C.47 can be derived in constant degree.

What remains is to show that Axiom C.35 can be derived from $\mathrm{Circuit}_s(f)$ in constant degree. This can be checked analogous to Axiom C.34 and we thus omit it here.

We conclude that all axioms of $\mathrm{Circuit}_s^{\widetilde{\mathrm{CNF}}}(f)$ can be derived from $\mathrm{Circuit}_s(f)$ in constant degree and thus a degree d SoS refutation of $\mathrm{Circuit}_s^{\mathrm{CNF}}(f)$ gives rise to a degree $O(d)$ SoS refutation of $\mathrm{Circuit}_s(f)$. Equivalently, a degree d lower bound for $\mathrm{Circuit}_s(f)$ implies a degree $\Omega(d)$ lower bound for $\mathrm{Circuit}_s^{\mathrm{CNF}}(f)$ as claimed.

C.8 Concluding Remarks

We have shown degree and size lower bounds in the Sum-of-Squares proof system for the minimum circuit size problem. There are a number of interesting questions left open for further study. Let us name a few.

Better Size Lower Bounds Whereas our degree lower bounds apply for all Boolean functions f, the corresponding size lower bounds only apply to an albeit rich but still restricted class of functions.

Monotone Circuit Lower Bounds For monotone circuits, we were only able to obtain lower bounds for slice functions (essentially because they behave in many ways like non-monotone functions). An intriguing question is whether this limitation can be overcome, or whether it is inherent and there exist some monotone circuit lower bounds that SoS *is* able to prove.

References

[ABRW04] M. Alekhnovich, E. Ben-Sasson, A. A. Razborov and A. Wigderson, "Pseudorandom generators in propositional proof complexity", *SIAM Journal on Computing*, vol. 34, no. 1, pp. 67–88, 2004, Preliminary version in *FOCS '00* (cit. on p. 175)

[AOW15] S. R. Allen, R. ODonnell and D. Witmer, "How to refute a random csp", in *2015 IEEE 56th Annual Symposium on Foundations of Computer Science (FOCS)*, Los Alamitos, CA, USA: IEEE Computer Society, Oct. 2015, pp. 689–708. DOI: 10.1109/FOCS.2015.48. [Online]. Available: https://doi.ieeecomputersociety.org/10.1109/FOCS.2015.48 (cit. on p. 175)

[AH19] A. Atserias and T. Hakoniemi, "Size-Degree Trade-Offs for Sums-of-Squares and Positivstellensatz Proofs", in *34th Computational Complexity Conference (CCC 2019)*, A. Shpilka, Ed., ser. Leibniz International Proceedings in Informatics (LIPIcs), vol. 137, Dagstuhl, Germany: Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 2019, 24:1–24:20, ISBN: 978-3-95977-116-0. DOI: 10.4230/LIPIcs.CCC.2019.24. [Online]. Available: http://drops.dagstuhl.de/opus/volltexte/2019/10846 (cit. on pp. 178, 181)

[BHK+16] B. Barak, S. B. Hopkins, J. Kelner, P. Kothari, A. Moitra and A. Potechin, "A nearly tight sum-of-squares lower bound for the planted clique problem", in *2016 IEEE 57th Annual Symposium on Foundations of Computer Science (FOCS)*, 2016, pp. 428–437 (cit. on p. 175)

[BW06] A. Beimel and E. Weinreb, "Monotone circuits for monotone weighted threshold functions", *Inf. Process. Lett.*, vol. 97, no. 1, pp. 12–18, Jan. 2006, ISSN: 0020-0190 (cit. on p. 196)

[Ber82] S. J. Berkowitz, "On some relationships between monotone and non-monotone circuit complexity", Technical Report, University of Toronto, Tech. Rep., 1982 (cit. on p. 180)

[BT06] A. Bogdanov and L. Trevisan, "Average-case complexity", *Foundations and Trends in Theoretical Computer Science*, vol. 2, no. 1, pp. 1–106, 2006, ISSN: 1551-305X. DOI: 10.1561/0400000004. [Online]. Available: http://dx.doi.org/10.1561/0400000004 (cit. on p. 176)

[GW95] M. X. Goemans and D. P. Williamson, "Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming", *J. ACM*, vol. 42, no. 6, pp. 1115–1145, 1995. DOI: 10.1145/227683.227684. [Online]. Available: https://doi.org/10.1145/227683.227684 (cit. on p. 175)

[Gol20] O. Goldreich, "On (valiant's) polynomial-size monotone formula for majority", in *Computational Complexity and Property Testing: On the Interplay Between Randomness and Computation*, O. Goldreich, Ed. Cham: Springer International Publishing, 2020, pp. 17–23, ISBN: 978-3-030-43662-9. DOI: 10.1007/978-3-030-43662-9_3. [Online]. Available: https://doi.org/10.1007/978-3-030-43662-9_3 (cit. on p. 196)

[Gri01] D. Grigoriev, "Linear lower bound on degrees of positivstellensatz calculus proofs for the parity", *Theoretical Computer Science*, vol. 259, no. 1, pp. 613–622, 2001, ISSN: 0304-3975. DOI: https://doi.org/10.1016/S0304-3975(00)00157-2. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0304397500001572 (cit. on pp. 175, 181)

[GUV09] V. Guruswami, C. Umans and S. Vadhan, "Unbalanced expanders and randomness extractors from Parvaresh–Vardy codes", *Journal of the ACM*, vol. 56, no. 4, 20:1–20:34, Jul. 2009, Preliminary version in *CCC '07* (cit. on pp. 178, 179)

[Hir18] S. Hirahara, ”Non-black-box worst-case to average-case reductions within np”, in *2018 IEEE 59th Annual Symposium on Foundations of Computer Science (FOCS)*, 2018, pp. 247–258. DOI: 10.1109/FOCS.2018.00032 (cit. on p. 174)

[KC00] V. Kabanets and J.-Y. Cai, ”Circuit minimization problem”, in *Proceedings of the Thirty-Second Annual ACM Symposium on Theory of Computing*, ser. STOC ’00, Portland, Oregon, USA: Association for Computing Machinery, 2000, pp. 73–79, ISBN: 1581131844. DOI: 10.1145/335305.335314. [Online]. Available: https://doi.org/10.1145/335305.335314 (cit. on p. 174)

[KMS98] D. Karger, R. Motwani and M. Sudan, ”Approximate graph coloring by semidefinite programming”, *J. ACM*, vol. 45, no. 2, pp. 246–265, Mar. 1998, ISSN: 0004-5411. DOI: 10.1145/274787.274791. [Online]. Available: https://doi.org/10.1145/274787.274791 (cit. on p. 175)

[KMOW17] P. K. Kothari, R. Mori, R. O’Donnell and D. Witmer, ”Sum of squares lower bounds for refuting any csp”, in *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing*, ser. STOC 2017, Montreal, Canada: Association for Computing Machinery, 2017, pp. 132–145, ISBN: 9781450345286. DOI: 10.1145/3055399.3055485. [Online]. Available: https://doi.org/10.1145/3055399.3055485 (cit. on p. 175)

[MPW15] R. Meka, A. Potechin and A. Wigderson, ”Sum-of-squares lower bounds for planted clique”, in *Proceedings of the 47th Annual ACM Symposium on Theory of Computing (STOC ’15)*, Jun. 2015, pp. 87–96 (cit. on p. 175)

[MW15] C. D. Murrayand and R. R. Williams, ”On the (Non) NP-Hardness of Computing Circuit Complexity”, in *30th Conference on Computational Complexity (CCC 2015)*, D. Zuckerman, Ed., ser. Leibniz International Proceedings in Informatics (LIPIcs), vol. 33, Dagstuhl, Germany: Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 2015, pp. 365–380, ISBN: 978-3-939897-81-1. DOI: 10.4230/LIPIcs.CCC.2015.365. [Online]. Available: http://drops.dagstuhl.de/opus/volltexte/2015/5074 (cit. on p. 174)

[RRS17] P. Raghavendra, S. Rao and T. Schramm, ”Strongly refuting random csps below the spectral threshold”, in *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Comput-*

ing, ser. STOC 2017, Montreal, Canada: Association for Computing Machinery, 2017, pp. 121–131, ISBN: 9781450345286. DOI: 10.1145/3055399.3055417. [Online]. Available: https://doi.org/10.1145/3055399.3055417 (cit. on p. 175)

[Raz98] A. A. Razborov, "Lower bounds for the polynomial calculus", *Computational Complexity*, vol. 7, no. 4, pp. 291–324, Dec. 1998 (cit. on pp. 175, 182)

[Raz04] ——, "Resolution lower bounds for perfect matching principles", *Journal of Computer and System Sciences*, vol. 69, no. 1, pp. 3–27, Aug. 2004, Preliminary version in *CCC '02* (cit. on pp. 175, 182)

[Raz15] ——, "Pseudorandom generators hard for k-DNF resolution and polynomial calculus resolution", *Annals of Mathematics*, vol. 181, no. 2, pp. 415–472, Mar. 2015 (cit. on pp. 175, 178)

[Raz21] ——, *P, NP and Proof Complexity*, Youtube, 2021. [Online]. Available: https://youtu.be/ZVL_HsPC4xE?t=2646 (visited on 05/04/2022) (cit. on p. 176)

[Raz22] ——, *Open problems*, 2022. [Online]. Available: https://people.cs.uchicago.edu/~razborov/teaching/index.html (visited on 05/04/2022) (cit. on p. 176)

[Val84] L. G. Valiant, "Short monotone formulae for the majority function", *J. Algorithms*, vol. 5, pp. 363–366, 1984 (cit. on p. 196)

Paper D

Exponential Resolution Lower Bounds for Weak Pigeonhole Principle and Perfect Matching Formulas over Sparse Graphs

Susanna F. de Rezende, Jakob Nordström,
Kilian Risse, and Dmitry Sokolov

Abstract

We show exponential lower bounds on resolution proof length for pigeonhole principle (PHP) formulas and perfect matching formulas over highly unbalanced, sparse expander graphs, thus answering the challenge to establish strong lower bounds in the regime between balanced constant-degree expanders as in [Ben-Sasson and Wigderson '01] and highly unbalanced, dense graphs as in [Raz '04] and [Razborov '03, '04]. We obtain our results by revisiting Razborov's pseudo-width method for PHP formulas over dense graphs and extending it to sparse graphs. This further demonstrates the power of the pseudo-width method, and we believe it could potentially be useful for attacking also other longstanding open problems for resolution and other proof systems.

D.1 Introduction

In one sentence, proof complexity is the study of efficient certificates of unsatisfiability for formulas in conjunctive normal form (CNF). In its most general form, this is the question of whether coNP can be separated from NP or not, and as such appears out of reach for current techniques. However, if one instead focuses on concrete proof systems, which can be thought of as restricted models of nondeterministic computation, this opens up the view to a rich landscape of results.

One line of research in proof complexity has been to prove superpolynomial lower bounds for stronger and stronger proof systems, as a way of approaching the distant goal of establishing $\mathsf{NP} \neq \mathsf{coNP}$. A perhaps even more fruitful direction, however, has been to study different combinatorial principles and investigate what kind of reasoning is needed to efficiently establish the validity of these principles. In this way, one can quantify the "depth" of different mathematical truths, measured in terms of how strong a proof system is required to prove them.

In this paper, we consider the proof system *resolution* [Bla37], in which one derives new disjunctive clauses from the formula until an explicit contradiction is reached. This is arguably the most well-studied proof system in proof complexity, for which numerous exponential lower bounds on proof size have been shown (starting with [Hak85; Urq87; CS88]). Yet many basic questions about resolution remain stubbornly open. One such set of questions concerns the *pigeonhole principle (PHP)* stating that there is no injective mapping of m pigeons into n holes if $m > n$. This is one of the simplest, and yet most useful, combinatorial principles in mathematics, and it has been topic of extensive study in proof complexity.

When studying the pigeonhole principle, it is convenient to think of it in terms of a bipartite graph $G = (U \,\dot\cup\, V, E)$ with pigeons $U = [m]$ and holes $V = [n]$ for $m \geq n + 1$. Every pigeon i can fly to its neighbouring pigeonholes $N(i)$ as specified by G, which for now we fix to be the complete bipartite graph $K_{m,n}$ with $N(i) = [n]$ for all $i \in [m]$. Since we wish to study unsatisfiable formulas, we encode the claim that there does in fact exist an injective mapping of pigeons to holes as a CNF formula consisting of *pigeon axioms*

$$P^i = \bigvee_{j \in N(i)} x_{ij} \qquad \text{for } i \in [m] \tag{D.1a}$$

and *hole axioms*

$$H_j^{i,i'} = (\overline{x}_{ij} \vee \overline{x}_{i'j}) \qquad \text{for } i \neq i' \in [m], j \in N(i) \cap N(i') \tag{D.1b}$$

(where the intended meaning of the variables is that $x_{i,j}$ is true if pigeon i flies to hole j). To rule out multi-valued mappings one can also add *functionality axioms*

$$F^{i}_{j,j'} = (\overline{x}_{ij} \lor \overline{x}_{ij'}) \qquad \text{for } i \in [m], j \neq j' \in N(i) \ , \tag{D.1c}$$

and a further restriction is to include *surjectivity* or *onto axioms*

$$S_j = \bigvee_{i \in N(j)} x_{ij} \qquad \text{for } j \in [n] \tag{D.1d}$$

requiring that every hole should get a pigeon. Clearly, the "basic" *pigeonhole principle (PHP) formulas* with clauses D.1a and D.1b are the least constrained. As one adds clauses D.1c to obtain the *functional pigeonhole principle (FPHP)* and also clauses D.1d to get the *onto functional pigeonhole principle (onto-FPHP)*, the formulas become more overconstrained and thus (potentially) easier to disprove, meaning that establishing lower bounds becomes harder. A moment of reflection reveals that onto-FPHP formulas are just saying that complete bipartite graphs with m left vertices and n right vertices have perfect matchings, and so these formulas are also referred to as *perfect matching formulas*.

Another way of varying the hardness of PHP formulas is by letting the number of pigeons m grow larger as a function of the number of holes n. What this means is that it is not necessary to count exactly to refute the formulas. Instead, it is sufficient to provide a precise enough estimate to show that $m > n$ must hold (where the hardness of this task depends on how much larger m is than n). Studying the hardness of such so-called *weak PHP formulas* gives a way of measuring how good different proof systems are at approximate counting. A second application of lower bounds for weak PHP formulas is that they can be used to show that proof systems cannot produce efficient proofs of the claim that $\mathsf{NP} \not\subseteq \mathsf{P/poly}$ [Raz98; Raz04b].

Yet another version of more constrained formulas is obtained by restricting what choices the pigeons have for flying into holes, by defining the formulas not over $K_{m,n}$ but sparse bipartite graphs with bounded left degree—such instances are usually called *graph PHP formulas*. Again, this makes the formulas easier to disprove in the sense that pigeons are more constrained, and it also removes the symmetry in the formulas that plays an essential role in many lower bound proofs.

Our work focuses on the most challenging setting in terms of lower bounds, when all of these restrictions apply: the PHP formulas contain both functionality and onto axioms, the number of pigeons m is very large compared to the number of holes n, and the choices of holes are restricted by a sparse graph. But before discussing our contributions, let us review what has been known about resolution and pigeonhole principle formulas. We emphasize that what will follow is a brief and selective overview focusing on resolution only—see Razborov's beautiful survey paper [Raz02] for a discussion of upper and lower bounds on PHP formulas in other proof systems.

D.1.1 Previous Work

In a breakthrough result, which served as a strong impetus for further developments in proof complexity, Haken [Hak85] proved a lower bound $\exp(\Omega(n))$ on resolution proof length for $m = n + 1$ pigeons. Haken's proof was for the basic PHP formulas, but easily extends to onto-FPHP formulas. This result was simplified and improved in a sequence of works [BT88; BP96; BW01; Urq03] to a lower bound of the form $\exp(n^2/m)$, which, unfortunately, does not yield anything nontrivial for $m = \Omega(n^2)$ pigeons.

Buss and Pitassi [BP97] showed that the pigeonhole principle does in fact get easier for resolution when m becomes sufficiently large: namely, for $m = \exp(\Omega(\sqrt{n \log n}))$ PHP formulas can be refuted in length $\exp(O(\sqrt{n \log n}))$. This is in contrast to what holds for the weaker subsystem *tree-like resolution*, for which the formulas remain equally hard as the number of pigeons increases, and where the complexity was even sharpened in [BP97; Dan02; DR01b; BGL10] to an $\exp(\Omega(n \log n))$ length lower bound.

Obtaining lower bounds beyond $m = n^2$ pigeons for non-tree-like resolution turned out to be quite challenging. Haken's bottleneck counting method fundamentally breaks down when the number of pigeons is quadratic in the number of holes, and the same holds for the celebrated length-width lower bound in [BW01]. Some progress was made for restricted forms of resolution in [RWY02] and [PR04], leading up to an $\exp(n^{\varepsilon})$ lower bound for so-called *regular resolution*. In a technical tour de force, Raz [Raz04a] finally proved that general, unrestricted resolution requires length $\exp(n^{\varepsilon})$ to refute the basic PHP formulas even with arbitrary many pigeons. Razborov followed up on this in three papers where he first simplified and slightly strengthened Raz's result in [Raz01], then extended it to FPHP formulas in [Raz03] and lastly established an analogous lower bound for onto-FPHP formulas in [Raz04b].

More precisely, what Razborov showed is that for any version of the PHP formula with m pigeons and n holes, the minimal proof length required in resolution is $\exp(\Omega(n/\log^2 m))$. It is easy to see that this implies a lower bound $\exp(\Omega(\sqrt[3]{n}))$ for any number of pigeons—for $m = \exp(O(\sqrt[3]{n}))$ we can appeal directly to the bound above, and if a resolution proof would use $\exp(\Omega(\sqrt[3]{n}))$ pigeons, then just mentioning all these different pigeons already requires $\exp(\Omega(\sqrt[3]{n}))$ distinct clauses. It is also clear that considering complexity in terms of the number of holes n is the right measure. Since any formula contains a basic PHP subformula with $n+1$ pigeons that can be refuted in length $\exp(O(n))$, we can never hope for exponential lower bounds in terms of formula size as the number of pigeons m grows to exponential.

So far we have stated results only for the standard PHP formulas over $K_{m,n}$, where any pigeon can fly to any hole. However, the way Ben-Sasson and Wigderson [BW01] obtained their result was by considering graph PHP formulas over balanced bipartite expander graphs of constant left degree, from which the lower bound for $K_{m,n}$ easily follows by a restriction argument. It was shown in [IOSS16] that an analogous bound holds for onto-FPHP formulas, i.e., perfect matching formulas, on bipartite expanders. In this context is is also relevant to mention the exponential lower bounds in [Ale04; DR01a] on *mutilated chessboard formulas*, which can be viewed as perfect matching formulas on balanced, sparse bipartite graphs with very bad expansion. At the other end of the spectrum, Razborov's PHP lower bound in [Raz04b] for highly unbalanced bipartite graphs also applies in a more general setting than $K_{m,n}$: namely, for any graph where the minimal degree of any left vertex is δ, the minimal length of any resolution proof is $\exp\left(\Omega\left(\delta/\log^2 m\right)\right)$. Thus, for graph PHP formulas we have exponential lower bounds on the one hand [BW01] for $m \ll n^2$ pigeons, where each pigeon is adjacent to a constant number of holes, and on the other hand [Raz04b] for any number of pigeons given that each pigeon is adjacent to a polynomial $n^{\Omega(1)}$ number of holes, but nothing has been known in between these extremes. In [Raz04b], Razborov asks whether a *"common generalization"* of the techniques in [BW01] and [Raz03; Raz04b] can be found *"that would uniformly cover both cases?"* Urquhart [Urq07] also discusses Razborov's lower bound technique, but notes that *"the search for a yet more general point of view remains a topic for further research."*

D.1.2 Our Results

In this work, we give an answer to the questions raised in [Raz04b; Urq07] by presenting a general technique that applies for any number of pigeons m all the way from linear to weakly exponential, and that establishes exponential lower bounds on resolution proof length for all flavours of graph PHP formulas (including perfect matching formulas) even over sparse graphs.

Let us state below three examples of the kind of lower bounds we obtain—the full, formal statements will follow in later sections. Our first theorem is an average-case lower bound for onto-FPHP formulas with slightly superpolynomial number of pigeons.

Theorem D.1.1 (Informal)**.** *Let* G *be a randomly sampled bipartite graph with* n *right vertices,* $m = n^{o(\log n)}$ *left vertices, and left degree* $\Theta(\log^2 m)$*. Then refuting the onto-FPHP formula (a.k.a. perfect matching formula) over* G *in resolution requires length* $\exp(\Omega(n^{1-o(1)}))$ *asymptotically almost surely.*

Note that as the number of pigeons grow larger, it is clear that the left degree also has to grow—otherwise we will get a small number of pigeons constrained to fly to a small number of holes by a birthday paradox argument, yielding a small unsatisfiable subformula that can easily be refuted by brute force.

If the number of pigeons increases further to weakly exponential, then randomly sampled graphs no longer have good enough expansion for our technique to work, but there are explicit constructions of unbalanced expanders for which we can still get lower bounds.

Theorem D.1.2 (Informal)**.** *There are explicitly constructible bipartite graphs* G *with* n *right vertices,* $m = \exp(O(n^{1/16}))$ *left vertices, and left degree* $\Theta(\log^4 m)$ *such that resolution requires length* $\exp(\Omega(n^{1/8-\varepsilon}))$ *to refute the perfect matching formula over* G*.*

Finally, for functional pigeonhole principle formulas we can also prove an exponential lower bound for *constant* left degree even if the number of pigeons is a large polynomial.

