Electronic Colloquium on Computational Complexity, Report No. 10 (2023)

Sum-of-Squares Lower Bounds for the Minimum Circuit Size Problem[*]

Per Austrin

KTH Royal Institute of Technology

Kilian Risse

EPFL

February 13, 2023

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 $s > \mathsf{poly}(n)$). As a corollary we obtain that there are no low degree SoS proofs of the statement $\mathbf{NP} \not\subseteq \mathbf{P}/\mathsf{poly}$.

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.

Our approach is quite general. Namely, we show that if a proof system Q has strong enough constraint satisfaction problem lower bounds that only depend on good expansion of the constraint-variable incidence graph and, furthermore, Q is expressive enough that variables can be substituted by local Boolean functions, then the MCSP problem is hard for Q.

[*]Supported by the Approximability and Proof Complexity project funded by the Knut and Alice Wallenberg Foundation and the Swiss National Science Foundation project 200021-184656 “Randomness in Problem Instances and Randomized Algorithms”.

ISSN 1433-8092

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** $\neq$ **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 $\mathsf{Circuit}_s(f)$ 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.

A closely related question to the MCSP problem is the question of how hard it is to actually prove strong circuit lower bounds. For example, are there efficient refutations of the statement $\mathbf{NP} \subseteq \mathbf{P}/\mathsf{poly}$, assuming the statement is false? This question, as proposed by Razborov [Raz98], can also be investigated by studying above formula: consider $\mathsf{Circuit}_{n^{O(1)}}(\mathsf{SAT})$, where SAT is the function that outputs 1 if and only if the input is an encoding of a satisfiable CNF. This is, essentially, a propositional encoding of the claim that SAT has a circuit in $\mathbf{P}/\mathsf{poly}$. Hence proving lower bounds for $\mathsf{Circuit}_{n^{O(1)}}(\mathsf{SAT})$ rules out efficient proofs of $\mathbf{NP} \not\subseteq \mathbf{P}/\mathsf{poly}$ in the proof system under consideration.

Experience suggests that studying such meta-mathematical questions is difficult. This problem is no exception to this rule and, even though the formula has been conjectured to be hard for strong proof systems such as extended Frege, progress has been slow. The only proof systems for which we have unconditional, superpolynomial lower bounds on proofs of the $\mathsf{Circuit}_s(f)$ formula are Resolution [Raz04a, Raz04b], small width DNF-Resolution [Raz15] and Polynomial Calculus [Raz98, Raz15]. The resolution size and Polynomial Calculus degree lower bounds follow from a reduction of the pigeonhole principle to $\mathsf{Circuit}_s(f)$. In fact, this reduction was a main motivation for a long line of work [RWY02, PR04, Raz04a, Raz04b] eventually establishing strong resolution lower bounds for the weak pigeonhole principle. The other size lower bounds follow from a general connection between pseudo-random generator lower bounds and MCSP lower bounds as outlined by Razborov [Raz15].

As the pigeonhole principle is easy for the SoS proof system [GHP02], we cannot hope to borrow the hardness from that formula. Neither do we have strong enough pseudorandom generator lower bounds for SoS to employ that connection. In fact, to date, we have no unconditional (degree) lower bounds for any semi-algebraic proof system, that is, proof systems that manipulate polynomial inequalities such as SoS or Cutting Planes. Furthermore it has been stated [Raz21, Raz22] as an explicit open problem to prove SoS degree lower bounds for the formula $\mathsf