Theorem D.1.3 (Informal)**.** *Let* G *be a randomly sampled bipartite graph with* n *right vertices,* $m = n^k$ *left vertices, and left degree* $\Theta((k/\varepsilon)^2)$*. Then refuting the functional pigeonhole principle formula over* G *in resolution requires length* $\exp(\Omega(n^{1-\varepsilon}))$ *asymptotically almost surely.*

D.1.3 Techniques

At a very high level, what we do in terms of techniques is to revisit the pseudo-width method introduced by Razborov for functional PHP formulas in [Raz03]. We strengthen this method to work in the setting of sparse graphs by combining it with the closure operation on expander graphs in [AR03; ABRW04], which is a way to restore expansion after a small set of (potentially adversarially chosen) vertices have been removed. To extend the results further to perfect matching formulas, we apply a "preprocessing step" on the formulas as in [Raz04b]. In what remains of this section, we focus on graph FPHP formulas and give an informal overview of the lower bound proof in this setting, which already contains most of the interesting ideas (although the extension to onto-FPHP also raises significant additional challenges).

Let FPHP(G) denote the functional pigeonhole principle formula over the graph G consisting of clauses D.1a–D.1c. A first, quite naive (and incorrect), description of the proof structure is that we start by defining a *pseudo-width* measure on clauses C that counts pigeons i that appear in C in many variables x_{ij} for distinct j. We then show that any short resolution refutation of FPHP(G) can be transformed into a refutation where all clauses have small pseudo-width. By a separate argument, we establish that any refutation of FPHP(G) requires large pseudo-width. Hence, no short refutations can exist, which is precisely what we were aiming to prove.

To fill in the details (and correct) this argument, let us start by making clear what we mean by pseudo-width. Suppose that the graph G has left degree Δ. In what follows, we identify a mapping of pigeon i to a neighbouring hole j with the partial assignment ρ such that $\rho(x_{i,j}) = 1$ and $\rho(x_{i,j'}) = 0$ for all $j' \in N(i) \setminus \{j\}$. We denote by $d_i(C)$ the number of mappings of pigeon i that satisfy C. Note that if C contains at least one negated literal $\overline{x}_{i,j}$, then $d_i(C) \geq \Delta - 1$, and otherwise $d_i(C)$ is the number of positive literals $x_{i,j}$ for $j \in N(i)$. Given a judiciously chosen "filter vector" $d = (d_1, \ldots, d_m)$ for $d_i \approx \Delta$ and a "slack" $\delta \approx \Delta / \log m$, we say that pigeon i is *heavy* in C if $d_i(C) \geq d_i - \delta$ and *super-heavy* if $d_i(C) \geq d_i$. We define the *pseudo-width* of a clause C to be the number of heavy pigeons in C.

With these definitions in hand, we can give a description of the actual proof:

1. Given any resolution refutation π of FPHP(G) in small length L, we argue that all clauses can be classified as having either low or high pseudo-width, where an important additional guarantee is that the high-width clauses not only have many heavy pigeons but actually many super-heavy pigeons.

2. We replace all clauses C with many super-heavy pigeons with "fake axioms" $C' \subseteq C$ obtained by throwing away literals from C until we have nothing left but a medium number of super-heavy pigeons. By construction, the set A of such fake axioms is of size $|A| \leq L$, and after making the replacement we have a resolution refutation π' of FPHP(G) $\cup$ A in low pseudo-width.

3. However, since A is not too large, we are able to show that any resolution refutation of FPHP(G) $\cup$ A must still require large pseudo-width. Hence, L cannot be small, and the lower bound follows.

Part 1 is similar to [Raz03], but with a slight twist. We show that if the length of π is $L < 2^{w_0}$ and if we choose $\delta \leq \varepsilon \Delta \log n / \log m$, then there exists a vector $d = (d_1, \ldots, d_m)$ such that for all clauses in π either the number of super-heavy pigeons is at least w_0 or else the number of heavy pigeons is at most $O(w_0 \cdot n^{\varepsilon})$. The proof of this is by sampling the coordinates d_i independently from a suitable probability distribution and then applying a union bound argument. Once this has been established, part 2 follows easily: we just replace all clauses with at least w_0 super-heavy pigeons by (stronger) fake axioms. Including all fake axioms A yields a refutation π' of $\text{FPHP}(G) \cup A$ (since we can add a weakening rule deriving C from $C' \subseteq C$ to resolution without loss of generality) and clearly all clauses in π' have pseudo-width $O(w_0 \cdot n^{\varepsilon})$.

Part 3 is where most of the hard work is. Suppose that G is an excellent expander graph, so that for some value r all left vertex sets U' of size $|U'| \leq r$ have at least $(1 - \varepsilon \log n / \log m)\Delta|U'|$ unique neighbours on the right-hand side. We show that, under the assumptions above, refuting $\text{FPHP}(G) \cup A$ requires pseudo-width $\Omega(r \cdot \log n / \log m)$. Tuning the parameters appropriately, this yields a contradiction with part 2.

Before outlining how the proof of part 3 goes, we remark that the requirements we place on the expansion of G are quite severe. Clearly, any left vertex set U can have at most $\Delta|U'|$ neighbours in total, and we are asking for all except a vanishingly small fraction of these neighbours to be unique. This is why we can etablish Theorem D.1.1 but not Theorem D.1.2 for randomly sampled graphs. We see no reason to believe that the latter theorem would not hold also for random graphs, but the expansion properties required for our proof are so stringent that they are not satisfied in this parameter regime. This seems to be a fundamental shortcoming of our technique, and it appears that new ideas would be required to circumvent this problem.

In order to argue that refuting $\text{FPHP}(G) \cup A$ in resolution requires large pseudo-width, we want to estimate how much progress the resolution derivation has made up to the point when it derives some clause C. Following Razborov's lead, we measure this by looking at what fraction of partial matchings of all the heavy pigeons in C do not satisfy C (meaning, intuitively, that the derivation has managed to rule out this part of the search space). It is immediate by inspection that all pigeons mentioned in the real axiom clauses D.1a–D.1c are heavy, and any matching of such pigeons satisfies the clauses. Thus, the original axioms in $\text{FPHP}(G)$ do not rule out any matchings. Also, it is easy to show that fake axioms rule out only an exponentially small fraction of matchings, since they contain many super-heavy pigeons and it is hard to match all of these pigeons without satisfying the clause. However, the contradictory empty clause $\perp$ rules out 100% of partial matchings, since it contains no heavy pigeons to match in the first place.

What we would like to prove now is that for any derivation in small pseudo-width it holds that the derived clause cannot rule out any matching other than those already eliminated by the clauses used to derive it. This means that the fake axioms together need to rule out all partial matchings, but since every fake axiom contributes only an exponentially small fraction they are too few to achieve this. Hence, it is not possible to derive contradiction in small pseudo-width, which completes part 3 of our proof outline.

There is one problem, however: the last claim above is not true, and so what is outlined above is only a fake proof. While we have to defer the discussion of what the full proof actually looks like in detail, we conclude this section by attempting to hint at a couple of technical issues and how to resolve them.

Firstly, it does not hold that a derived clause C eliminates only those matchings that are also forbidden by one of the predecessor clauses used to derive C. The issue is that a pigeon i that is heavy in both predecessors might cease to be heavy in C—for instance, if C was derived by a resolution step over a variable $x_{i,j}$. If this is so, then we would need to show that any matching of the heavy pigeons in C can be extended to match also pigeon i to any of its neighbouring holes without satisfying both predecessor clauses. But this will not be true, because a non-heavy pigeon can still have some variable $x_{i,j}$ occurring in both predecessors. The solution to this, introduced in [Raz03], is to do a "lossy counting" of matchings by associating each partial matching with a linear subspace of some suitable vector space, and then to consider the span of all matchings ruled out by C. When we accumulate a "large enough" number of matchings for a pigeon i, then the whole subspace associated to i is spanned and we can stop counting.

But this leads to a second problem: when studying matchings of the heavy pigeons in C we might already have assigned pigeons $i'_1, \ldots, i'_w$ that occupy holes where pigeon i might want to fly. For standard PHP formulas over complete bipartite graphs this is not a problem, since at least $n - w$ holes are still available and this number is "large enough" in the sense described above. But for a sparse graph it will typically be the case that $w \gg \Delta$, and so it might well be the case that pigeons $i'_1, \ldots, i'_w$ are already occupying all the Δ holes available for pigeon i according to G. Although it is perhaps hard to see from our (admittedly somewhat informal) discussion, this turns out to be a very serious problem, and indeed it is one of the main technical challenges we need to overcome.

To address this problem we consider not only the heavy pigeons in C, but also any other pigeons in G that risk becoming far too constrained when the heavy pigeons of C are matched. Inspired by [AR03; ABRW04], we define the *closure* to be a superset S of the heavy pigeons such that when S and the neighbouring holes of S are removed it holds that the residual graph is still guaranteed to be a good expander. Provided that G is an excellent expander to begin with, and that the number of heavy pigeons in C is not too large, it can then be shown that an analogue of the original argument outlined above goes through.

D.1.4 Outline of This Paper

We review the necessary preliminaries in Section D.2 and introduce two crucial technical tools in Section D.3. The lower bounds for weak graph FPHP formulas are then presented in Section D.4, after which the perfect matching lower bounds follow in Section D.5. We conclude with a discussion of questions for future research in Section D.6.

D.2 Preliminaries

We denote natural logarithms (base e) by ln, and base 2 logarithms by log. For positive integers $n \in \mathbb{N}^+$ we write $[n] = \{1, \ldots, n\}$.

A *literal* over a Boolean variable x is either the variable x itself (a *positive literal*) or its negation $\overline{x}$ (a *negative literal*). A *clause* $C = \ell_1 \lor \cdots \lor \ell_w$ is a disjunction of literals. We write $\bot$ to denote the empty clause without any literals. A *CNF formula* $F = C_1 \land \cdots \land C_m$ is a conjunction of clauses. We think of clauses and CNF formulas as sets: order is irrelevant and there are no repetitions. We let F denote the set of variables of F.

A *resolution refutation* π of an unsatisfiable CNF formula F, or *resolution proof* for (the unsatisfiability of) F, is an ordered sequence of clauses $\pi = (D_1, \ldots, D_L)$ such that $D_L = \bot$ and for each $i \in [L]$ either D_i is a clause in F (an *axiom*) or there exist $j < i$ and $k < i$ such that D_i is derived from D_j and D_k by the *resolution rule*

$$\frac{B \lor x \qquad C \lor \overline{x}}{B \lor C} \; . \tag{D.2}$$

We refer to $B \lor C$ as the *resolvent* of $B \lor x$ and $C \lor \overline{x}$ over x, and to x as the *resolved variable*. For technical reasons it is sometimes convenient to also allow clauses to be derived by the *weakening* rule

$$\frac{C}{D} \; [C \subseteq D] \tag{D.3}$$

(and for two clauses $C \subseteq D$ we will sometimes refer to C as a *strengthening* of D).

The *length* $L(\pi)$ of a refutation $\pi = (D_1, \ldots, D_L)$ is L. The length of refuting F is $\min_{\pi:F\vdash\bot}\{L(\pi)\}$, where the minimum is taken over all resolution refutations π of F. It is easy to show that removing the weakening rule D.3 does not increase the refutation length.

A *partial assignment* or a *restriction* on a formula F is a partial function $\rho : F \to \{0, 1\}$. The clause C *restricted by* ρ, denoted $C\lceil_\rho$, is the trivial 1-clause if any of the literals in C is satisfied by ρ and otherwise it is C with all falsified literals removed. We extend this definition to CNF formulas in the obvious way by taking unions. For a variable $x \in F$ we write $\rho(x) = *$ if $x \notin \mathrm{dom}(\rho)$, i.e., if ρ does not assign a value to x.

We write $G = (V, E)$ to denote a graph with vertices V and edges E, where G is always undirected and without loops or multiple edges. Moreover, for bipartite graphs we write $G = (U \dot\cup V, E)$, where edges in E have one endpoint in the left vertex set U and the other in the right vertex set V. A *partial matching* φ in G is a subset of edges that are vertex-disjoint. Let $V(\varphi) = \{v \mid \exists e \in \varphi : v \in e\}$ be the vertices of φ and for $v \in V(\varphi)$ denote by φ_v the unique vertex u such that $\{u, v\} \in \varphi$. A vertex v is *covered* by φ if $v \in V(\varphi)$. If φ is a partial matching in a bipartite graph $G = (U \dot\cup V, E)$, we identify it with a partial mapping of U to V. When referring to the pigeonhole formula, this mapping will also be identified with an assignment ρ_φ to the variables defined by

$$\rho_\varphi(x_{i,j}) = \begin{cases} * & \text{if } i \notin \mathrm{Dom}(\varphi), \\ 0 & \text{if } i \in \mathrm{Dom}(\varphi) \text{ and } \varphi(i) \neq j, \\ 1 & \text{if } i \in \mathrm{Dom}(\varphi) \text{ and } \varphi(i) = j. \end{cases} \tag{D.4}$$

Given a vertex $v \in V(G)$, we write $N_G(v)$ to denote the set of *neighbours of* v in the graph G and $\deg_G(v) = |N_G(v)|$ to denote the degree of v. We extend this notion to sets and denote by $N_G(S) = \{v \mid \exists (u, v) \in E \text{ for } u \in S\}$ the *neighbourhood* of a set of vertices $S \subseteq V$. The *boundary*, or *unique neighbourhood*, $\partial_G(S) = \{v \in V \setminus S : |N_G(v) \cap S| = 1\}$ of a set of vertices $S \subseteq V$ contains all vertices in $V \setminus S$ that have a single neighbour in S. If the graph is bipartite, there is of course no need to subtract S from the neighbour set. We will sometimes drop the subscript G when the graph is clear from context. For a set $U \subseteq V$ we denote by $G \setminus U$ the subgraph of G induced by the vertex set $V \setminus U$.

A graph $G = (V, E)$ is an (r, Δ, c)-*expander* if all vertives $v \in V$ have degree at most Δ and for all sets $S \subseteq V$, $|S| \leq r$, it holds that $|N(S) \setminus S| \geq c \cdot |S|$. Similarly, $G = (V, E)$ is an (r, Δ, c)-*boundary expander* if all vertices $v \in V$ have degree at most Δ and for all sets $S \subseteq V$, $|S| \leq r$, it holds that $|\partial(S)| \geq c \cdot |S|$. For bipartite graphs, the degree and expansion requirements only apply to the left vertex set: $G = (U \dot\cup V, E)$ is an (r, Δ, c)-*bipartite expander* if all vertices $u \in U$ have degree at most Δ and for all sets $S \subseteq U$, $|S| \le r$, it holds that $|N(S)| \ge c \cdot |S|$, and an (r, Δ, c)-*bipartite boundary expander* if for all sets $S \subseteq U$, $|S| \le r$, it holds that $|\partial(S)| \ge c \cdot |S|$. For bipartite graphs we will only ever be interested in bipartite notions of expansions, and so which kind of expansion is meant will always be clear from context. A simple but useful observation is that

$$|N(S) \setminus S| \le |\partial(S)| + \frac{\Delta|S| - |\partial(S)|}{2} = \frac{\Delta|S| + |\partial(S)|}{2} , \tag{D.5}$$

since all non-unique neighbours in $N(S) \setminus S$ have at least two incident edges. This implies that if a graph G is an $(r, \Delta, (1 - \xi)\Delta)$-expander then it is also an $(r, \Delta, (1 - 2\xi)\Delta)$-boundary expander.

We often denote random variables in boldface and write $\mathbf{X} \sim \mathcal{D}$ to denote that $\mathbf{X}$ is sampled from the distribution $\mathcal{D}$. We will use the following standard forms of the multiplicative Chernoff bounds: if $\mathbf{S}$ is a sum of independent 0-1 random variables (not necessarily equidistributed) with expectation $\mu = \mathbb{E}[\mathbf{S}]$, then for $\delta \ge 0$ we have that $\Pr\left[\mu - \mathbf{S} \ge \delta\right] \le \exp(-\frac{\delta^2}{2\mu})$ and $\Pr\left[\mathbf{S} - \mu \ge \delta\right] \le \exp(-\frac{\delta^2}{2\mu+\delta})$. Combining these two inequalities yields the following statement.

Theorem D.2.1. *Let* $\mathbf{S}$ *be the sum of independent* 0-1 *random variables* (*not necessarily equidistributed*) *with expectation* $\mu = \mathbb{E}[\mathbf{S}]$. *Then for* $\delta \ge 0$ *it holds that*

$$\Pr\left[|\mathbf{S} - \mu| \ge \delta\right] \le 2\exp\left(-\frac{\delta^2}{2\mu + \delta}\right) .$$

For $n, m, \Delta \in \mathbb{N}$, we denote by $\mathcal{G}(m, n, \Delta)$ the distribution over bipartite graphs with disjoint vertex sets $U = \{u_1, \ldots, u_m\}$ and $V = \{v_1, \ldots, v_n\}$ where the neighbourhood of a vertex $u \in U$ is chosen by sampling a subset of size Δ uniformly at random from V. A property is said to hold *asymptotically almost surely* on $\mathcal{G}(f(n), n, \Delta)$ if it holds with probability that approaches 1 as n approaches infinity.

For the right parameters, a randomly sampled graph $G \sim \mathcal{G}(m, n, \Delta)$ is asymptotically almost surely a good boundary expander as stated next.

Lemma D.2.2. *Let* m, n *and* Δ *be large enough integers such that* $m > n \ge \Delta$. *Let* $\xi, \chi \in \mathbb{R}^+$ *be such that* $\xi < 1/2$, $\xi \ln \chi \ge 2$ *and* $\xi\Delta \ln \chi \ge 4 \ln m$. *Then for* $r = n/(\Delta \cdot \chi)$ *and* $c = (1 - 2\xi)\Delta$ *it holds asymptotically almost surely for a randomly sampled graph* $G \sim \mathcal{G}(m, n, \Delta)$ *that* G *is an* (r, Δ, c)*-boundary expander.*

Proof. Let $G = (U \mathbin{\dot\cup} V, E)$. We first estimate the probability that a set $S \subseteq U$ of size at most r violates the boundary expansion. For brevity, let us write

$s = |S|$ and $c' = (1 - \xi)\Delta$. In view of D.5, the probability that S violates the boundary expansion can be bounded by

$$
\begin{align}
\Pr\big[|\partial(S)| < cs\big] &\leq \Pr\left[|N(S)| < \frac{\Delta s + cs}{2}\right] \tag{D.6a}\\
&= \Pr\big[|N(S)| < c's\big] \tag{D.6b}\\
&\leq \binom{n}{c's} \cdot \left(\frac{\binom{c's}{\Delta}}{\binom{n}{\Delta}}\right)^{s} \tag{D.6c}\\
&\leq \binom{n}{c's} \cdot \left(\frac{c's}{n}\right)^{\Delta s} \tag{D.6d}\\
&\leq \left[\left(\frac{en}{c's}\right)^{c'} \cdot \left(\frac{c's}{n}\right)^{\Delta}\right]^{s} \tag{D.6e}\\
&= \left[e^{(1-\xi)\Delta} \cdot \left(\frac{n}{c's}\right)^{-\xi\Delta}\right]^{s} \tag{D.6f}\\
&\leq \exp\left(\Delta s \left(1 - \xi \ln\left(\frac{n}{c's}\right)\right)\right) \tag{D.6g}\\
&\leq \exp\left(\Delta s \left(1 - \xi \ln\left(\frac{\chi}{1-\xi}\right)\right)\right) \tag{D.6h}\\
&\leq \exp\left(\Delta s(1 - \xi \ln \chi)\right) \tag{D.6i}\\
&\leq \exp\left(-(\Delta s \xi \ln \chi)/2\right) \quad , \tag{D.6j}
\end{align}
$$

where (D.6h) holds since $s \leq r \leq n/(\Delta\chi)$ and (D.6j) holds since $\xi \ln \chi \geq 2$. Hence, the probability that G is not a boundary expander can be bounded by

$$
\begin{align}
\Pr\big[G \text{ is not an expander}\big] &\leq \sum_{s \in [r]} \binom{m}{s} \exp(-(\Delta s \xi \ln \chi)/2) \notag\\
&\leq \sum_{s \in [r]} \exp(-s((\xi \Delta \ln \chi)/2 - \ln m)) \tag{D.7}\\
&\leq \sum_{s \in [r]} \exp(-s \ln m) \leq \frac{1}{m-1} \quad , \notag
\end{align}
$$

where the second-to-last inequality holds since $\xi \Delta \ln \chi \geq 4 \ln m$. □

We will also consider some parameter settings where randomly sampled graphs do not have strong enough expansion for our purposes, but where we can resort to explicit constructions as follows.

Theorem D.2.3 ([GUV09]). *For all positive integers m, $r \le m$, all $\xi > 0$, and all constant $\nu > 0$, there is an explicit $(r, \Delta, (1-\xi)\Delta)$-expander $G = (U \mathbin{\dot\cup} V, E)$, with $|U| = m$, $|V| = n$, $\Delta = O\left(((\log m)(\log r)/\xi)^{1+1/\nu}\right)$ and $n \le \Delta^2 \cdot r^{1+\nu}$.*

Corollary D.2.4. *Let κ, ε, ν be positive constants, $\kappa < \frac{1}{8}$, and let n be a large enough integer. Then there is an explicit graph $G = (U \mathbin{\dot\cup} V, E)$, with $|U| = m = 2^{\Omega(n^\kappa)}$ and $|V| \le n$, that is an $(n^{\frac{1}{1+\nu} - \frac{4\kappa}{\nu}}, \Delta, (1-2\xi)\Delta)$-boundary expander for $\xi = \frac{\varepsilon \log n}{\log m}$ and $\Delta = O(\log^{2(1+1/\nu)} m)$.*

Proof. Let G be the expander from Theorem D.2.3 for the parameters $m = 2^{\varepsilon' n^\kappa}$, $r = n^{\frac{1}{1+\nu} - \frac{4\kappa}{\nu}}$, and $\xi = \frac{\varepsilon \log n}{\log m}$, where ε' is chosen to be a small enough constant so that $\Delta^2 \cdot r^{1+\nu} \le n$. Such a graph G is an $(r, \Delta, (1-\xi)\Delta)$-expander for Δ as in the Corollary. By D.5 it follows that an (r, Δ, c)-expander is an $(r, \Delta, 2c - \Delta)$-boundary expander, and hence G is an $\big(r, \Delta, (1-2\xi)\Delta\big)$-boundary expander. Note that Theorem D.2.3 guarantees that the right side of G has size at most $\Delta^2 \cdot r^{1+\nu} \le n$. □

D.3 Two Key Technical Tools

In this section we review two crucial technical ingredients of the resolution lower bound proofs.

D.3.1 Pigeon Filtering

The following lemma is a generalization of [Raz03, Lemma 6]. The difference is that we have an additional parameter α (which is implicitly fixed to $\alpha = 2$ in [Raz03]) that allows us to get a better upper bound on the numbers r_i. This turns out to be crucial for us—we discuss this in more detail in Section D.4.

Lemma D.3.1 (Filter lemma). *Let $m, L \in \mathbb{N}^+$ and suppose that $w_0, \alpha \in [m]$ are such that $w_0 > \ln L$ and $w_0 \ge \alpha^2 \ge 4$. Further, let $r(1), \ldots, r(L)$ be integer vectors, each of the form $r(\ell) = (r_1(\ell), \ldots, r_m(\ell))$. Then there exists a vector $r = (r_1, \ldots, r_m)$ of positive integers $r_i \le \left\lfloor \frac{\log m}{\log \alpha} \right\rfloor - 1$ such that for all $\ell \in [L]$ at least one of the following holds:*

1. $\Big|\{i \in [m] : r_i(\ell) \le r_i\}\Big| \ge w_0$,
2. $\Big|\{i \in [m] : r_i(\ell) \le r_i + 1\}\Big| \le O(\alpha \cdot w_0)$.

Proof. We first define a weight function $W(r)$ for vectors $r = (r_1, \ldots, r_m)$ as

$$W(r) = \sum_{i \in [m]} \alpha^{-r_i} \quad . \tag{D.8}$$

In order to establish the lemma, it is sufficient to show that there exist constants γ and γ' and a vector $r = (r_1, \ldots, r_m)$ such that for all $\ell \in [L]$ the implications

$$W(r(\ell)) \geq \frac{\gamma' w_0}{\alpha} \Rightarrow |\{i \in [m] \mid r_i(\ell) \leq r_i\}| \geq w_0 \ , \tag{D.9a}$$

$$W(r(\ell)) \leq \frac{\gamma' w_0}{\alpha} \Rightarrow |\{i \in [m] \mid r_i(\ell) \leq r_i + 1\}| \leq \gamma\alpha w_0 \tag{D.9b}$$

hold. Let $t = \left\lfloor \frac{\log m}{\log \alpha} \right\rfloor - 1$ and let μ be a probability distribution on $[t]$ given by $\Pr[\mathbf{r} = i] = \beta \cdot \alpha^{-i}$ for all $i \in [t]$, where $\beta = \frac{\alpha-1}{1-\alpha^{-t}}$. Note that

$$\beta \sum_{i \in [t]} \alpha^{-i} = \frac{\alpha - 1}{1 - \alpha^{-t}} \left(\frac{1 - \alpha^{-t}}{\alpha - 1} \right) = 1 \tag{D.10}$$

and thus μ is a valid distribution. Let us write $\mathbf{r} = (\mathbf{r}_1, \ldots, \mathbf{r}_m)$ to denote a random vector with coordinates sampled independently according to μ. We claim that for every $\ell \in [L]$ the implications D.9a and D.9b are true asymptotically almost surely. Let us proceed to verify this.

1. Suppose that $W(r(\ell)) \geq \frac{\gamma' w_0}{\alpha}$. We wish to show that $|\{i \in [m] : r_i \geq r_i(\ell)\}| \geq w_0$. Observe that coordinates larger than t contribute only

$$\sum_{r_i(\ell) > t} \alpha^{-r_i(\ell)} \leq m \cdot \alpha^{-t-1} < \alpha \tag{D.11}$$

to $W(r(\ell))$, and hence the weight function truncated at t is

$$\sum_{r_i(\ell) \leq t} \alpha^{-r_i(\ell)} \geq \frac{\gamma' w_0}{\alpha} - \alpha \geq (\gamma' - 1)\frac{w_0}{\alpha} \ , \tag{D.12}$$

since $w_0 \geq \alpha^2$. Note that for every coordinate i with $r_i(\ell) \leq t$ we have that $\Pr[\mathbf{r}_i \geq r_i(\ell)] \geq \beta \cdot \alpha^{-r_i(\ell)}$. Consider the random set $P_{\mathbf{r}}(\ell) = \{i \in [m] \mid r_i(\ell) \leq t \text{ and } \mathbf{r}_i \geq r_i(\ell)\}$. We can appeal to D.12 to derive that

$$\begin{aligned}\mathbb{E}\left[|P_{\mathbf{r}}(\ell)|\right] &= \sum_{r_i(\ell) \leq t} \Pr[\mathbf{r}_i \geq r_i(\ell)] \\ &\geq \sum_{r_i(\ell) \leq t} \beta\alpha^{-r_i(\ell)} \\ &\geq \beta(\gamma' - 1)\frac{w_0}{\alpha} \geq \frac{\gamma' - 1}{2} w_0 \end{aligned} \tag{D.13}$$

is a lower bound on the expected size of $P_{\mathbf{r}}(\ell)$. As the events $\mathbf{r}_i \geq r_i(\ell)$ are independent, by the multiplicative Chernoff bound we get that

$$\begin{aligned}
\Pr\big[|P_{\mathbf{r}}(\ell)| < w_0\big] &\leq \Pr\big[|P_{\mathbf{r}}(\ell)| - \mathbb{E}\big[|P_{\mathbf{r}}(\ell)|\big] \leq w_0 - \mathbb{E}\big[|P_{\mathbf{r}}(\ell)|\big]\big] && \text{(D.14a)}\\
&= \Pr\big[\mathbb{E}\big[|P_{\mathbf{r}}(\ell)|\big] - |P_{\mathbf{r}}(\ell)| \geq \mathbb{E}\big[|P_{\mathbf{r}}(\ell)|\big] - w_0\big] && \text{(D.14b)}\\
&\leq \exp\Big(-\frac{\big(\mathbb{E}\big[|P_{\mathbf{r}}(\ell)|\big] - w_0\big)^2}{2\,\mathbb{E}\big[|P_{\mathbf{r}}(\ell)|\big]}\Big) && \text{(D.14c)}\\
&= \exp\left(-\frac{\mathbb{E}\big[|P_{\mathbf{r}}(\ell)|\big]^2 - 2\,\mathbb{E}\big[|P_{\mathbf{r}}(\ell)|\big]w_0 + w_0^2}{2\,\mathbb{E}\big[|P_{\mathbf{r}}(\ell)|\big]}\right) && \text{(D.14d)}\\
&\leq \exp\left(-\frac{\mathbb{E}\big[|P_{\mathbf{r}}(\ell)|\big] - 2w_0}{2}\right) && \text{(D.14e)}\\
&\leq \exp\left(-\frac{(\gamma' - 5)}{4}w_0\right) && \text{(D.14f)}\\
&\leq \exp(-2w_0) && \text{(D.14g)}\\
&\leq L^{-2}\ , && \text{(D.14h)}
\end{aligned}$$

where the second to last inequality holds for $\gamma' \geq 13$.

2. Suppose that $W(r(\ell)) \leq \frac{\gamma' w_0}{\alpha}$. Now we need to show that $|\{i \in [m] : r_i \geq r_i(\ell) - 1\}| \leq \gamma\alpha w_0$ holds asymptotically almost surely. Note that

$$\begin{aligned}
\Pr[\mathbf{r}_i \geq r_i(\ell) - 1] &= \beta \sum_{j=r_i(\ell)-1}^{t} \alpha^{-j} && \text{(D.15a)}\\
&= \frac{\alpha - 1}{1 - \alpha^{-t}}\left(\frac{\alpha^{-r_i(\ell)+2} - \alpha^{-t}}{\alpha - 1}\right) && \text{(D.15b)}\\
&= \frac{\alpha^{-r_i(\ell)+2} - \alpha^{-t}}{1 - \alpha^{-t}} && \text{(D.15c)}\\
&= \frac{\alpha^{t-r_i(\ell)+2} - 1}{\alpha^{t} - 1} && \text{(D.15d)}\\
&\leq \frac{\alpha^{t-r_i(\ell)+2}}{\alpha^{t}/2} = 2\alpha^{2-r_i(\ell)}\ . && \text{(D.15e)}
\end{aligned}$$

Similar to the previous case, let $Q_{\mathbf{r}}(\ell) = \{i \in [m] \mid \mathbf{r}_i \geq r_i(\ell) - 1\}$. We can upper-bound the expected cardinality of $Q_{\mathbf{r}}(\ell)$ by

$$\mathbb{E}\left[|Q_{\mathbf{r}}(\ell)|\right] = \sum_{i \in [m]} \Pr[\mathbf{r}_i \geq r_i(\ell) - 1] \leq 2\alpha^2 W(r(\ell)) \leq 2\gamma' \alpha w_0 \ . \tag{D.16}$$

Again, we apply the Chernoff bound in Theorem D.2.1 and conclude that

$$\begin{aligned} \Pr\left[|Q_{\mathbf{r}}(\ell)| \geq \gamma\alpha w_0\right] &\leq \Pr\left[|Q_{\mathbf{r}}(\ell)| - \mathbb{E}\left[|Q_{\mathbf{r}}(\ell)|\right] \geq \gamma\alpha w_0 - 2\gamma'\alpha w_0\right] \\ &\leq \exp\left(-\frac{(\gamma - 2\gamma')^2(\alpha w_0)^2}{4\gamma'\alpha w_0 + (\gamma - 2\gamma')\alpha w_0}\right) \qquad \text{(D.17)} \\ &\leq \exp(-\alpha w_0) \\ &\leq L^{-2} \end{aligned}$$

where the second to last inequality holds for γ sufficiently larger than γ', say $\gamma \geq 5\gamma'$.

A union bound argument over all vectors in $\{r(\ell) : \ell \in [L]\}$ for both cases shows that for $\gamma' \geq 13$ and $\gamma \geq 5\gamma'$ there exists a choice of $r = (r_1, \ldots, r_m)$ such that both implications D.9a and D.9b hold. □

D.3.2 Graph Closure

A key concept in our work will be that of a *closure* of a vertex set, which seems to have originated in [AR03; ABRW04]. Intuitively, for an expander graph G, the closure of $T \subseteq V(G)$ is a suitably small set S that contains T such that $G \setminus S$ is an expander. In order to have a definition that makes sense for both expanders and bipartite expanders, we define $V_{\mathsf{exp}}(G)$ to be the set of vertices of G that expand, that is, if $G = (V, E)$ is an expander then $V_{\mathsf{exp}}(G) = V$, and if $G = (U \dot\cup V, E)$ is a bipartite expander then $V_{\mathsf{exp}}(G) = U$.

Definition D.3.2 (Closure)**.** For an expander graph G and vertex sets $S \subseteq V_{\mathsf{exp}}(G)$ and $U \subseteq V(G)$, we say that the set S is (U, r, ν)-*contained* if $|S| \leq r$ and $\left|\partial(S) \setminus U\right| < \nu \cdot |S|$.

For any expander graph G and any set $T \subseteq V_{\mathsf{exp}}(G)$ of size $|T| \leq r$, we will let $\mathrm{closure}_{r,\nu}(T)$ denote an arbitrary but fixed maximal set such that $T \subseteq \mathrm{closure}_{r,\nu}(T) \subseteq V_{\mathsf{exp}}(G)$ and $\mathrm{closure}_{r,\nu}(T)$ is $(N(T), r, \nu)$-contained.

Note that the closure of any set T of size $|T| \leq r$ as defined above does indeed exist, since T itself is $(N(T), r, \nu)$-contained.

Lemma D.3.3. *Suppose that* G *is an* (r, Δ, c)*-boundary expander and that* $T \subseteq V_{\mathsf{exp}}(G)$ *has size* $|T| \leq k \leq r$*. Then* $|\mathrm{closure}_{r,\nu}(T)| < \frac{k\Delta}{c-\nu}$.

Proof. By definition we have that $\left|\partial(\mathrm{closure}_{r,\nu}(T)) \setminus N(T)\right| < \nu \cdot |\mathrm{closure}_{r,\nu}(T)|$. Furthermore, since $|\mathrm{closure}_{r,\nu}(T)| \leq r$ by definition, we can use the expansion property of the graph to derive the inequality $\left|\partial(\mathrm{closure}_{r,\nu}(T)) \setminus N(T)\right| \geq |\partial(\mathrm{closure}_{r,\nu}(T))| - |N(T)| \geq c \cdot |\mathrm{closure}_{r,\nu}(T)| - k\Delta$. Note that we also use the fact that the neighbourhood of T is of size at most $k\Delta$. The conclusion follows by combining both statements. □

Suppose G is an excellent boundary expander and that $T \subseteq V_{\mathsf{exp}}(G)$ is not too large. Then Lemma D.3.3 shows that the closure of T is not much larger. And if the closure is not too large, then after removing the closure and its neighbourhood from the graph we are still left with a decent expander, a fact which will play a key role in the technical arguments in later sections. The following lemma makes this intuition precise.

Lemma D.3.4. *For* G *an* (r, Δ, c)*-boundary expander, let* $T \subseteq V_{\mathsf{exp}}(G)$ *be such that* $|T| \leq r$ *and* $|\mathrm{closure}_{r,\nu}(T)| \leq r/2$*, let* $G' = G \setminus \left(\mathrm{closure}_{r,\nu}(T) \cup N(\mathrm{closure}_{r,\nu}(T))\right)$ *and* $V_{\mathsf{exp}}(G') = V_{\mathsf{exp}}(G) \cap V(G')$*. Then any set* $S \subseteq V_{\mathsf{exp}}(G')$ *of size* $|S| \leq r/2$ *satisfies* $|\partial_{G'}(S)| \geq \nu|S|$.

Proof. Suppose the set $S \subseteq V_{\mathsf{exp}}(G')$ is of size $|S| \leq r/2$ and does not satisfy $|\partial_{G'}(S)| \geq \nu|S|$. Since $\mathrm{closure}_{r,\nu}(T)$ is also of size at most $r/2$, we have that the set $(\mathrm{closure}_{r,\nu}(T) \cup S)$ is $(N(T), r, \nu)$-contained in the graph G. But this contradicts the maximality of $\mathrm{closure}_{r,\nu}(T)$. □

D.4 Lower Bounds for Weak Graph FPHP Formulas

We now proceed to establish lower bounds on the length of resolution refutations of functional pigeonhole principle formulas defined over bipartite graphs. We write $G = (V_P \dot\cup V_H, E)$ to denote the graph over which the formulas are defined and $\mathcal{M}$ to denote the set of partial matchings on G (also viewed as partial mappings of V_P to V_H). Let us start by making more precise some of the technical notions discussed in the introduction (which were originally defined in [Raz01]).

For a clause C and a pigeon i we denote the set of holes j with the property that C is satisfied if i is matched to j by

$$N_C(i) = \{j \in \underset{H}{V} \mid e = \{i, j\} \in E \text{ and } \rho_{\{e\}}(C) = 1\} \qquad \text{(D.18)}$$

and we define the i*th pigeon degree* $\deg_C(i)$ *of* C as

$$\deg_C(i) = |N_C(i)| \ . \qquad \text{(D.19)}$$

We think of a pigeon i with large $\deg_C(i)$ as a pigeon on which the derivation has not made any significant progress up to the point of deriving C, since the clause rules out very few holes. The pigeons with high enough pigeon degree in a clause are the *heavy pigeons* of the clause as defined next.

Definition D.4.1 (Pigeon weight, pseudo-width and (w_0, d)-axioms)**.** Let C be a clause and let $d = (d_1, \dots, d_m)$ and $\delta = (\delta_1, \dots, \delta_m)$ be two vectors of positive integers such that d is elementwise greater than δ. We say that pigeon i is d-*super-heavy for* C if $\deg_C(i) \geq d_i$ and that pigeon i is (d, δ)-*heavy for* C if $\deg_C(i) \geq d_i - \delta_i$. When d and δ are understood from context, which is most often the case, we omit the parameters and just refer to *super-heavy* and *heavy* pigeons. Pigeons that are not heavy are referred to as *light pigeons*. The set of pigeons that are super-heavy for C is denoted by

$$P_d(C) = \{i \in [m] \mid \deg_C(i) \geq d_i\}$$

and the set of pigeons that are heavy for C is denoted by

$$P_{d,\delta}(C) = \{i \in [m] \mid \deg_C(i) \geq d_i - \delta_i\} \ .$$

The *pseudo-width* of C is the number of heavy pigeons in C and the pseudo-width of a resolution refutation π, denoted by $w_{d,\delta}(\pi)$, is $\max_{C \in \pi} w_{d,\delta}(C)$. Finally, we will refer to clauses C with precisely w_0 super-heavy pigeons, i.e., such that $|P_d(C)| = w_0$, as (w_0, d)-*axioms*.

Note that according to Definition D.4.1 super-heavy pigeons are also heavy. Making the connection back to our informal discussion in the introduction, the “fake axioms” mentioned there are nothing other than (w_0, d)-axioms.

Now that we have all the notions needed, let us give a detailed proof outline. Given a short resolution refutation π of the formula FPHP(G), we use the Filter lemma (Lemma D.3.1) to get a filter vector $d = (d_1, \dots, d_m)$ such that each clause either has many super-heavy pigeons or there are not too many heavy pigeons (for an appropriately chosen vector δ). Clearly, clauses that fall into the second case of the filter lemma have bounded pseudo-width. On the other hand, clauses in the first case may have very large pseudo-width. In order to obtain a proof of low pseudo-width, these clauses are strengthened to (w_0, d)-axioms and added to a special set A. This then gives a refutation π' that refutes the formula FPHP(G) $\cup$ A in bounded pseudo-width. The following lemma summarizes the upper bound on pseudo-width that we obtain.

Lemma D.4.2. *Let* $G = (V_P \dot\cup V_H, E)$ *be a bipartite graph with* $|V_P| = m$ *and* $|V_H| = n$; *let* π *be a resolution refutation of* FPHP(G); *let* $w_0, \alpha \in [m]$ *be such* *that* $w_0 > \log L(\pi)$ *and* $w_0 \geq \alpha^2 \geq 4$*, and let* $\delta = (\delta_1, \ldots, \delta_m)$ *be defined by* $\delta_i = \frac{\deg_G(i) \log \alpha}{\log m}$*. Then there exists an integer vector* $d = (d_1, \ldots, d_m)$*, with* $\delta_i < d_i \leq \deg_G(i)$ *for all* $i \in V_P$*, a set of* (w_0, d)*-axioms* A *with* $|A| \leq L(\pi)$*, and a resolution refutation* π' *of* $\mathrm{FPHP}(G) \cup A$ *such that* $w_{d,\delta}(\pi') = O(\alpha \cdot w_0)$*.*

As mentioned above, this upper bound is a straightforward application of Lemma D.3.1. We defer the formal proof to Section D.4.2. What we will need from Lemma D.4.2 is that a resolution refutation of $\mathrm{FPHP}(G)$ in length less than 2^{w_0} can be transformed into a refutation of $\mathrm{FPHP}(G) \cup A$ in pseudo-width at most $O(\alpha \cdot w_0)$.

The second step in the proof is to show that any resolution refutation π of $\mathrm{FPHP}(G) \cup A$ requires large pseudo-width. The high-level idea is to define a progress measure on clauses $C \in \pi$ by counting the number of matchings on $P_{d,\delta}(C)$ that do not satisfy C. We then show that in order to increase this progress measure we need large pseudo-width. The following lemma states the pseudo-width lower bound.

Lemma D.4.3. *Let* $\xi \leq 1/4$ *and* $m, n, r, \Delta \in \mathbb{N}$*; let* $G = (V_P \dot\cup V_H, E)$ *with* $|V_P| = m$ *and* $|V_H| = n$ *be an* $(r, \Delta, (1-2\xi)\Delta)$*-boundary expander, and let* $\delta = (\delta_1, \ldots, \delta_m)$ *be defined by* $\delta_i = 4\deg_G(i)\xi$*. Suppose that* $d = (d_1, \ldots, d_m)$ *is an integer vector such that* $\delta_i < d_i \leq \deg_G(i)$ *for all* $i \in V_P$*. Let* w_0 *be an arbitrary parameter and* A *be an arbitrary set of* (w_0, d)*-axioms with* $|A| \leq (1+\xi)^{w_0}$*. Then every resolution refutation* π *of* $\mathrm{FPHP}(G) \cup A$ *must satisfy* $w_{d,\delta}(\pi) \geq r\xi/4$*.*

In one sentence, the lemma states that if the set of "fake axioms" A is not too large, then resolution requires large pseudo-width to refute $\mathrm{FPHP}(G) \cup A$. Note that this lemma holds for any filter vector and not just for the one obtained from Lemma D.4.2.

In order to prove Lemma D.4.3, we wish to define a progress measure on clauses that indicates how close the derivation is to refuting the formula (i.e., it should be small for axiom clauses but large for contradiction). A first attempt would be to define the progress of a clause C as the number of ruled-out matchings (i.e., matchings that do not satisfy C) on the pigeons mentioned by C. This definition does not quite work, but we can refine it by counting matchings less carefully. Namely, if for a pigeon i there are more than $\deg_G(i) - d_i + \delta_i/4$ holes to which it can be mapped without satisfying C, then we think of C as ruling out *all holes* for this pigeon. Since the pigeon degree of a light pigeon i is at most $d_i - \delta_i$, such a pigeon will certainly have at least $\deg_G(i) - d_i + \delta_i \geq \deg_G(i) - d_i + \delta_i/4$ holes to which it can be mapped, and the "lossy counting" will ensure that all holes are considered as ruled out.

We realize this "lossy counting" through a linear space Λ, in which each partial matching φ is associated with a subspace $\lambda(\varphi)$. Roughly speaking, the progress $\lambda(C)$ of a clause C is then defined to be the span of all partial matchings that are ruled out by C. We design the association between matchings and subspaces so that the contradictory empty clause $\perp$ has $\lambda(\perp) = \Lambda$ but so that the span of all the axioms $\mathrm{span}(\{\lambda(A) \mid A \in \mathrm{FPHP}(G) \cup \mathcal{A}\})$ is a proper subspace of Λ. This implies that in a refutation π of $\mathrm{FPHP}(G) \cup \mathcal{A}$ there must exist a resolution step deriving a clause C from clauses C_0 and C_1 such that the linear space of the resolvent $\lambda(C)$ is not contained in $\mathrm{span}(\lambda(C_0), \lambda(C_1))$. But the main technical lemma of this section (Lemma D.4.10) says that for any derivation in low pseudo-width the linear space of the resolvent is contained in the span of the linear spaces of the clauses being resolved. Hence, in order for π to be a refutation it must contain a clause with large pseudo-width, and this establishes Lemma D.4.3.

So far our argument follows that of Razborov very closely, but it turns out we cannot realize this proof idea if we only keep track of heavy and light pigeons. Let us attempt a proof of the claim in Lemma D.4.10 that low-width resolution steps cannot increase the span to illustrate what the problem is. The interesting case is when there is a pigeon i that is heavy for C_0 or C_1 but not for their resolvent C. Then, following Razborov, for any matching φ on the heavy pigeons of C that fails to satisfy C, we need to be able to extend φ in at least $\deg_G(i) - d_i + \delta_i/4$ different ways to a matching including also pigeon i that falsifies either C_0 or C_1. If this can be done, then we think of C_0 and C_1 as together ruling out (essentially) all holes for i, and the linear space associated with C will be contained in the span of the spaces for C_0 and C_1. The problem, though, is that φ may send all heavy pigeons to the neighbourhood of pigeon i. In this scenario, there might be very few holes, or even no holes, to which i can be mapped when extending φ, and even our lossy counting will not be able to pick up enough holes for the argument to go through. We resolve this problem by not only considering the heavy pigeons but a larger set of *relevant* pigeons including all pigeons i' that can become overly constrained when some matching on the heavy pigeons shrinks the neighbourhood of i' too much. Formally, the *closure* of the set of heavy pigeons, as defined in Definition D.3.2, is the notion that we need.

D.4.1 Formal Statements of Graph FPHP Formula Lower Bounds

Deferring the proofs of all technical lemmas for now, let us state our lower bounds for graph FPHP formulas and see how they follow from Lemmas D.4.3 and D.4.2 above.

Theorem D.4.4. *Let* $m = |U|$ *and* $n = |V|$ *and suppose that* $G = (U \dot\cup V, E)$ *is an* $(r, \Delta, (1 - \frac{\log \alpha}{2 \log m})\Delta)$*-boundary expander for* $\alpha \in [m]$ *such that* $8 \le \frac{\alpha^3}{\log \alpha} = o(\frac{r}{\log m})$*. Then resolution requires length* $\exp\left(\Omega\left(\frac{r \log^2 \alpha}{\alpha \log^2 m}\right)\right)$ *to refute* FPHP(G).

As promised in Section D.3, let us briefly discuss the parameter α. Note that, on the one hand, the larger α is, the more relaxed we can be with respect to the expansion requirements, and hence the set of formulas to which the lower bound applies becomes larger. On the other hand, the strength of the lower bound deteriorates quickly with α. Hence, we need to choose α carefully to find a good compromise between these two concerns.

Proof of Theorem D.4.4. Let $\xi = \frac{\log \alpha}{4 \log m}$ and let $w_0 = \frac{\varepsilon_0 r \xi}{\alpha}$ for some small enough $\varepsilon_0 > 0$. We note that the choice of parameters and the condition on α ensure that $4 \le \alpha^2 \le w_0$. Furthermore, in terms of ξ, the graph G is an $(r, \Delta, (1 - 2\xi)\Delta)$-boundary expander.

We proceed by contradiction. Suppose π is a resolution refutation with $L(\pi) < 2^{\varepsilon' w_0 \xi}$ for a small enough constant $\varepsilon' > 0$. Applying Lemma D.4.2 we get a set of (w_0, d)-axioms A with $|A| \le L(\pi)$ and a resolution refutation π' of $\mathrm{FPHP}(G) \cup A$ such that $w_{d,\delta}(\pi') \le K\alpha w_0$ for some large enough constant K.

Note that $|A| \le L(\pi) < 2^{\varepsilon' w_0 \xi} \le (1 + \xi)^{w_0}$ for $\varepsilon' < 1/2$. Applying Lemma D.4.3 to π' yields a pseudo-width lower bound of $r\xi/4$. We conclude that

$$r\xi/4 \le w_{d,\delta}(\pi') \le K\alpha w_0 = \varepsilon_0 K r \xi \ . \tag{D.20}$$

Choosing $\varepsilon_0 < \frac{1}{4K}$ yields a contradiction. □

The following corollary summarizes our claims for random graphs.

Corollary D.4.5. *Let* m *and* n *be positive integers and let* $\Delta : \mathbb{N}^+ \longrightarrow \mathbb{N}^+$ *and* $\varepsilon : \mathbb{N}^+ \longrightarrow [0, 1]$ *be any monotone functions of* n *such that* $n < m \le n^{(\varepsilon/16)^2 \log n}$ *and* $n \ge \Delta \ge \left(\frac{16 \log m}{\varepsilon \log n}\right)^2$*. Then asymptotically almost surely resolution requires length* $\exp\left(\Omega\left(n^{1-\varepsilon}\right)\right)$ *to refute* FPHP(G) *for* $G \sim \mathcal{G}(m, n, \Delta)$.

Proof. Let us assume that $n^{(\varepsilon/16)^2 \log n}$ and $\left((16 \log m)/(\varepsilon \log n)\right)^2$ are integers. Observe that if $G \sim \mathcal{G}(m, n, \Delta)$ for $\Delta > \left((16 \log m)/(\varepsilon \log n)\right)^2$, then we can sample a random subgraph $G' \sim \mathcal{G}\left(m, n, ((16 \log m)/(\varepsilon \log n))^2\right)$ by choosing a random subset of appropriate size of each neighbourhood of a left vertex (and applying a restriction zeroing out the other edges). Hence, we can restrict our attention to the case where $\Delta = \left((16 \log m)/(\varepsilon \log n)\right)^2$. Also, it is sufficient to prove the claim for $m = n^{(\varepsilon/16)^2 \log n}$, since choosing

m smaller can only make the formula less constrained and hence makes the lower bound easier to obtain.

We want to apply Lemma D.2.2 for $\chi = \alpha = n^{\varepsilon/4}$ and $\xi = \frac{\log \alpha}{4 \log m}$. In order to do so, we need to verify the inequalities

$$\xi < 1/2 \ , \tag{D.21a}$$

$$\xi \ln \chi \geq 2 \ , \tag{D.21b}$$

$$\xi \Delta \ln \chi \geq 4 \ln m \ . \tag{D.21c}$$

For D.21a we observe that $\xi = \frac{16}{\varepsilon \log n}$ and since $n < n^{(\varepsilon/16)^2 \log n}$ we see that $\frac{1}{\log n} < \left(\frac{\varepsilon}{16}\right)^2$. Hence, the first condition holds for n large enough. To check D.21b, we compute

$$\xi \ln \chi = \frac{16}{\varepsilon \log n} \frac{\varepsilon \ln n}{4} \geq 2 \ . \tag{D.22}$$

For (D.21c), we observe that $\Delta = \log m$ and hence

$$\xi \Delta \ln \chi = \frac{4}{\log e} \log m = 4 \ln m \ . \tag{D.23}$$

We conclude that asymptotically almost surely, $G \sim \mathcal{G}(m, n, \Delta)$ is an $\left(n^{1-\varepsilon/2}, \Delta, (1-2\xi)\Delta\right)$-boundary expander. Theorem D.4.4 then gives a length lower bound of $\exp\left(\Omega\left(n^{1-\varepsilon}\right)\right)$, as required. □

The following two corollaries are simple consequences of Corollary D.4.5, optimizing for different parameters. The first corollary gives the strongest lower bounds, while the second minimizes the degree.

Corollary D.4.6. *Let m, n be such that $m \leq n^{o(\log n)}$. Then asymptotically almost surely resolution requires length $\exp\left(\Omega\left(n^{1-o(1)}\right)\right)$ to refute $\mathrm{FPHP}(G)$ for $G \sim \mathcal{G}\left(m, n, \log m\right)$.*

Proof. Let $m = n^{f(n)}$, where $f(n) = o(\log n)$. Applying Corollary D.4.5 for $\varepsilon = 16\sqrt{\frac{f(n)}{\log n}} = o(1)$ we get the desired statement. □

Corollary D.4.7 (Restatement of Theorem D.1.3)**.** *Let k and n be positive integers and let $m = n^k$ and $\varepsilon \in \mathbb{R}^+$. Then asymptotically almost surely resolution requires length $\exp\left(\Omega\left(n^{1-\varepsilon}\right)\right)$ to refute $\mathrm{FPHP}(G)$ for $G \sim \mathcal{G}\left(m, n, \left(\frac{16k}{\varepsilon}\right)^2\right)$.*

Proof. We appeal to Corollary D.4.5 with $\Delta = \left(\frac{16k}{\varepsilon}\right)^2$, $m = n^k$ and ε constant. A short calculation shows that all conditions are met. □

Our final corollary shows that we can get meaningful lower bounds even for a weakly exponential number of pigeons. Unfortunately, the statement does not hold for random graphs.

Corollary D.4.8. *Let $\kappa < 3/2 - \sqrt{2}$ and $\varepsilon > 0$ be constant and n be integer. Then there is a family of explicitly constructible graphs G with $m = 2^{\Omega(n^{\kappa})}$ and left degree $O(\log^{1/\sqrt{\kappa}}(m))$ such that resolution requires length $\exp\big(\Omega\big(n^{1-2\sqrt{\kappa}(2-\sqrt{\kappa})-\varepsilon}\big)\big)$ to refute* $\mathrm{FPHP}(G)$*.*

Proof. Let G be the graph from Corollary D.2.4 with $\nu = \frac{2\sqrt{\kappa}}{1-2\sqrt{\kappa}}$. An appeal to Theorem D.4.4 using the graph G yields the desired lower bound. □

D.4.2 A Pseudo-Width Upper Bound for Graph FPHP Formulas with Extra Axioms

Let us now prove Lemma D.4.2. For this proof, let us identify V_P with $[m]$. For every clause C in the refutation π, let $r(C) = (r_1(C), \ldots, r_m(C))$ be the vector where each coordinate is given by

$$r_i(C) = \left\lfloor \frac{\deg_G(i) - \deg_C(i)}{\delta_i} \right\rfloor + 1 \ . \tag{D.24}$$

We apply the filter lemma (Lemma D.3.1) to the set of vectors $\{r(C) \mid C \in \pi\}$. Denote by $r = (r_1, \ldots, r_m)$ a vector as guaranteed to exist by Lemma D.3.1. Let

$$d_i = \deg_G(i) - \lceil \delta_i r_i \rceil + 1 \ . \tag{D.25}$$

A short calculation establishes that d_i is the smallest integer such that $\left\lfloor \frac{\deg_G(i)-d_i}{\delta_i} \right\rfloor + 1 \le r_i$.

Note that every pigeon $i \in [m]$ such that $r_i(C) \le r_i$ is super-heavy for C. Also, every heavy pigeon of a clause C satisfies that $r_i(C) \le r_i + 1$.

To obtain a refutation π' that satisfies the conclusions of the lemma, we consider every clause $C \in \pi$ and either add a strengthening of C to the (w_0, d)-axiom set A or conclude that the pseudo-width of C is small enough that the clause can stay in π'. More concretely, we make a case distinction whether $r(C)$ satisfies case 1 of Lemma D.3.1 or only case 2. In one case C can be strengthened to a (w_0, d)-axiom, while in the other the pseudo-width of C is bounded:

1. C satisfies $\big|\{i \in [m] \mid r_i(C) \le r_i\}\big| \ge w_0$: As every pigeon $i \in [m]$ with $r_i(C) \le r_i$ also satisfies $\deg_C(i) \ge d_i$, we can strengthen this clause to a (w_0, d)-axiom and add it to A. This reduces the pseudo-width of this clause to w_0.

2. C satisfies $\left|\{i \in [m] \mid r_i(C) \le r_i + 1\}\right| \le O(\alpha \cdot w_0)$: As every heavy pigeon always satisfies $r_i(C) \le r_i + 1$, the pseudo-width of C is $O(\alpha \cdot w_0)$.

This concludes the proof as $|A| \le L(\pi)$ and the pseudo-width of π' is $O(\alpha \cdot w_0)$ by construction.

D.4.3 A Pseudo-Width Lower Bound for Graph FPHP Formulas with Extra Axioms

We continue to the proof of Lemma D.4.3. Using Definition D.3.2, we define the set of *relevant* pigeons of a clause C as

$$\mathrm{closure}(C) = \mathrm{closure}_{r,(1-3\xi)\Delta}(P_{d,\delta}(C)) \enspace , \tag{D.26}$$

where $P_{d,\delta}(C)$ denotes the set of (d, δ)-heavy pigeons for C as defined in Definition D.4.1. By definition, the closure of a set T contains T itself but is only defined if $|T| \le r$. However, if $\left|P_{d,\delta}(C)\right| \ge r \ge r\xi/4$ then we already have the lower bound claimed in the lemma, and so we may assume that the closure is well defined for all clauses in the refutation π. This implies, in particular, that for every clause $C \in \pi$ we have $P_{d,\delta}(C) \subseteq \mathrm{closure}(C)$.

Let us next construct the linear space Λ and describe how matchings are mapped into it. Fix a field $\mathbb{F}$ of characteristic 0 and for each pigeon $i \in V_P$ let Λ_i be a linear space over $\mathbb{F}$ of dimension $\deg_G(i) - d_i + \delta_i/4$. Let Λ be the tensor product $\Lambda = \bigotimes_{i \in V_P} \Lambda_i$ and denote by $\lambda_i : V_H \mapsto \Lambda_i$ a function with the property that any subset of holes $J \subseteq V_H$ of size at least $\dim(\Lambda_i)$ spans Λ_i. In other words, for J as above we have that $\Lambda_i = \mathrm{span}(\lambda_i(j) : j \in J)$. This is how we will realize the idea of "lossy counting." For $J \subseteq V_H$ such that $|J| \le \dim(\Lambda_i)$ we have exact counting $\dim(\mathrm{span}(\{\lambda_i(j) \mid j \in J\})) = |J|$, but when $|J| > \dim(\Lambda_i)$ gets large enough we have $\dim(\mathrm{span}(\{\lambda_i(j) \mid j \in J\})) = \dim(\Lambda_i)$.

In order to map functions $V_P \mapsto V_H$ into Λ, we define $\lambda : V_H^{V_P} \mapsto \Lambda$ by $\lambda(j_1, \ldots, j_m) = \bigotimes_{i \in V_P} \lambda_i(j_i)$, where will we abuse notions slightly in that we identify a vector with the 1-dimensional space spanned by this vector. For a partial function $\varphi : V_P \mapsto V_H$, we let $\lambda(\varphi)$ be the span of all total extensions of φ (not necessarily matchings), or equivalently

$$\lambda(\varphi) = \bigotimes_{i \in \mathrm{dom}(\varphi)} \lambda_i(\varphi_i) \otimes \bigotimes_{i \notin \mathrm{dom}(\varphi)} \Lambda_i \enspace . \tag{D.27}$$

Recall that $\mathcal{M}$ is the set of all partial matchings on the graph G and that we interchangeably think of partial matchings as partial functions $\varphi : V_P \to V_H$ or as Boolean assignments ρ_φ as defined in D.4. For each clause C, we are interested in the partial matchings $\varphi \in \mathcal{M}$ with domain $\mathrm{dom}(\varphi) = \mathrm{closure}(C)$ such that ρ_φ does not satisfy C. We refer to the set of such matchings as the *zero space* of C and denote it by

$$Z(C) = \{\varphi \in \mathcal{M} \mid \mathrm{dom}(\varphi) = \mathrm{closure}(C) \land \rho_\varphi(C) \neq 1\} \ . \tag{D.28}$$

We associate C with the linear space

$$\lambda(C) = \mathrm{span}(\{\lambda(\varphi) \mid \varphi \in Z(C)\}) \ . \tag{D.29}$$

Note that contradiction is mapped to Λ, i.e., $\lambda(\bot) = \Lambda$.

We assert that the span of the axioms $\mathrm{span}(\{\lambda(A) \mid A \in \mathrm{FPHP}(G) \cup \mathcal{A}\})$ is a proper subspace of Λ.

Lemma D.4.9. *If* $|\mathcal{A}| \leq (1+\xi)^{w_0}$*, then* $\mathrm{span}(\{\lambda(A) \mid A \in \mathrm{FPHP}(G) \cup \mathcal{A}\}) \subsetneq \Lambda$.

Accepting this claim without proof for now, this implies that in π there is some resolution step deriving C from C_0 and C_1 where the subspace of the resolvent is not contained in the span of the subspaces of the premises, or in other words $\lambda(C) \not\subseteq \mathrm{span}(\lambda(C_0), \lambda(C_1))$. Our next lemma, which is the heart of the argument, says that this cannot happen as long as the closures of the clauses are small.

Lemma D.4.10. *Let* C *be derived from* C_0 *and* C_1*. If*

$$\max\{|\mathrm{closure}(C_0)|, |\mathrm{closure}(C_1)|, |\mathrm{closure}(C)|\} \leq r/4 \ ,$$

then $\lambda(C) \subseteq \mathrm{span}(\lambda(C_0), \lambda(C_1))$.

Since contradiction cannot be derived while the closure is of size at most $r/4$, any refutation π must contain a clause C with $|\mathrm{closure}(C)| > r/4$. But then Lemma D.3.3 implies that C has pseudo-width at least $r\xi/4$, and Lemma D.4.3 follows. All that remains for us is to establish Lemmas D.4.10 and D.4.9.

Proof of Lemma D.4.9. We need to show that the axioms $\mathrm{FPHP}(G) \cup \mathcal{A}$ do not span all of Λ. We start with the axioms in $\mathrm{FPHP}(G)$.

Let A be pigeon axiom P^i as in (D.1a) or a functionality axiom $F^i_{j,j'}$ as in (D.1c). Note that i is a heavy pigeon for A. Clearly, there are no pigeon-to-hole assignments for pigeon i that do not satisfy A. Thus there are no matchings on $\mathrm{closure}(A)$ that do not satisfy A. We conclude that $\lambda(A) = \emptyset$. If instead A is a hole axiom $H^{i,i'}_j$ as in (D.1b), then we can observe that $\deg_G(i) - 1 \geq d_i - \delta_i$ since $\delta_i = 4\xi \deg_G(i) \geq 2\xi\Delta \geq 1$ (by boundary expansion). This implies that A has two heavy pigeons. Observe that there are no matchings on these two pigeons that do not satisfy A. Thus $Z(A) = \emptyset$ and we conclude that $\lambda(A) = \emptyset$.

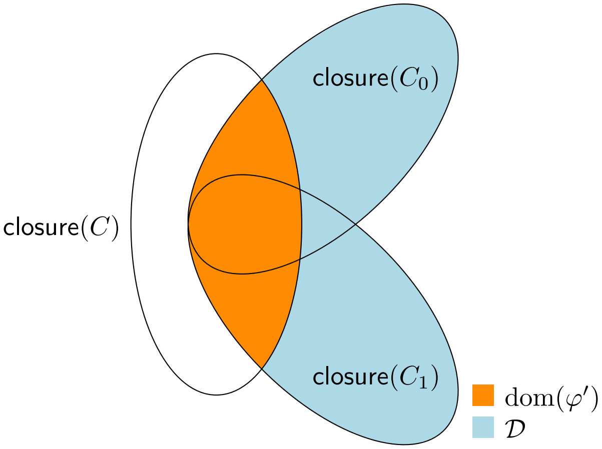


Figure D.1: Depiction of relations between $\mathrm{closure}(C), \mathrm{closure}(C_i), i = 1, 2, \mathrm{dom}(\varphi')$ and $\mathcal{D}$ in proof of Lemma D.4.10.

Now consider the (w_0, d)-axioms in $\mathcal{A}$. We wish to show that any $A \in \mathcal{A}$ can only span a very small fraction of Λ. We can estimate the the number of dimensions $\lambda(A)$ spans by

$$\dim \lambda(A) \leq \prod_{i \notin P_d(A)} \dim \Lambda_i \cdot \prod_{i \in P_d(A)} (\deg_G(i) - d_i) \ . \tag{D.30}$$

Hence the fraction of the space Λ that A may span is bounded by

$$\frac{\dim \lambda(A)}{\dim \Lambda} \leq \prod_{i \in P_d(A)} \frac{\deg_G(i) - d_i}{\deg_G(i) - d_i + \delta_i/4} \leq (1 - \xi)^{w_0} \ . \tag{D.31}$$

As $|\mathcal{A}| \leq (1 + \xi)^{w_0}$ we can conclude that not all of Λ is spanned by the axioms. □

Proof of Lemma D.4.10. For conciseness of notation, let us write $S_{01} = \mathrm{closure}(C_0) \cup \mathrm{closure}(C_1)$ and $S = \mathrm{closure}(C)$. In order to establish the lemma, we need to show for all $\varphi \in Z(C)$ that

$$\lambda(\varphi) \subseteq \mathrm{span}(\lambda(C_0), \lambda(C_1)) \ . \tag{D.32}$$

To comprehend the argument that will follow below, it might be helpful to refer to the illustration in Figure D.1.

Denote by φ' the restriction of φ to the domain $S \cap S_{01}$ and note that C is not satisfied under $\rho_{\varphi'}$. Also, observe that if a matching η extends a matching η', then $\lambda(\eta)$ is contained in $\lambda(\eta')$. This is so since for any pigeon $i \in \mathrm{dom}(\eta) \setminus \mathrm{dom}(\eta')$ we have from D.27 that η' picks up the whole subspace Λ_i while η only gets a single vector. Thus, if we can show that $\lambda(\varphi') \subseteq \mathrm{span}(\lambda(C_0), \lambda(C_1))$, then we are done as φ extends φ' and hence $\lambda(\varphi) \subseteq \lambda(\varphi')$.

Let $\mathcal{D} = S_{01} \setminus S$ and denote by $\mathcal{M}_{\mathcal{D}}$ the set of matchings that extend φ' to the domain $\mathcal{D}$ and do not satisfy C. Since each matching $\psi \in \mathcal{M}_{\mathcal{D}}$ fails to satisfy C, by the soundness of the resolution rule we have that it also fails to satisfy either C_0 or C_1. Assume without loss of generality that ψ does not satisfy C_0 and denote by ψ' the restriction of ψ to the domain of $\mathrm{closure}(C_0)$. From D.28 we see that $\psi' \in Z(C_0)$ and therefore $\lambda(\psi) \subseteq \lambda(\psi') \subseteq \lambda(C_0)$.

So far we have argued that for all matchings $\psi \in \mathcal{M}_{\mathcal{D}}$ it holds that $\lambda(\psi) \subseteq \mathrm{span}(\lambda(C_0), \lambda(C_1))$. Let $\lambda(\mathcal{M}_{\mathcal{D}}) = \mathrm{span}(\lambda(\psi) \mid \psi \in \mathcal{M}_{\mathcal{D}})$. If we can show that the set of matchings $\mathcal{M}_{\mathcal{D}}$ is large enough for $\lambda(\mathcal{M}_{\mathcal{D}}) = \lambda(\varphi')$ to hold, then the lemma follows. In other words, we want to show that $\lambda(\mathcal{M}_{\mathcal{D}})$ projected to $\Lambda_{\mathcal{D}} = \bigotimes_{i \in \mathcal{D}} \Lambda_i$ spans all of the space $\Lambda_{\mathcal{D}}$.

To argue this, note first that $\mathcal{D}$ is completely outside the closure(C). Furthermore, by assumption we have $|\mathrm{closure}(C)| \leq r/4$ and $|\mathcal{D}| \leq |S_{01}| \leq r/2$. An application of Lemma D.3.4 now tells us that

$$|\partial_{G \setminus (\mathrm{closure}(C) \cup N(\mathrm{closure}(C)))}(\mathcal{D})| \geq (1 - 3\xi)\Delta|\mathcal{D}| \quad . \tag{D.33}$$

By an averaging argument, there must exist a pigeon $i_1 \in \mathcal{D}$ that has more than $(1 - 3\xi)\Delta$ unique neighbours in $\partial_{G \setminus (\mathrm{closure}(C) \cup N(\mathrm{closure}(C)))}(\mathcal{D})$. The same argument applied to $\mathcal{D} \setminus \{i_1\}$ show that some pigeon i_2 has more than $(1 - 3\xi)\Delta$ unique neighbours on top of the neighbours reserved for pigeon i_1. Iterating this argument, we derive by induction that for each pigeon $i \in \mathcal{D}$ we can find $(1-3\xi)\Delta$ distinct holes in $N(\mathcal{D})$. Since all pigeons in $\mathcal{D}$ are light in C, it follows that at most $d_i - \delta_i$ mappings of pigeon i can satisfy the clause C. Hence, there are at least

$$\begin{aligned}(1 - 3\xi)\Delta - (d_i - \delta_i) &\geq (1 - 3\xi)\deg_G(i) - d_i + 4\xi \deg_G(i) \\ &\geq \deg_G(i) - d_i + \delta_i/4 \end{aligned} \tag{D.34}$$

many holes to which each pigeon in $\mathcal{D}$ can be sent, independently of all other pigeons in $\mathcal{D}$, without satisfying C. As we have that $\dim(\Lambda_i) = \deg_G(i) - d_i + \delta_i/4$, we conclude that $\lambda(\mathcal{M}_{\mathcal{D}})$ projected to $\Lambda_{\mathcal{D}}$ spans the whole space. This concludes the proof of the lemma. □

D.5 Lower Bounds for Perfect Matching Principle Formulas

In this section, we show that the perfect matching principle formulas defined over even highly unbalanced bipartite graphs require exponentially long resolution refutations if the graphs are expanding enough.

Just as in [Raz04b], our proof is by an indirect reduction to the FPHP lower bound, and therefore there is a significant overlap in concepts and notation with Section D.4. However, since there are also quite a few subtle shifts in meaning, we restate all definitions in full below to make the exposition in this section self-contained and unambiguous.

We first review some useful notions from [Raz01]. Let $G = (V, E)$ denote the graph over which the formulas are defined. For a clause C and a vertex $v \in V(G)$, let the *clause-neighbourhood of* v *in* C, denoted by $N_C(v)$, be the vertices $u \in V(G)$ with the property that C is satisfied if v is matched to u, that is,

$$N_C(v) = \{u \in V \mid e = \{u, v\} \in E \text{ and } \rho_{\{e\}}(C) = 1\} \ . \tag{D.35}$$

For a set $V \subseteq V(G)$ let $N_C(V)$ be the union of the clause-neighbourhoods of the vertices in V, i.e., $N_C(V) = \bigcup_{v \in V} N_C(v)$ and let the *vth vertex degree of* C be

$$\deg_C(v) = |N_C(v)| \ . \tag{D.36}$$

We think of a vertex v with large degree $\deg_C(v)$ as a vertex on which the derivation has not made any progress up to the point of deriving C, since the clause rules out very few neighbours. The vertices with high enough vertex degree in a clause are the *heavy vertices* of the clause as defined next.

Definition D.5.1 (Vertex weight, pseudo-width and (w_0, d)-axioms)**.** Let $d = (d_1, \ldots, d_{m+n})$ and $\delta = (\delta_1, \ldots, \delta_{m+n})$ be two vectors such that d is elementwise greater than δ. We say that a vertex v is d-*super-heavy for* C if $\deg_C(v) \geq d_v$ and that vertex v is (d, δ)-*heavy for* C if $\deg_C(v) \geq d_v - \delta_v$. When d and δ are understood from context we omit the parameters and just refer to *super-heavy* and *heavy* vertices. Vertices that are not heavy are referred to as *light vertices*. The set of vertices that are super-heavy for C is denoted by

$$V_d(C) = \{v \in V \mid \deg_C(v) \geq d_v\} \tag{D.37}$$

and the set of heavy vertices for C is denoted by

$$V_{d,\delta}(C) = \{v \in V \mid \deg_C(v) \geq d_v - \delta_v\} \ . \tag{D.38}$$

The *pseudo-width* $w_{d,\delta}(C) = |V_{d,\delta}(C)|$ of a clause C is the number of heavy vertices in it, and the pseudo-width of a resolution refutation π is $w_{d,\delta}(\pi) = \max_{C \in \pi} w_{d,\delta}(C)$. We refer to clauses C with precisely w_0 super-heavy vertices as (w_0, d)-*axioms*.

To a large extent, the proof of the lower bounds for perfect matching formulas follows the general idea of the proof of Theorem D.4.4: given a short refutation we first apply the filter lemma to obtain a refutation of small pseudo-width; we then prove that in small pseudo-width contradiction cannot be derived and can thus conclude that no short refutation exists. In more detail, given a short resolution refutation π, we use the filter lemma (Lemma D.3.1) to get a filter vector $d = (d_1, \dots, d_{m+n})$ such that each clause either has many super-heavy vertices or not too many heavy vertices (for an appropriately chosen vector δ). Clearly, clauses that fall into the second case of the filter lemma have bounded pseudo-width. Clauses in the first case, however, may have very large pseudo-width. In order to obtain a proof of low pseudo-width, these latter clauses are strengthened to (w_0, d)-axioms and added to a special set A. This then gives a refutation π' that refutes the formula $\mathrm{PM}(G) \cup A$ in bounded pseudo-width as stated in the next lemma.

Lemma D.5.2. *Let* $G = (V_L \dot\cup V_R, E)$ *be a bipartite graph with* $|V_L| = m$ *and* $|V_R| = n$*; let* π *be a resolution refutation of* $\mathrm{PM}(G)$*; let* $w_0, \alpha \in [m+n]$ *be such that* $w_0 > \log L(\pi)$ *and* $w_0 \geq \alpha^2 \geq 4$*, and let* $\delta = (\delta_1, \dots, \delta_{m+n})$ *be defined by* $\delta_v = \frac{\deg_G(v) \log \alpha}{\log(m+n)}$ *for* $v \in V(G)$*. Then there exists an integer vector* $d = (d_1, \dots, d_{m+n})$*, with* $\delta_v < d_v \leq \deg_G(v)$ *for all* $v \in V(G)$*, a set of* (w_0, d)*-axioms* A *with* $|A| \leq L(\pi)$*, and a resolution refutation* π' *of* $\mathrm{PM}(G) \cup A$ *such that* $L(\pi') \leq L(\pi)$ *and* $w_{d,\delta}(\pi') \leq O(\alpha \cdot w_0)$*.*

The proof of the above lemma is omitted as it is syntactically equivalent to the proof of Lemma D.4.2. Until this point, we have almost mimicked the proof of Theorem D.4.4. The main differences will appear in the proof of the counterpart to Lemma D.5.2, which states a pseudo-width lower bound.

Lemma D.5.3. *Assume for* $\xi \leq 1/64$ *and* $m, n, r, \Delta \in \mathbb{N}$ *that* $G = (V_L \dot\cup V_R, E)$ *is an* $(r, \Delta, (1-2\xi)\Delta)$*-boundary expander with* $|V_L| = m$*,* $|V_R| = n$*,* $\Delta \geq \log m/\xi^2$*, and* $\min\{\deg_G(v) : v \in V_R\} \geq r/\xi$*. Let* $\delta = (\delta_v \mid v \in V(G))$ *be defined by* $\delta_v = 64 \deg_G(v)\xi$ *and suppose that* $d = (d_v \mid v \in V(G))$ *is an integer vector such that* $\delta_v < d_v \leq \deg_G(v)$ *for all* $v \in V(G)$*. Fix* w_0 *such that* $64 \leq w_0 \leq r\xi - \log n$ *and let* A *be an arbitrary set of* (w_0, d)*-axioms with* $|A| \leq (1 + 16\xi)^{w_0/8}$*. Then every resolution refutation* π *of* $\mathrm{PM}(G) \cup A$ *has either length* $L(\pi) \geq 2^{w_0/32}$ *or pseudo-width* $w_{d,\delta}(\pi) \geq r\xi$*.*

The proof of the above lemma is based on a sort of reduction to the FPHP(G) case. The idea, due to Razborov [Raz04b], is to first pick a partition of the vertices of G that looks random to every clause in the refutation and then simulate the FPHP(G) lower bound on this partition. In our setting, however, this process gets quite involved. Already implementing the partition idea of Razborov is non-trivial: for a fixed clause C some vertices that are light may be super-heavy with respect to the partition, and we do not have an upper bound on the pseudo-width any longer. The insight needed to solve this issue is to show that by expansion there are not too many such vertices per clause, and then adapt the closure definition to take these vertices into account.

Another issue we run into is that the span argument from Section D.4 cannot be applied to all the vertices in the graph. Instead, for the vertices in V_R, we need to resort to the span argument from [Raz03]. Moreover, vertices in the neighbourhood of $\mathcal{D}$ (as defined in the proof of Lemma D.4.10) may already be matched and we are hence unable to attain enough matchings. Our solution is to consider a "lazy" edge removal procedure from the original matching, which with a careful analysis can be shown to circumvent the problem—see Section D.5.3 for details.

D.5.1 Formal Statements of Perfect Matching Formula Lower Bounds

Let us state our lower bounds for the perfect matching formulas and defer the proof of Lemma D.5.3 to Section D.5.3.

Theorem D.5.4. *Let* $G = (U \dot\cup V, E)$ *be a bipartite graph with* $m = |U|$ *and* $n = |V|$. *Suppose that* G *is an* $(r, \Delta, (1-2\xi)\Delta)$*-boundary expander for* $\Delta \geq \frac{\log(m+n)}{\xi^2}$ *and* $\xi = \frac{\log \alpha}{64 \log(m+n)}$ *where* $\alpha \geq 2$ *and* $\frac{\alpha^3}{\log \alpha} = o\left(\frac{r}{\log(m+n)}\right)$, *which furthermore satisfies the degree requirement* $\min\{\deg_G(v) : v \in V\} \geq r/\xi$. *Then resolution requires length* $\exp\left(\Omega\left(\frac{r \log^2 \alpha}{\alpha \log^2(m+n)}\right)\right)$ *to refute the perfect matching formula* $\mathrm{PM}(G)$ *defined over* G.

We remark that this theorem also holds if we replace the minimum degree constraint of V with an expansion guarantee from V to U. We state the theorem in the above form as we want to apply it to the graphs from [GUV09] for which we have no expansion guarantee from V to U.

Proof of Theorem D.5.4. Let $w_0 = \frac{\varepsilon_0 r \xi}{\alpha}$, for some small enough $\varepsilon_0 > 0$. Suppose for the sake of contradiction that π is a resolution refutation of $\mathrm{PM}(G)$ such that $L(\pi) < (1+16\xi)^{w_0/8}$. Since $w_0 > \log L(\pi)$, by Lemma D.5.2 we have that there exists an integer vector $d = (d_1, \ldots, d_{m+n})$, with $\delta_v < d_v \leq \deg_G(v)$, a set of (w_0, d)-axioms A with $|A| \leq L(\pi) < (1+16\xi)^{w_0/8}$, and a resolution refutation π' of $\mathrm{PM}(G) \cup A$ such that $L(\pi') \leq L(\pi)$ and $w_{d,\delta}(\pi') \leq K\alpha w_0$ for some large enough constant K. Since $L(\pi') < (1+16\xi)^{w_0/8} \leq 2^{w_0/32}$, by Lemma D.5.3, we have that $w_{d,\delta}(\pi') \geq r\xi \geq \alpha w_0/\varepsilon_0$. Choosing $\varepsilon_0 < 1/K$, we get a contradiction and, thus, $L(\pi) \geq (1+16\xi)^{w_0/8} = \exp\left(\Omega\left(\frac{r\xi^2}{\alpha}\right)\right)$. □

As in Section D.4, we have a general statement for random graphs.

Corollary D.5.5. *Let* m *and* n *be positive integers, let* $\Delta : \mathbb{N}^+ \longrightarrow \mathbb{N}^+$ *and* $\varepsilon : \mathbb{N}^+ \longrightarrow [0,1]$ *be any monotone functions of* n *such that* $n^3 < m \leq n^{(\varepsilon/128)^2 \log n}$ *and* $n \geq \Delta \geq \log(m+n)\left(\frac{128\log(m+n)}{\varepsilon \log n}\right)^2$. *Then asymptotically almost surely resolution requires length* $\exp\left(\Omega\left(n^{1-\varepsilon}\right)\right)$ *to refute* $\mathrm{PM}(G)$ *for* $G \sim \mathcal{G}(m,n,\Delta)$.

Proof. For simplicity, let us assume that $m^+ = n^{(\varepsilon/128)^2 \log n}$ and $\Delta^- = \log(m+n) \cdot \left((128\log(m+n))/(\varepsilon \log n)\right)^2$ are integers. It suffices to prove the claim for $m = m^+$ and $\Delta = \Delta^-$. Indeed, if $G \sim \mathcal{G}(m,n,\Delta)$, for $\Delta > \Delta^-$, we can sample a random subgraph $G' \sim \mathcal{G}(m,n,\Delta^-)$ of G by choosing a random subset of appropriate size of each neighbourhood of a left vertex and applying a restriction zeroing out the other edges. Furthermore, as for smaller m the formula gets less constrained and hence the lower bound is easier to obtain, it suffices to prove it for $m = m^+$.

We want to apply Lemma D.2.2 for $\chi = \alpha = n^{\varepsilon/4}$ and $\xi = \frac{\log \alpha}{64 \log m}$, and towards this end we argue that the inequalities

$$\xi < 1/2 \ , \tag{D.39a}$$

$$\xi \ln \chi \geq 2 \ , \tag{D.39b}$$

$$\xi \Delta \ln \chi \geq 4 \ln m \tag{D.39c}$$

all hold. First observe that $\xi = \frac{32}{\varepsilon \log n}$ and $n < n^{(\varepsilon/128)^2 \log n}$, from which we conclude that $\frac{1}{\log n} < \left(\frac{\varepsilon}{128}\right)^2$. Hence, the first inequality D.39a holds for n large enough. A simple calculation

$$\xi \ln \chi = \frac{32}{\varepsilon \log n} \frac{\varepsilon \ln n}{4} \geq 2 \tag{D.40}$$

shows that D.39b is also true. Finally, for (D.39c), we observe that $\Delta \geq \log^2 m$ and hence

$$\xi \Delta \ln \chi \geq \frac{8}{\log e} \log^2 m \geq 4 \ln m \ . \tag{D.41}$$

We conclude that asymptotically almost surely $G \sim \mathcal{G}(m,n,\Delta)$ is an $\left(n^{1-\varepsilon/2}, \Delta, (1-2\xi)\Delta\right)$-boundary expander. Furthermore, by the Chernoff inequality asymptotically almost surely all right vertices have degree at least $n \cdot \frac{64\log(m+n)}{\varepsilon \log n}$. Thus, Theorem D.5.4 gives a length lower bound of $\exp\left(\Omega\left(n^{1-\varepsilon}\right)\right)$ as claimed. □

The following corollary is a simple consequence of Corollary D.5.5, optimizing for the strongest lower bounds.

Corollary D.5.6 (Restatement of Theorem D.1.1)**.** *Let* m, n *be such that* $m \le n^{o(\log n)}$*. Then asymptotically almost surely resolution requires length* $\exp\left(\Omega\left(n^{1-o(1)}\right)\right)$ *to refute* $\mathrm{PM}(G)$ *for* $G \sim \mathcal{G}\left(m, n, 8\log^2 m\right)$.

Proof. Let $m = n^{f(n)}$, where $f(n) = o(\log n)$. Applying Corollary D.5.5 for $\varepsilon = 128\sqrt{\frac{f(n)}{\log n}} = o(1)$, we get the desired statement. □

Our final corollary shows that we even get meaningful lower bounds for highly unbalanced bipartite graphs. As was the case for FPHP(G), the required expansion is too strong to hold for random graphs with such large imbalance, but does hold for explicitly constructed graphs from [GUV09].

Corollary D.5.7 (Restatement of Theorem D.1.2)**.** *Let* $\kappa < 3/2 - \sqrt{2}$ *and* $\varepsilon > 0$ *be constants, and let* n *be an integer. Then there is a family of* (*explicitly constructible*) *graphs* G *with* $m = 2^{\Omega(n^{\kappa})}$ *and left degree* $O(\log^{1/\sqrt{\kappa}}(m))$*, such that resolution requires length* $\exp(\Omega(n^{1-2\sqrt{\kappa}(2-\sqrt{\kappa})-\varepsilon}))$ *to refute* $\mathrm{PM}(G)$.

Proof. Let G be the graph from Corollary D.2.4 with $\nu = \frac{2\sqrt{\kappa}}{1-2\sqrt{\kappa}}$. In order to apply Theorem D.5.4 we need to satisfy the minimum right degree constraint. A simple way of doing this is by adding n^2 edges to G such that each vertex on the right has exactly n incident edges added while each vertex on the left has at most one incident edge added. This will leave us with a graph which has large enough right degree while each left degree increased by at most one. The additional edges may reduce the boundary expansion a bit, but a short calculation shows that by choosing $\xi = \frac{\log \alpha}{128\log(m+n)}$ in Corollary D.2.4, we can still guarantee the needed boundary expansion for Theorem D.5.4. The corollary bound follows. □

D.5.2 Defining Pigeons and Holes

As stated earlier, we prove the PM(G) lower bound by simulating the FPHP(G) lower bound from Section D.4 on a partition $V_P \dot\cup V_H$ of the vertices of G. As the notation suggests, we think of the vertices in V_P as pigeons and of the vertices in V_H as holes.

Let us first motivate the properties—captured in Lemma D.5.8—that such a partition must satisfy in order for the FPHP(G) simulation to go through. To begin with, recall that in the proof of Lemma D.4.9 we show that a (w_0, d)-axiom only spans an exponentially small fraction of the linear space Λ. The argument crucially relies on the fact that there are many super-heavy pigeons in every (w_0, d)-axiom. To make this work over the partition $V_P \dot\cup V_H$, we require that a constant fraction of the super-heavy vertices of every (w_0, d)-axiom are in V_P and that super-heavy vertices remain super-heavy with respect to this partition. This first issue is addressed by property 1 of Lemma D.5.8 whereas the second issue is guaranteed by the other properties: property 2 ensures that for every vertex roughly half of its neighbours are in V_H while properties 3 and 4 ensure that most clause-neighbourhoods behave in the same manner, i.e., up to a small set of vertices per clause every clause-neighbourhood of a vertex has roughly half of its vertices in V_H. Combining these arguments, we can bound the fraction of the space spanned by a (w_0, d)-axiom.

The other main step of the FPHP(G) lower bounds is Lemma D.4.10 which state that in low pseudo-width the linear space associated with a resolvent never leaves the span of the premises. This argument relies on the expansion guarantee of the underlying graph and the fact that light pigeons are unconstrained. The required graph expansion (see Lemma D.5.10) will follow from property 2 and properties 2–4 are used to argue that light pigeons are also unconstrained with repect to the partition.

Lemma D.5.8. *Let* $G = (V_L \dot\cup V_R, E)$ *be an* $(r, \Delta, (1 - 2\xi)\Delta)$*-boundary expander for* $\xi \leq 1/4$ *and* $|V_L| \geq 4$*. Fix* w_0 *such that* $64 \leq w_0 \leq r$ *and let* $\mathcal{A}$ *be a set of* (w_0, d)*-axioms of size* $|\mathcal{A}| \leq \exp(w_0/32)$*. Moreover, suppose that* $\Delta \geq \log|V_L|/\xi^2$ *and* $\min\{\deg_G(v) : v \in V_R\} \geq (\log|V_R| + w_0)/\xi^2$*. If* π *is a resolution refutation of* $\mathrm{PM}(G) \cup \mathcal{A}$ *with* $L(\pi) \leq \exp(w_0/32)$*, then there exists a vertex partition* $V(G) = V_P \dot\cup V_H$ *such that*

1. *for every* $A \in \mathcal{A}$*:*
$$|V_d(A) \cap V_P| \geq w_0/4 \ ,$$

2. *for every* $v \in V$*:*
$$\big||N_G(v) \cap V_H| - 1/2|N_G(v)|\big| \leq 4\xi|N_G(v)| \ ,$$

3. *for every* $C \in \pi$ *and for every* $v \in V_R$*:*
$$\big||N_C(v) \cap V_H| - 1/2|N_C(v)|\big| \leq 4\xi|N_G(v)| \ ,$$

4. *for every* $C \in \pi$ *there is a set of vertices* $\widetilde{V}(C) \subseteq V_L$*, satisfying* $|\widetilde{V}(C)| \leq w_0/8$*, such that for every* $v \in V_L \setminus \widetilde{V}(C)$*:*
$$\big||N_C(v) \cap V_H| - 1/2|N_C(v)|\big| \leq 4\xi\Delta \ .$$

The analogue of above lemma in [Raz04b] is Claim 19. The main difference is that in our setting property 4 does not always hold for all vertices in the graph while in Razborov's setting the corresponding property always holds.

In order to argue that this error set $\widetilde{V}(C)$ is small, we need G to be a good expander. To this end we use the following claim which states that if for a fixed clause C there are many vertices $v \in V_L$ such that $|N_C(v) \cap V_H|$ does not behave as expected, then, by expansion, we can find a large set of vertices $\widetilde{V}^\star(C)$ whose clause-neighbourhood in V_H (i.e., the set $N_C(\widetilde{V}^\star(C)) \cap V_H$) deviates from its expected size.

Claim D.5.9. *Let* $G = (V_L \dot\cup V_R, E)$ *be an* $(r, \Delta, (1-2\xi)\Delta)$*-boundary expander. Fix any partition* $V(G) = V_P \dot\cup V_H$ *and any clause* C*. Let*

$$\widetilde{V}(C) = \{v \in \underset{L}{V} : \big||N_C(v) \cap \underset{H}{V}| - 1/2|N_C(v)|\big| > 4\xi\Delta\} \ .$$

If $|\widetilde{V}(C)| > w_0/8$*, then there is a set of vertices* $\widetilde{V}^\star(C) \subseteq \widetilde{V}(C)$*, with* $|\widetilde{V}^\star(C)| = w_0/16$*, such that*

$$\big||N_C(\widetilde{V}^\star(C)) \cap \underset{H}{V}| - 1/2|N_C(\widetilde{V}^\star(C))|\big| > 2\xi\Delta|\widetilde{V}^\star(C)| \ .$$

Proof. Denote by $\widetilde{V}^+(C)$ ($\widetilde{V}^-(C)$ respectively) the vertices in $\widetilde{V}(C)$ that have more neighbours (less neighbours respectively) in V_H than the expected $1/2\big|N_C(v)\big|$. As $|\widetilde{V}(C)| > w_0/8$, one of the sets $\widetilde{V}^+(C)$ or $\widetilde{V}^-(C)$ is of cardinality at least $w_0/16$.

Case 1: Suppose $\widetilde{V}^+(C) \geq w_0/16$ and let $\widetilde{V}^\star(C)$ be any subset of $\widetilde{V}^+(C)$ of size $w_0/16$. As boundary expansion of G guarantees that $\widetilde{V}^\star(C)$ has at most $2\xi\Delta|\widetilde{V}^\star(C)|$ *edges* to non-unique neighbours in G we derive

$$\begin{aligned} |N_C(\widetilde{V}^\star(C)) \cap \underset{H}{V}| &\geq \sum_{v \in \widetilde{V}^\star(C)} |N_C(v) \cap \partial(\widetilde{V}^\star(C)) \cap \underset{H}{V}| && \text{(D.42)} \\ &\geq \sum_{v \in \widetilde{V}^\star(C)} |N_C(v) \cap \underset{H}{V}| - 2\xi\Delta|\widetilde{V}^\star(C)| && \text{(D.43)} \\ &> \sum_{v \in \widetilde{V}^\star(C)} \big(1/2|N_C(v)| + 4\xi\Delta\big) - 2\xi\Delta|\widetilde{V}^\star(C)| && \text{(D.44)} \\ &\geq 1/2|N_C(\widetilde{V}^\star(C))| + 2\xi\Delta|\widetilde{V}^\star(C)| \ , && \text{(D.45)} \end{aligned}$$

where the strict inequality follows by definition of $\widetilde{V}^+(C)$.

Case 2: Suppose $\widetilde{V}^-(C) \geq w_0/16$ and let $\widetilde{V}^\star(C)$ be any subset of $\widetilde{V}^-(C)$ of size $w_0/16$. Similar to the previous case we can conclude that

$$|N_C(\widetilde{V}^\star(C)) \cap V_H| \leq \sum_{v \in \widetilde{V}^\star(C)} |N_C(v) \cap V_H| \tag{D.46}$$

$$< \sum_{v \in \widetilde{V}^\star(C)} \big(1/2|N_C(v)| - 4\xi\Delta\big) \tag{D.47}$$

$$\leq 1/2|N_C(\widetilde{V}^\star(C))| - 2\xi\Delta|\widetilde{V}^\star(C)| \ , \tag{D.48}$$

where the last inequality uses that $\widetilde{V}^\star(C)$ has at most $2\xi\Delta|\widetilde{V}^\star(C)|$ edges incident to non-unique neighbours in G.

Combining both cases yields the claim. □

Proof of Lemma D.5.8. Pick a partition $V = V_{\mathbf{P}} \mathbin{\dot\cup} V_{\mathbf{H}}$ uniformly at random. In what follows we show that property 1 holds with probability at least 3/4 and properties 2, 3 and 4 each hold with probability at least 7/8. Hence there exists a partition that satisfies all four properties simultaneously.

For the first property, since $\mathbb{E}\big[|V_d(A) \cap V_{\mathbf{P}}|\big] = w_0/2$, by the multiplicative Chernoff bound we have that

$$\Pr\big[|V_d(A) \cap V_{\mathbf{P}}| \leq w_0/4\big] \leq \exp\left(-w_0/16\right) \ . \tag{D.49}$$

Since $|A| \leq \exp(w_0/32)$ and $w_0 \geq 64$, a union bound over A gives us that property 1 holds except with probability $\exp(-w_0/32) \leq 1/4$.

To analyse properties 2 and 3, let C either be a clause in π or be the graph G (i.e., the clause that contains all variables) and fix an arbitrary $v \in V(G)$. By Chernoff bound (Theorem D.2.1) we get that

$$\begin{aligned}\Pr\Big[\big||N_C(v) \cap V_{\mathbf{H}}| - 1/2|N_C(v)|\big| \geq 4\xi|N_G(v)|\Big]& \\ \leq 2\exp\left(-\frac{(4\xi|N_G(v)|)^2}{|N_C(v)| + 4\xi|N_G(v)|}\right)& \\ \leq \exp\left(-8\xi^2|N_G(v)| + 1\right) \ ,&\end{aligned} \tag{D.50}$$

where the last inequality holds as $|N_C(v)| \leq |N_G(v)|$ and $\xi \leq 1/4$.

By a union bound argument over the clauses in π and $v \in V_R$, we have that Property 3 holds except with probability 1/8. For property 2, we need to analyse vertices in V_L and in V_R separately. On the one hand, since $\min\{\deg_G(v) : v \in V_L\} \geq (1-2\xi)\Delta \geq \frac{\log|V_L|}{2\xi^2}$ and $|V_L| \geq 4$, a union bound over $v \in V_L$ shows that property 2 holds for all vertices V_L except with probability 1/16. On the other, as $\min\{\deg_G(v) : v \in V_R\} \geq$

$(\log|V_R|+w_0)/\xi^2$, a union bound yields that property 2 holds for all $v \in V_R$ except with probability $1/16$.

To obtain property 4, fix a clause C and consider the set $\widetilde{\mathbf{V}}(C)$ that contains all vertices $v \in V_L$ satisfying

$$\big||N_C(v) \cap \underset{\mathbf{H}}{V}| - 1/2|N_C(v)|\big| > 4\xi\Delta \ . \tag{D.51}$$

We want to show that it is unlikely that $|\widetilde{\mathbf{V}}(C)| \geq w_0/8$. Note that such a large $\widetilde{\mathbf{V}}(C)$ implies by Claim D.5.9 that there is a set $S \subseteq V_L$ of size $w_0/16$ such that $\big||N_C(S) \cap V_{\mathbf{H}}| - 1/2|N_C(S)|\big| \geq 2\xi\Delta|S|$. By a union bound over all such sets S and applying Chernoff bound (Theorem D.2.1) we have that

$$\begin{aligned}
&\Pr\big[|\widetilde{\mathbf{V}}(C)| \geq w_0/8\big] \\
&\quad \leq \binom{|V_L|}{w_0/16} \max_{\substack{S \subseteq V_L: \\ |S|=w_0/16}} \Pr\Big[\big||N_C(S) \cap \underset{\mathbf{H}}{V}| - 1/2|N_C(S)|\big| \geq \xi\Delta w_0/8\Big] && \text{(D.52)} \\
&\quad \leq |\underset{L}{V}|^{w_0/16} \cdot 2\exp\left(-\frac{(\xi\Delta w_0/8)^2}{\Delta w_0/16 + \xi\Delta w_0/8}\right) && \text{(D.53)} \\
&\quad \leq \exp\left(-\xi^2\Delta w_0/8 + 1 + \log|\underset{L}{V}| \cdot w_0/16\right) && \text{(D.54)} \\
&\quad \leq \exp\left(-\log|\underset{L}{V}| \cdot w_0/16 + 1\right) \ , && \text{(D.55)}
\end{aligned}$$

where for D.53 we observe that $|N_C(S)| \leq \Delta|S|$, for D.54 we need that $\xi \leq 1/4$ and for D.55 that $\Delta \geq \log|V_L|/\xi^2$. By a union bound over all clauses in π we see that property 4 holds except with probability $1/8$. □

Let $V_P \dot\cup V_H$ be a partition of $V(G)$ as guaranteed to exist by Lemma D.5.8. For an overview of the vertex sets and how they relate we refer to Figure D.2. The following lemma shows that the vertices in V_L expand into the set $V_R \cap V_H$. Let $G' = G \setminus (V_R \cap V_P)$ with vertex partition $(V_L \dot\cup (V_R \setminus V_P))$.

Lemma D.5.10. *The graph* G' *is an* $(r, (1+8\xi)\Delta/2, (1-12\xi)\Delta/2)$*-boundary expander.*

Proof. By Lemma D.5.8, property 2, every vertex in $V_P \cap V_L$ has degree at most $(1+8\xi)|N_G(v)|/2$ and at least $(1-8\xi)|N_G(v)|/2$. By the expansion guarantee of G, we know that $|N_G(v)| \geq (1-2\xi)\Delta$. Therefore all sets of size 1 are good enough boundary expanders. We continue by induction on the size of the set. Let S be a set of vertices of size at most r. In the original graph G, this set S has at least $(1-2\xi)\Delta|S|$ many unique neighbours. Thus there is a vertex v in S that has at least $(1-2\xi)\Delta$ many unique neighbours in G. Further, by Lemma D.5.8, property 2, the vertex v has at least $(1-8\xi)\Delta/2$ many neighbours in $V_R \cap V_H$. Hence v has at least $(1-12\xi)\Delta/2$ many unique neighbours in V_H. From the induction hypothesis on $S \setminus \{v\}$, it follows that S has the required number of unique neighbours in V_H. □

D.5.3 Pseudo-Width Lower Bound

We start by setting up the notation we will need to prove Lemma D.5.3.

Let C be a clause in π, let $\widetilde{V}(C) = \{v \in V_L : \big||N_C(v) \cap V_H| - 1/2|N_C(v)|\big| > 4\xi\Delta\}$ and $\overline{V}(C) = (V_{d,\delta}(C) \cap V_L) \cup \widetilde{V}(C)$. The closure of C is a subset of V_L in the graph G', defined by

$$\operatorname{closure}(C) = \operatorname{closure}_{r,(1-20\xi)\Delta/2}(\overline{V}(C)) \ . \tag{D.56}$$

We define the closure only on V_L as we only have an expansion guarantee from V_L into $V_R \cap V_H$. As the concept of closure only makes sense on vertex sets which are expanding, we do not define it on V_R. The set of relevant vertices of a clause C are the vertices in $\operatorname{closure}(C) \cup V_{d,\delta}(C)$. With this definition at hand we proceed to set up the linear spaces that realize the lossy counting (see Section D.4). Let us stress the fact that only vertices in V_P are associated with a linear space.

Fix a field $\mathbb{F}$ of characteristic 0 and for each vertex $v \in V_P$ let Λ_v be a linear space over $\mathbb{F}$ of dimension $1/2(\deg_G(v) - d_v + \delta_v/2)$. Let $\Lambda = \bigotimes_{v \in V_P} \Lambda_v$ and denote by $\lambda_v : V_H \mapsto \Lambda_v$ a function with the property that any image of a subset $S \subseteq V_H$ of size $|S| \geq \dim(\Lambda_v)$ spans Λ_v, i.e., $\operatorname{span}(\lambda_v(u) : u \in S) = \Lambda_v$.

Let $\mathcal{M}$ be the set of partial matchings in G that contain no edges from $V_P \times V_P$. To map partial matchings $\varphi \in \mathcal{M}$ into Λ, we define $\lambda : \mathcal{M} \mapsto \Lambda$ by

$$\lambda(\varphi) = \bigotimes_{v \in V(\varphi) \cap V_P} \lambda_v(\varphi_v) \otimes \bigotimes_{v \in V_P \setminus V(\varphi)} \Lambda_v \ . \tag{D.57}$$

Recall that each partial matching $\varphi \in \mathcal{M}$ has an associated partial boolean assignment ρ_φ as defined in D.4. For each clause C, we are interested in the partial matchings $\varphi \in \mathcal{M}$ that match all of $\operatorname{closure}(C) \cup V_{d,\delta}(C)$ such that ρ_φ does not satisfy C. We refer to the set of such matchings as the *zero space* of C and denote it by

$$Z(C) = \{\varphi \in \mathcal{M} \mid V(\varphi) \supseteq (\operatorname{closure}(C) \cup V_{d,\delta}(C)) \land C(\rho_\varphi) \neq 1\} \ . \tag{D.58}$$

We associate C with the linear space

$$\lambda(C) = \operatorname{span}(\lambda(\varphi) \mid \varphi \in Z(C)) \ . \tag{D.59}$$

Note that contradiction is mapped to Λ, i.e., $\lambda(\perp) = \Lambda$.

The following lemma asserts that the span of the axioms $\mathrm{span}(\{\lambda(A) \mid A \in \mathrm{PM}(G) \cup \mathcal{A}\})$ is a proper subspace of Λ.

Lemma D.5.11. *If* $|\mathcal{A}| \leq (1+16\xi)^{w_0/8}$, *then* $\mathrm{span}(\{\lambda(A) \mid A \in \mathrm{PM}(G) \cup \mathcal{A}\}) \subsetneq \Lambda$.

Deferring the proof of this lemma for now, note this implies that in the refutation π there is a resolution step deriving C from C_0 and C_1 where the subspace of the resolvent is not contained in the span of the subspaces of the premises, or in other words $\lambda(C) \not\subseteq \mathrm{span}(\lambda(C_0), \lambda(C_1))$. The following lemma, which is the heart of the argument, says that this cannot happen while the sets of relevant vertices of the clauses are small.

Lemma D.5.12. *Let* C *be derived from* C_0 *and* C_1. *If* $\max\{|\mathrm{closure}(C_0) \cup V_{d,\delta}(C_0)|, |\mathrm{closure}(C_1) \cup V_{d,\delta}(C_1)|, |\mathrm{closure}(C) \cup V_{d,\delta}(C)|\} \leq r/4$, *then* $\lambda(C) \subseteq \mathrm{span}(\lambda(C_0), \lambda(C_1))$.

Deferring the proof of Lemma D.5.12 to Section D.5.4, we proceed to show how Lemma D.5.3 follows from what we have established so far.

Proof of Lemma D.5.3. Lemma D.5.11 and Lemma D.5.12 imply that contradiction cannot be derived while the set of relevant vertices is of size at most $r/4$ and hence any refutation π must contain a clause C with $|\mathrm{closure}(C) \cup V_{d,\delta}(C)| \geq r/4$. If for such a clauses C it holds that $|V_{d,\delta}(C)| \geq r\xi$, then Lemma D.5.3 follows. Otherwise, recall that $\mathrm{closure}(C) = \mathrm{closure}_{r,\nu}(\overline{V}(C))$, for $\nu = (1-20\xi)\Delta/2$, and that G' is an (r, Δ', c)-boundary expander by Lemma D.5.10, where $\Delta' = (1+8\xi)\Delta/2$ and $c = (1-12\xi)\Delta/2$. Thus we can apply Lemma D.3.3 to G' and get that $|\overline{V}(C)| \geq \min\{r, (r/4 - r\xi) \cdot (c-\nu)/\Delta'\} \geq 3r\xi/2$. As by definition $\overline{V}(C) = (V_{d,\delta}(C) \cap V_L) \cup \widetilde{V}(C)$ and by property 4 of Lemma D.5.8 we have that $|\widetilde{V}(C)| \leq w_0/8$, we conclude that

$$w_{d,\delta}(\pi) \geq |V_{d,\delta}(C)| \geq |V_{d,\delta}(C) \cap V_L| \geq |\overline{V}(C)| - |\widetilde{V}(C)| \geq 3r\xi/2 - w_0/8 \geq r\xi \ . \tag{D.60}$$

This completes the proof of Lemma D.5.3. □

Proof of Lemma D.5.11. Suppose A is a vertex axiom P^v or a functionality axiom $F^v_{w,w'}$ as in D.1a and D.1c. Observe that v is a heavy vertex for A. Clearly, there are no matchings on v that do not satisfy A. We conclude that $\lambda(A) = \emptyset$.

Let us consider $A \in \mathcal{A}$. These axioms may span a part of the space Λ but the fraction of the space Λ they span is sufficiently small. We first estimate the dimension of $\lambda(A)$. By definition $\widetilde{V}(A) = \{v \in V_L : \big||N_A(v) \cap V_H| - 1/2|N_A(v)|\big| > 4\xi\Delta\}$ and by property 4 of Lemma D.5.8 it holds that

$|\widetilde{V}(A)| \leq w_0/8$. We partition V_P into two sets $U = V_P \setminus \big(V_{d,\delta}(A) \setminus \widetilde{V}(A)\big)$ and $W = V_P \cap \big(V_{d,\delta}(A) \setminus \widetilde{V}(A)\big)$. Note that all vertices $v \in W$ satisfy that $\big||N_A(v) \cap V_H| - 1/2|N_A(v)|\big| \leq 4\xi\Delta$. Using property 2 of Lemma D.5.8 we get that

$$\dim \lambda(A) \leq \prod_{v\in U} \dim \Lambda_v \cdot \prod_{v \in W} \big(|N_G(v) \cap \underset{H}{V}| - |N_A(v) \cap \underset{H}{V}|\big) \tag{D.61}$$

$$\leq \prod_{v\in U} \dim \Lambda_v \cdot \prod_{v \in W} \big(1/2|N_G(v)| + 4\xi|N_G(v)| - 1/2|N_A(v)| + 4\xi|N_G(v)|\big) \tag{D.62}$$

$$= \prod_{v\in U} \dim \Lambda_v \cdot \prod_{v \in W} \big(1/2\big(|N_G(v)| - |N_A(v)|\big) + 8\xi|N_G(v)|\big) \tag{D.63}$$

$$\leq \prod_{v\in U} \dim \Lambda_v \cdot \prod_{v \in W} \big(1/2(\deg_G(v) - d_v) + \delta_v/8\big) \tag{D.64}$$

$$\leq \prod_{v\in U} \dim \Lambda_v \cdot \prod_{v \in W} \big(\dim \Lambda_v - \delta_v/8\big) \enspace , \tag{D.65}$$

where the second to last inequality follows from the fact that $\delta_v = 64\xi|N_G(v)|$ and the last inequality from the definition of $\dim \Lambda_v$.

Note that by property 1 of Lemma D.5.8, $|V_P \cap V_{d,\delta}(A)| \geq w_0/4$ and hence $|W| \geq w_0/8$. We conclude that the fraction of the space Λ that A spans is bounded by

$$\frac{\dim \lambda(A)}{\dim \Lambda} \leq \prod_{v \in W} \frac{\dim \Lambda_v - \delta_v/8}{\dim \Lambda_v} \leq (1 - 16\xi)^{w_0/8} \enspace . \tag{D.66}$$

Along with the assumption on $|A|$, this shows that not all of Λ is spanned by the axioms. □

D.5.4 Proof of Lemma D.5.12

For conciseness of notation, let us write

$$S_{01} = (\mathrm{closure}(C_0) \cup \mathrm{closure}(C_1)) \cup (V_{d,\delta}(C_0) \cup V_{d,\delta}(C_1)) \tag{D.67}$$

and $S = \mathrm{closure}(C) \cup V_{d,\delta}(C)$. In order to establish Lemma D.5.12, we need to show for all $\varphi \in Z(C)$ that

$$\lambda(\varphi) \subseteq \mathrm{span}(\lambda(C_0), \lambda(C_1)) \enspace . \tag{D.68}$$

Denote by φ' the restriction of φ to the edges with at least one vertex in $S \cap S_{01}$ and note that C is not satisfied under $\rho_{\varphi'}$. Also, observe that if a

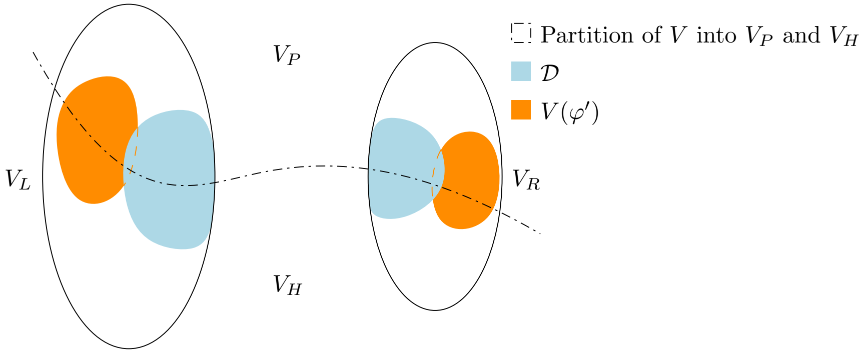


Figure D.2: Depiction of relations between V_L, V_R, V_P, V_H and the vertex sets in the proof of Lemma D.5.12

matching η extends a matching η', then $\lambda(\eta)$ is a subspace of $\lambda(\eta')$. This is so since for any vertex $v \in V_P \cap \big(V(\eta) \setminus V(\eta')\big)$ we have from D.57 that η' picks up the whole subspace Λ_v while η only gets a single vector. Thus, if we can show that $\lambda(\varphi') \subseteq \text{span}(\lambda(C_0), \lambda(C_1))$, the statement follows since φ extends φ' and hence $\lambda(\varphi) \subseteq \lambda(\varphi')$.

Let $\mathcal{D} = S_{01} \setminus S$ and for a set of matchings $\mathcal{N} \subseteq \mathcal{M}$ let $\lambda(\mathcal{N}) = \text{span}(\{\lambda(\psi) \mid \psi \in \mathcal{N}\})$. In the following we show that there exists a set of matchings $\mathcal{M}_{\mathcal{D}} \subseteq \mathcal{M}$ that do not satisfy C, that cover S_{01} and such that

$$\lambda(\varphi') \subseteq \lambda(\mathcal{M}_{\mathcal{D}}) \ . \tag{D.69}$$

Before arguing the existence of such a set $\mathcal{M}_{\mathcal{D}}$ let us argue that this would imply the lemma. Observe that by soundness of resolution, no matching in $\mathcal{M}_{\mathcal{D}}$ can satisfy both C_0 and C_1 simultaneously. Fix $\psi \in \mathcal{M}_{\mathcal{D}}$. Without loss of generality, assume that C_0 is not satisfied. Denote by $\psi' \subseteq \psi$ all edges in ψ with at least one vertex in $\text{closure}(C_0) \cup V_{d,\delta}(C_0)$. Clearly, $\psi' \in Z(C_0)$ and hence $\lambda(\psi) \subseteq \lambda(\psi') \subseteq \lambda(C_0)$. Thus, for all matchings $\psi \in \mathcal{M}_{\mathcal{D}}$ we have that $\lambda(\psi) \subseteq \text{span}(\lambda(C_0), \lambda(C_1))$. Combining with D.69, we get that

$$\lambda(\varphi') \subseteq \lambda(\mathcal{M}_{\mathcal{D}}) \subseteq \text{span}(\lambda(C_0), \lambda(C_1)) \tag{D.70}$$

and hence the lemma follows.

In the remainder, we show how to construct the set $\mathcal{M}_{\mathcal{D}}$. Observe that all vertices $v \in \mathcal{D}$ are light vertices of C. Using property 3 from Lemma D.5.8 we get that for all $v_r \in \mathcal{D} \cap V_R$ there are at most

$$\left|N_C(v_r) \cap \underset{H}{V}\right| \leq 1/2\left|N_C(v_r)\right| + 4\xi\left|N_G(v_r)\right| \leq 1/2\left(d_{v_r} - \delta_{v_r} + 8\xi\left|N_G(v_r)\right|\right) \tag{D.71}$$

mappings of v_r to a vertex in $N_G(v_r) \cap V_H$ that satisfy the clause C. Similarly, using property 4 from Lemma D.5.8 and the fact that $\mathcal{D} \cap \widetilde{V}(C) = \emptyset$ we see that for all $v_\ell \in \mathcal{D} \cap V_L$ there are at most

$$\left|N_C(v_\ell) \cap \underset{H}{V}\right| \leq 1/2\left|N_C(v_\ell)\right| + 4\xi\left|N_G(v_\ell)\right| \leq 1/2\left(d_{v_\ell} - \delta_{v_\ell} + 8\xi\Delta\right) \quad \text{(D.72)}$$

mappings of v_ℓ to a vertex in $N_G(v_\ell) \cap V_H$ that satisfy the clause C.

For a set of vertices $W \subseteq V_P \cup V_H$, let $\Lambda_W = \bigotimes_{w \in W \cap V_P} \Lambda_w$ and for a set $U \subseteq V(G)$ let λ^U be the projection of λ to the space Λ_U or in other words

$$\lambda^U(\eta) = \bigotimes_{v \in V(\eta) \cap V_P \cap U} \lambda_v(\eta_v) \otimes \bigotimes_{v \in (V_P \cap U) \setminus V(\eta)} \Lambda_v \ . \quad \text{(D.73)}$$

We extend the notation to sets of matchings as previously for λ. In order to establish D.69, we have to argue that $\lambda^{\mathcal{D} \setminus V(\varphi')}(\mathcal{M}_{\mathcal{D}})$ spans the space $\Lambda_{\mathcal{D} \setminus V(\varphi')}$. At this point, we deviate from the FPHP(G) proof. Note that we only have expansion for the vertices V_L into V_H but $\mathcal{D}$ may also contain vertices from V_R. Thus we cannot apply the argument from Section D.4 to all vertices.

Instead, we split the argument into 2 seperate parts. First, by an argument similar to the lower bound proof of the FPHP(G) formulas, we show that vertices in $\mathcal{D} \cap V_L$ can be matched in many ways. This will in particular imply that $\lambda^{(\mathcal{D} \cap V_L) \setminus V(\varphi')}(\mathcal{M}_{\mathcal{D}})$ spans all of $\Lambda_{(\mathcal{D} \cap V_L) \setminus V(\varphi')}$. After that we consider the vertices in $\mathcal{D} \cap V_R$. As these vertices have very high degree, there are always enough neighbours they can be matched to and therefore $\lambda^{(\mathcal{D} \cap V_R) \setminus V(\varphi')}(\mathcal{M}_{\mathcal{D}})$ spans all of $\Lambda_{(\mathcal{D} \cap V_R) \setminus V(\varphi')}$. Note that this second argument is essentially the span argument from [Raz03].

Consider the vertex set $\mathcal{D} \cap V_L$. Note that $\mathcal{D} \cap V_L$ is completely outside the closure(C). Since, by assumption, the cardinality of closure(C) is upper bounded by $r/4$ and $|\mathcal{D} \cap V_L| \leq |S_{01}| \leq r/2$, by Lemma D.3.4 we get that

$$|\partial_{G' \setminus (\text{closure}(C) \cup N_{G'}(\text{closure}(C)))}(\mathcal{D} \cap \underset{L}{V})| \geq 1/2(1 - 20\xi)\Delta|\mathcal{D} \cap \underset{L}{V}| \ . \quad \text{(D.74)}$$

By an averaging argument, there is a $v \in \mathcal{D} \cap V_L$ that has at least $(1-20\xi)\Delta/2$ unique neighbours in $\partial_{G' \setminus (\text{closure}(C) \cup N_{G'}(\text{closure}(C)))}(\mathcal{D} \cap V_L)$. By iterating this argument on $(\mathcal{D} \cap V_L) \setminus \{v\}$ we get a partition $V_{v_1} \dot{\cup} V_{v_2} \dots \dot{\cup} V_{v_{|\mathcal{D} \cap V_L|}}$ of the neighbourhood $\mathcal{D} \cap V_L$. The key properties of this partition are that every vertex $v_\ell \in \mathcal{D} \cap V_L$ can independently be matched to any vertex in V_{v_ℓ} and each set is of size at least $|V_{v_\ell}| \geq (1 - 20\xi)\Delta/2$. Using D.72, we have that each vertex $v_\ell \in \mathcal{D} \cap V_L$ can be matched to at least

$$\begin{aligned} 1/2(1 - 20\xi)\Delta - 1/2(d_{v_\ell} - \delta_{v_\ell} + 8\xi\Delta) &= 1/2\left(\Delta - d_{v_\ell} + \delta_{v_\ell} - 28\xi\Delta\right) \\ &\geq 1/2\left(\deg_G(v_\ell) - d_{v_\ell} + \delta_{v_\ell}/2\right) \end{aligned} \quad \text{(D.75)}$$

Algorithm 5 Extend Matching

```
procedure ExtendMatching($T, \psi, V_{v_1}, V_{v_2}, \dots, V_{v_{|T|}}$)    ▷ extend $\psi$ to domain $T$
    if $T \setminus V(\psi) \neq \emptyset$ then    ▷ still need to extend $\psi$
        $\mathcal{M} \leftarrow \emptyset$
        $v_\ell \leftarrow_{\text{any}} T \setminus V(\psi)$
        for $w \in V_{v_\ell}$ do    ▷ $v_\ell$ can be matched to $w$
            $\psi' \leftarrow \psi$
            if $\exists w'$ such that $\{w, w'\} \in \psi$ then
                $\psi' \leftarrow \psi' \setminus \{w, w'\}$    ▷ remove $w$ from the matching
            $\psi' \leftarrow \psi' \cup \{v_\ell, w\}$    ▷ match $v_\ell$ to $w$
            $\mathcal{M} = \mathcal{M} \cup$ ExtendedMatching($T, \psi', V_{v_1}, V_{v_2}, \dots, V_{v_{|T|}}$)
        return $\mathcal{M}$
    else
        return $\psi$
```

many vertices in V_{v_ℓ} without satisfying C. Denote these vertices by V'_{v_ℓ}. As in section Section D.4, we would like to conclude that every vertex has many choices of vertices it can independently be mapped to and therefore there are enough matchings to span the space $\Lambda_{\mathcal{D} \cap V_L}$. Unfortunately this argument does not work since vertices in V'_{v_ℓ} can be matched in φ' and are hence not available to be matched to v_ℓ, so there might be too few matchings of v_ℓ to span the whole space Λ_{v_ℓ}.

We could attempt to overcome this problem by removing all edges in φ' with a vertex in one of the sets V'_{v_ℓ}. This allows us to independently match all the vertices in $\mathcal{D} \cap V_L$ to sufficiently many neighbours. Regrettably, this edge removal strategy turns out to be too aggressive: it can occur that a vertex from $S_{01} \cap V_R$, previously matched by φ', now has no neighbour available to be matched to. Fortunately, this only happens to vertices that were matched in φ'. The solution that suggests itself is to remove edges from φ' in a "lazy" manner: only remove an edge $\{u, v\}$ from φ' when one of the vertices should be matched to some $v_\ell \in V_L$. This ensures that no vertex in V_R that was previously matched by φ' is suddenly unmatched. This is the main idea of Algorithm 5 which takes care of the necessary edge removals.

Let $\mathcal{M}_{\mathcal{D} \cap V_L} = \text{ExtendMatching}(\mathcal{D} \cap V_L, \varphi', V'_{v_1}, \dots, V'_{v_{|\mathcal{D} \cap V_L|}})$. Note that the algorithm terminates on this input as the sets $V'_{v_1}, V'_{v_2}, \dots, V'_{v_{|\mathcal{D} \cap V_L|}}$ are disjoint. Let us establish some claims regarding $\mathcal{M}_{\mathcal{D} \cap V_L}$.

The first claim states that the algorithm cannot remove edges from φ' with a vertex in $S \cap S_{01} \cap V_L$. This is important as we want to get matchings that are defined on all of $S_{01} \cap V_L$. As the algorithm only tries to match vertices in $\mathcal{D} \cap V_L = (S \setminus S_{01}) \cap V_L$, we must ensure that the edges in φ' with an endpoint in $S \cap S_{01} \cap V_L$ are not erased. Note that all edges that are removed by the algorithm have an endpoint in the neighbourhood of $\mathcal{D} \cap V_L$. Hence it suffices to show that the vertices from $S \cap S_{01} \cap V_L$ are not matched to a vertex in the neighbourhood of $\mathcal{D} \cap V_L$.

Claim D.5.13. *The matching φ' contains no edge $\{w, w'\}$ such that*

$$w \in N_{G' \setminus (\mathrm{closure}(C) \cup N_{G'}(\mathrm{closure}(C)))}(\mathcal{D} \cap \underset{L}{V})$$

and $w' \in S \cap S_{01} \cap V_L$.

Proof. Suppose there is an edge $\{w, w'\} \in \varphi$ for w, w' as in the lemma statement. As $S \cap S_{01} \cap V_L \subseteq \mathrm{closure}(C)$, we see that $w \in N_{G'}(\mathrm{closure}(C))$. But this is a contradiction since w is not in the graph $G' \setminus (\mathrm{closure}(C) \cup N_{G'}(\mathrm{closure}(C)))$. □

Next, we consider edges in φ' with a vertex in the set $V_P \cap V_R$. Observe that if the algorithm removed such an edge, then the linear space associated with the new matching would differ from the original space in a non-trivial way. Fortunately, this cannot happen.

Claim D.5.14. *All matchings $\psi \in \mathcal{M}_{\mathcal{D} \cap V_L}$ cover the set $S_{01} \cap V_L$ and an edge $e \in V_L \times (V_P \cap V_R)$ is contained in ψ if and only if it is contained in φ'. Furthermore, if a vertex $v \in V_R$ is matched in φ', then it is matched in every $\psi \in \mathcal{M}_{\mathcal{D} \cap V_L}$.*

Proof. By Claim D.5.13, Algorithm 5 never removes edges from φ' that are incident to a vertex in $S_{01} \cap S \cap V_L$. As φ' covers all of $S_{01} \cap S \cap V_L$, it follows that every $\psi \in \mathcal{M}_{\mathcal{D} \cap V_L}$ also covers the set $S \cap S_{01} \cap V_L$. Furthermore, the algorithm ensures that every $\psi \in \mathcal{M}_{\mathcal{D} \cap V_L}$ covers the set $\mathcal{D} \cap V_L = (S_{01} \setminus S) \cap V_L$. Combining these statements we see that every matching $\psi \in \mathcal{M}_{\mathcal{D} \cap V_L}$ covers $S_{01} \cap V_L$.

We observe that all edges in φ' that may be deleted by the algorithm must have an endpoint in one of the sets V'_{v_ℓ} and all these sets are contained in $V_H \cap V_R$. As the graph is bipartite (with bipartition $V_L \dot{\cup} V_R$) and the set $\mathcal{M}$ does not contain matchings with edges from $V_P \times V_P$, we see that vertices from $V_P \cap V_R$ can only be matched to vertices in $V_H \cap V_L$. Therefore the algorithm cannot change edges in φ' with an endpoint in $V_P \cap V_R$. This implies that if an edge $e \in V_L \times (V_P \cap V_R)$ is in φ', then it is also in ψ. For the other direction, observe that since the algorithm can only add edges to ψ with an endpoint in $V_H \cap V_R$, and since the graph is bipartite, no edge from $V_L \times (V_P \cap V_R)$ gets added by the algorithm.

Finally, the fact that all matched vertices $v \in V_R$ in φ' are also matched in every $\psi \in \mathcal{M}_{\mathcal{D} \cap V_L}$ follows from the "lazy" removal of edges from φ'. □

We can now show that our set of matchings spans the appropriate space when projected to V_L. Note that for a matching η it holds that $\lambda(\eta) = \lambda^{U}(\eta) \otimes \lambda^{V_P \setminus U}(\eta)$, for any set U but the same does not hold for sets of matchings: span does not commute with tensor.

Claim D.5.15. $\lambda^{V_L}(\varphi') \subseteq \lambda^{V_L}(\mathcal{M}_{\mathcal{D} \cap V_L})$

Proof. Let us write

$$\begin{aligned}\lambda^{V_L}(\varphi') &= \lambda^{V(\varphi') \cap V_L}(\varphi') \otimes \Lambda_{V_L \setminus V(\varphi')} &\text{(D.76)}\\ &= \lambda^{V(\varphi') \cap \mathcal{D} \cap V_L}(\varphi') \otimes \lambda^{(V(\varphi') \cap V_L) \setminus \mathcal{D}}(\varphi') \otimes \\ &\qquad \Lambda_{(\mathcal{D} \cap V_L) \setminus V(\varphi')} \otimes \Lambda_{V_L \setminus (\mathcal{D} \cup V(\varphi'))} \ . &\text{(D.77)}\end{aligned}$$

Note that no matching $\psi \in \mathcal{M}_{\mathcal{D} \cap V_L}$ covers any of the vertices in $V_L \setminus (\mathcal{D} \cup V(\varphi'))$. This holds as the algorithm can only add edges from the set $(\mathcal{D} \cap V_L) \times (V_H \cap V_R)$. Hence we can write

$$\lambda^{V_L}(\mathcal{M}_{\mathcal{D} \cap V_L}) = \lambda^{V_L \cap (\mathcal{D} \cup V(\varphi'))}(\mathcal{M}_{\mathcal{D} \cap V_L}) \otimes \Lambda_{V_L \setminus (\mathcal{D} \cup V(\varphi'))} \ . \qquad \text{(D.78)}$$

Thus we can ignore the space $\Lambda_{V_L \setminus (\mathcal{D} \cup V(\varphi'))}$ for the remainder of this argument. From the algorithm it should be evident that

$$\lambda^{(\mathcal{D} \cap V_L) \setminus V(\varphi')}(\mathcal{M}_{\mathcal{D} \cap V_L}) = \Lambda_{(\mathcal{D} \cap V_L) \setminus V(\varphi')} \qquad \text{(D.79)}$$

as every vertex in $v \in (\mathcal{D} \cap V_L) \setminus V(\varphi')$ is independently matched to every vertex in V'_v of size $|V'_v| \geq 1/2(\deg_G(v) - d_v + \delta_v/2)$. As the dimension of $\dim(\Lambda_v) = 1/2(\deg_G(v) - d_v + \delta_v/2)$, we conclude that $\Lambda_{(V_L \cap \mathcal{D}) \setminus V(\varphi')}$ is spanned.

To continue the argument, we need the following equivalence relation on matchings. Two matchings $\psi, \psi' \in \mathcal{M}_{\mathcal{D} \cap V_L}$ are equivalent on a vertex set V if they match the vertices in V in the same way, that is, for $v \in V$ we have that $\psi_v = \psi'_v$. We denote the equivalence class with respect to the vertex set V over $\mathcal{M}_{\mathcal{D} \cap V_L}$ of a matching $\psi \in \mathcal{M}_{\mathcal{D} \cap V_L}$ by $\{\psi\}_V$.

We want to show that for every $\psi \in \mathcal{M}_{\mathcal{D} \cap V_L}$ it holds that

$$\lambda^{V(\varphi') \cap \mathcal{D} \cap V_L}(\varphi') \subseteq \operatorname{span}(\lambda^{V(\varphi') \cap \mathcal{D} \cap V_L}(\psi') \mid \psi' \in \{\psi\}_{(\mathcal{D} \cap V_L) \setminus V(\varphi')}) \ . \qquad \text{(D.80)}$$

Note that in combination with D.79 we get that

$$\begin{aligned}\lambda^{\mathcal{D} \cap V_L}(\varphi') &= \lambda^{(\mathcal{D} \cap V_L) \setminus V(\varphi')}(\varphi') \otimes \lambda^{V(\varphi') \cap \mathcal{D} \cap V_L}(\varphi') &\text{(D.81)}\\ &= \Lambda_{(\mathcal{D} \cap V_L) \setminus V(\varphi')} \otimes \lambda^{V(\varphi') \cap \mathcal{D} \cap V_L}(\varphi') &\text{(D.82)}\\ &\subseteq \lambda^{\mathcal{D} \cap V_L}(\mathcal{M}_{\mathcal{D} \cap V_L}) \ . &\text{(D.83)}\end{aligned}$$

We prove D.80 by induction on subsets of $V(\varphi')\cap\mathcal{D}\cap V_L$. The statement clearly holds for the empty set. Fix $U \subseteq V(\varphi')\cap\mathcal{D}\cap V_L$ and a vertex $u \in U$. By induction, we may assume that

$$\lambda^{U\setminus\{u\}}(\varphi') \subseteq \operatorname{span}(\lambda^{U\setminus\{u\}}(\psi') \mid \psi' \in \{\psi\}_{(\mathcal{D}\cap V_L)\setminus V(\varphi')}) \ . \tag{D.84}$$

We want to show that the statement also holds for the set U. Note that $\lambda^{U}(\varphi') = \lambda^{U\setminus\{u\}}(\varphi') \otimes \lambda_u(\varphi'_u)$. Further,

$$\begin{align}
&\operatorname{span}(\lambda^{U}(\psi') \mid \psi' \in \{\psi\}_{(\mathcal{D}\cap V_L)\setminus V(\varphi')}) = \tag{D.85}\\
&\operatorname{span}(\lambda^{U\setminus\{u\}}(\psi') \otimes \\
&\qquad \operatorname{span}(\lambda_u(\eta) \mid \eta \in \{\psi'\}_{((\mathcal{D}\cap V_L)\setminus V(\varphi'))\cup(U\setminus\{u\})}) \mid \psi' \in \{\psi\}_{(\mathcal{D}\cap V_L)\setminus V(\varphi')}) \ . \tag{D.86}
\end{align}$$

Suppose that for every $\psi' \in \{\psi\}_{(V_L\cap\mathcal{D})\setminus V(\varphi')}$ it holds that

$$\lambda_u(\varphi'_u) \subseteq \operatorname{span}(\lambda_u(\eta_u) \mid \eta \in \{\psi'\}_{((\mathcal{D}\cap V_L)\setminus V(\varphi'))\cup(U\setminus\{u\})}) \ . \tag{D.87}$$

Then, continuing from above, we see that

$$\begin{align}
&\operatorname{span}(\lambda^{U}(\psi') \mid\psi' \in \{\psi\}_{(\mathcal{D}\cap V_L)\setminus V(\varphi')}) \tag{D.88}\\
&\qquad \supseteq \operatorname{span}(\lambda^{U\setminus\{u\}}(\psi') \mid \psi' \in \{\psi\}_{(\mathcal{D}\cap V_L)\setminus V(\varphi')}) \otimes \lambda_u(\varphi'_u) \tag{D.89}\\
&\qquad \supseteq \lambda^{U\setminus\{u\}}(\varphi') \otimes \lambda_u(\varphi'_u) \tag{D.90}\\
&\qquad = \lambda^{U}(\varphi') \ , \tag{D.91}
\end{align}$$

where the second inclusion holds by the induction hypothesis D.84. Thus, to show the statement for U we just need to show D.87. To this end, fix a matching $\psi' \in \{\psi\}_{(\mathcal{D}\cap V_L)\setminus V(\varphi')}$. Note that if there is a matching $\eta \in \{\psi'\}_{((\mathcal{D}\cap V_L)\setminus V(\varphi'))\cup(U\setminus\{u\})}$ such that $\eta_u = \varphi'_u$, then we are done. Otherwise, Algorithm 5 removed the edge that mached the vertex u in φ'. Hence the vertex u is matched by the procedure to at least $|V'_u| \geq 1/2(\deg_G(v) - d_v + \delta_v/2)$ different vertices. As the dimension of $\Lambda_u = 1/2(\deg_G(v) - d_v + \delta_v/2)$, we see that all of the space is spanned. We conclude that D.87 holds.

What remains is to argue that for every $\psi \in \mathcal{M}_{\mathcal{D}\cap V_L}$ it holds that

$$\lambda^{(V(\varphi')\cap V_L)\setminus\mathcal{D}}(\varphi') \subseteq \operatorname{span}(\lambda^{(V(\varphi')\cap V_L)\setminus\mathcal{D}}(\psi') \mid \psi' \in \{\psi\}_{\mathcal{D}\cap V_L}) \ . \tag{D.92}$$

The argument goes along the same lines as for the vertices in $V(\varphi')\cap\mathcal{D}\cap V_L$ and we thus omit it.

We can then combine D.81 and D.92 to conclude the claim. □

Observe that the matchings in $\mathcal{M}_{\mathcal{D}\cap V_L}$ are not necessarily extensions of φ'. This is not a problem, however, since the matchings only differ in edges that contain vertices which either do not show up in the linear space or for which the whole linear space associated to the vertex is spanned. Furthermore, vertices from $\mathcal{D} \cap V_H \cap V_L$ are matched to many vertices even though a single vertex would have been sufficient.

It remains only to show that every matching $\psi \in \mathcal{M}_{\mathcal{D}\cap V_L}$ can be extended in many ways to the set $\mathcal{D} \cap V_R$. Fix a matching $\psi \in \mathcal{M}_{\mathcal{D}\cap V_L}$ and recall that these are defined on $S_{01} \cap V_L$. Note that by Lemma D.5.8, property 2, each $v \in \mathcal{D} \cap V_R$ has at least

$$\left|N_G(v) \cap \underset{H}{V}\right| \geq 1/2\left|N_G(v)\right| - 4\xi\left|N_G(v)\right| \tag{D.93}$$

many neighbours in V_H. Using D.71 we can now bound the number of matchings that do not satisfy C.

Note that the matching ψ contains at most $|S_{01}| \leq r/2$ many edges. Since G is bipartite, this implies that for any $v \in V_R$ at most $r/2$ neighbours are already matched. Observe that some vertex $v \in \mathcal{D} \cap V_H \cap V_R$ may have been matched by Algorithm 5. As these vertices are not associated with a linear space, we only need to match these vertices with a single vertex and hence we can just leave them matched as in ψ. Further, by Claim D.5.14, we see that the vertices in $\mathcal{D} \cap V_P \cap V_R$ were not matched by Algorithm 5. All these will be matched in many ways as needed: If $v \in \mathcal{D} \cap V_R$ is not matched by ψ, then by D.93 and D.71 it can be matched to at least

$$\begin{aligned} 1/2(\left|N_G(v)\right| - 8\xi\left|N_G(v)\right| &- d_v + \delta_v - 8\xi\left|N_G(v)\right| - r) \\ &= 1/2\left(\deg_G(v) - d_v + \delta_v - 16\xi\deg_G(v) - r\right) \\ &\geq 1/2\left(\deg_G(v) - d_v + \delta_v - 17\xi\deg_G(v)\right) \\ &\geq 1/2\left(\deg_G(v) - d_v + \delta_v/2\right) \end{aligned} \tag{D.94}$$

many vertices without satisfying the clause C. Note that in D.94 we used the assumption that $\deg_G(v) \geq r\xi$ for $v \in V_R$. As we have that $\dim(\Lambda_v) = 1/2(\deg_G(v) - d_v + \delta_v/2)$, we conclude that the extensions of ψ can span the linear space $\Lambda_{(\mathcal{D}\cap V_R)\setminus V(\varphi')}$. Hence, by extending each $\psi \in \mathcal{M}_{\mathcal{D}\cap V_L}$, we get a set of matchings $\mathcal{M}_{\mathcal{D}}$, which do not satisfy the clause C, are defined on S_{01} and $\lambda(\varphi') \subseteq \lambda(\mathcal{M}_{\mathcal{D}})$. This establishes the lemma.

D.6 Concluding Remarks

In this work, we extend the pseudo-width method developed by Razborov [Raz03; Raz04b] for proving lower bounds on severely overconstrained CNF formulas in resolution. In particular, we establish that pigeonhole principle formulas and perfect matching formulas over highly unbalanced bipartite graphs remain exponentially hard for resolution even when these graphs are sparse. This resolves an open problem in [Raz04b].

The main technical difference in our work compared to [Raz03; Raz04b] goes right to the heart of the proof, where one wants to argue that resolution in small pseudo-width cannot make progress towards a derivation of contradiction. Here Razborov uses the global symmetry properties of the formula, whereas we resort to a local argument based on graph expansion. This argument needs to be carefully combined with a graph closure operation as in [AR03; ABRW04] to ensure that the residual graph always remains expanding as matched pigeons and their neighbouring holes are removed. It is this change of perspective that allows us to prove lower bounds for sparse bipartite graphs with the size m of the left-hand side (i.e., the number of pigeons) varying all the way from linear to exponential in the size n of the right-hand size (i.e., the number of pigeonholes), thus covering the full range between [BW01] on the one hand and [Raz04a; Raz03; Raz04b] on the other.

One shortcoming of our approach is that the sparse expander graphs are required to have very good expansion—for graphs of left degree Δ, the size of the set of unique neighbours of any not too large left vertex set has to scale like $(1 - o(1))\Delta$. We would like to prove that graph PHP formulas are hard also for graphs with constant expansion $(1 - \varepsilon)\Delta$ for some $\varepsilon > 0$, but there appear to be fundamental barriers to extending our lower bound proof to this setting.

Another intriguing problem left over from [Raz04b] is to determine the true resolution complexity of weak PHP formulas over complete bipartite graphs $K_{m,n}$ as $m \to \infty$. The best known upper bound from [BP97] is $\exp\bigl(O\bigl(\sqrt{n \log n}\bigr)\bigr)$, whereas the lower bound in [Raz03; Raz04b] is $\exp\bigl(\Omega\bigl(\sqrt[3]{n}\bigr)\bigr)$. It does not seem unreasonable to hypothesize that $\exp\bigl(\Omega\bigl(\sqrt[2]{n}\bigr)\bigr)$ should be the correct lower bound (ignoring lower-order terms), but establishing such a lower bound again appears to require substantial new ideas.

We believe that one of the main contributions of our work is that it again demonstrates the power of Razborov's pseudo-width method, and we are currently optimistic that it could be useful for solving other open problems for resolution and other proof systems.

For resolution, an interesting question mentioned in [Raz04b] is whether pseudo-width can be useful to prove lower bounds for formulas that encode the Nisan–Wigderson generator [ABRW04; Raz15]. Since the clauses in such formulas encode local constraints, we hope that techniques from our paper could be helpful. Another long-standing open problem is to prove lower bounds on proofs in resolution that k-clique free sparse graph do not contain k-cliques, where the expected length lower bound would be $n^{\Omega(k)}$. Here we only know weakly exponential lower bounds for quite dense random graphs [BIS07; Pan19], although an asymptotically optimal $n^{\Omega(k)}$ lower bound has been established in the sparse regime for the restricted subsystem of regular resolution [ABdR+18].

Finally, we want to highlight that for the stronger proof system *polynomial calculus* [ABRW02; CEI96] no lower bounds on proof size are known for PHP formulas with $m \geq n^2$ pigeons. It would be very interesting if some kind of "pseudo-degree" method could be developed that would finally lead to progress on this problem.

Acknowledgements

First and foremost, we are most grateful to Alexander Razborov for many discussions about pseudo-width, graph closure, and other mysteries of the universe. We also thank Paul Beame and Johan Håstad for useful discussions, and Jonah Brown-Cohen for helpful references on expander graphs. Finally, we gratefully acknowledge the feedback from participants of the Dagstuhl workshop 19121 *Computational Complexity of Discrete Problems* in March 2019.

The authors were funded by the Knut and Alice Wallenberg grant KAW 2016.0066. In addition, the second author was supported by the Swedish Research Council grant 2016-00782 and the Independent Research Fund Denmark (DFF) grant 9040-00389B, and the fourth author by the Knut and Alice Wallenberg grant KAW 2016.0433. Part of this work was carried out while visiting the Simons Institute for the Theory of Computing in association with the DIMACS/Simons Collaboration on Lower Bounds in Computational Complexity, which is conducted with support from the National Science Foundation.

References

[Ale04] M. Alekhnovich, "Mutilated chessboard problem is exponentially hard for resolution", *Theoretical Computer Science*, vol. 310, no. 1–3, pp. 513–525, Jan. 2004 (cit. on p. 217)

[ABRW02] M. Alekhnovich, E. Ben-Sasson, A. A. Razborov and A. Wigderson, "Space complexity in propositional calculus", *SIAM Journal on Computing*, vol. 31, no. 4, pp. 1184–1211, Apr. 2002, Preliminary version in *STOC '00* (cit. on p. 261)

[ABRW04] ——, ”Pseudorandom generators in propositional proof complexity”, *SIAM Journal on Computing*, vol. 34, no. 1, pp. 67–88, 2004, Preliminary version in *FOCS ’00* (cit. on pp. 218, 222, 229, 260)

[AR03] M. Alekhnovich and A. A. Razborov, ”Lower bounds for polynomial calculus: Non-binomial case”, *Proceedings of the Steklov Institute of Mathematics*, vol. 242, pp. 18–35, 2003. [Online]. Available: `http://people.cs.uchicago.edu/~razborov/files/misha.pdf` (cit. on pp. 218, 222, 229, 260)

[ABdR+18] A. Atserias, I. Bonacina, S. F. de Rezende, M. Lauria, J. Nordström and A. Razborov, ”Clique is hard on average for regular resolution”, in *Proceedings of the 50th Annual ACM Symposium on Theory of Computing (STOC ’18)*, Jun. 2018, pp. 866–877 (cit. on p. 261)

[BIS07] P. Beame, R. Impagliazzo and A. Sabharwal, ”The resolution complexity of independent sets and vertex covers in random graphs”, *Computational Complexity*, vol. 16, no. 3, pp. 245–297, Oct. 2007, Preliminary version in *CCC ’01* (cit. on p. 261)

[BP96] P. Beame and T. Pitassi, ”Simplified and improved resolution lower bounds”, in *Proceedings of the 37th Annual IEEE Symposium on Foundations of Computer Science (FOCS ’96)*, Oct. 1996, pp. 274–282 (cit. on p. 216)

[BW01] E. Ben-Sasson and A. Wigderson, ”Short proofs are narrow—resolution made simple”, *Journal of the ACM*, vol. 48, no. 2, pp. 149–169, Mar. 2001, Preliminary version in *STOC ’99* (cit. on pp. 216, 217, 260)

[BGL10] O. Beyersdorff, N. Galesi and M. Lauria, ”A lower bound for the pigeonhole principle in tree-like resolution by asymmetric prover-delayer games”, *Information Processing Letters*, vol. 110, no. 23, pp. 1074–1077, 2010. DOI: `10.1016/j.ipl.2010.09.007`. [Online]. Available: `https://doi.org/10.1016/j.ipl.2010.09.007` (cit. on p. 216)

[Bla37] A. Blake, ”Canonical expressions in Boolean algebra”, Ph.D. dissertation, University of Chicago, 1937 (cit. on p. 214)

[BP97] S. R. Buss and T. Pitassi, ”Resolution and the weak pigeonhole principle”, in *11th International Workshop on Computer Science Logic (CSL ’97), Selected Papers*, ser. Lecture Notes in Computer Science, vol. 1414, Springer, Aug. 1997, pp. 149–156 (cit. on pp. 216, 260)

[BT88] S. R. Buss and G. Turán, ”Resolution proofs of generalized pigeonhole principles”, *Theoretical Computer Science*, vol. 62, no. 3, pp. 311–317, Dec. 1988 (cit. on p. 216)

[CS88] V. Chvátal and E. Szemerédi, ”Many hard examples for resolution”, *Journal of the ACM*, vol. 35, no. 4, pp. 759–768, Oct. 1988 (cit. on p. 214)

[CEI96] M. Clegg, J. Edmonds and R. Impagliazzo, ”Using the Groebner basis algorithm to find proofs of unsatisfiability”, in *Proceedings of the 28th Annual ACM Symposium on Theory of Computing (STOC '96)*, May 1996, pp. 174–183 (cit. on p. 261)

[Dan02] S. S. Dantchev, ”Resolution width-size trade-offs for the pigeon-hole principle”, in *Proceedings of the 17th Annual IEEE Conference on Computational Complexity (CCC '02)*, May 2002, pp. 39–43. DOI: 10.1109/CCC.2002.1004337. [Online]. Available: https://doi.org/10.1109/CCC.2002.1004337 (cit. on p. 216)

[DR01a] S. S. Dantchev and S. Riis, ””Planar” tautologies hard for resolution”, in *Proceedings of the 42nd Annual IEEE Symposium on Foundations of Computer Science (FOCS '01)*, Oct. 2001, pp. 220–229 (cit. on p. 217)

[DR01b] ——, ”Tree resolution proofs of the weak pigeon-hole principle”, in *Proceedings of the 16th Annual IEEE Conference on Computational Complexity (CCC '01)*, Jun. 2001, pp. 69–75. DOI: 10.1109/CCC.2001.933873. [Online]. Available: https://doi.org/10.1109/CCC.2001.933873 (cit. on p. 216)

[GUV09] V. Guruswami, C. Umans and S. Vadhan, ”Unbalanced expanders and randomness extractors from Parvaresh–Vardy codes”, *Journal of the ACM*, vol. 56, no. 4, 20:1–20:34, Jul. 2009, Preliminary version in CCC '07 (cit. on pp. 226, 243, 245)

[Hak85] A. Haken, ”The intractability of resolution”, *Theoretical Computer Science*, vol. 39, no. 2-3, pp. 297–308, Aug. 1985 (cit. on pp. 214, 216)

[IOSS16] D. Itsykson, V. Oparin, M. Slabodkin and D. Sokolov, ”Tight lower bounds on the resolution complexity of perfect matching principles”, *Fundamenta Informaticae*, vol. 145, no. 3, pp. 229–242, Aug. 2016. DOI: 10.3233/FI-2016-1358. [Online]. Available: https://doi.org/10.3233/FI-2016-1358 (cit. on p. 217)

[Pan19] S. Pang, ”Large clique is hard on average for resolution”, Electronic Colloquium on Computational Complexity (ECCC), Technical Report TR19-068, Apr. 2019 (cit. on p. 261)

[PR04] T. Pitassi and R. Raz, ”Regular resolution lower bounds for the weak pigeonhole principle”, *Combinatorica*, vol. 24, no. 3, pp. 503–524, 2004, Preliminary version in *STOC '01*. DOI: 10.1007/s00493-004-0030-y. [Online]. Available: https://doi.org/10.1007/s00493-004-0030-y (cit. on p. 216)

[Raz04a] R. Raz, ”Resolution lower bounds for the weak pigeonhole principle”, *Journal of the ACM*, vol. 51, no. 2, pp. 115–138, Mar. 2004, Preliminary version in *STOC '02* (cit. on pp. 216, 260)

[Raz98] A. A. Razborov, ”Lower bounds for the polynomial calculus”, *Computational Complexity*, vol. 7, no. 4, pp. 291–324, Dec. 1998 (cit. on p. 215)

[Raz01] ——, ”Improved resolution lower bounds for the weak pigeonhole principle”, Electronic Colloquium on Computational Complexity (ECCC), Technical Report TR01-055, Jul. 2001 (cit. on pp. 216, 230, 241)

[Raz02] ——, ”Proof complexity of pigeonhole principles”, in *5th International Conference on Developments in Language Theory, (DLT '01), Revised Papers*, ser. Lecture Notes in Computer Science, vol. 2295, Springer, Jul. 2002, pp. 100–116 (cit. on p. 216)

[Raz03] ——, ”Resolution lower bounds for the weak functional pigeonhole principle”, *Theoretical Computer Science*, vol. 1, no. 303, pp. 233–243, Jun. 2003 (cit. on pp. 216–219, 221, 226, 243, 254, 259, 260)

[Raz04b] ——, ”Resolution lower bounds for perfect matching principles”, *Journal of Computer and System Sciences*, vol. 69, no. 1, pp. 3–27, Aug. 2004, Preliminary version in *CCC '02* (cit. on pp. 215–218, 241, 242, 247, 259, 260)

[Raz15] ——, ”Pseudorandom generators hard for k-DNF resolution and polynomial calculus resolution”, *Annals of Mathematics*, vol. 181, no. 2, pp. 415–472, Mar. 2015 (cit. on p. 260)

[RWY02] A. A. Razborov, A. Wigderson and A. C. Yao, ”Read-once branching programs, rectangular proofs of the pigeonhole principle and the transversal calculus”, *Combinatorica*, vol. 22, no. 4, pp. 555–574, 2002, Preliminary version in *STOC '97*. DOI: 10.1007/s00493-002-0007-7. [Online]. Available:

https://doi.org/10.1007/s00493-002-0007-7 (cit. on p. 216)

[Urq87] A. Urquhart, ”Hard examples for resolution”, *Journal of the ACM*, vol. 34, no. 1, pp. 209–219, Jan. 1987 (cit. on p. 214)

[Urq03] ——, ”Resolution proofs of matching principles”, *Annals of Mathematics and Artificial Intelligence*, vol. 37, no. 3, pp. 241–250, Mar. 2003. DOI: 10.1023/A:1021231610627. [Online]. Available: https://doi.org/10.1023/A:1021231610627 (cit. on p. 216)

[Urq07] ——, ”Width versus size in resolution proofs”, *Theoretical Computer Science*, vol. 384, no. 1, pp. 104–110, Sep. 2007 (cit. on p. 217)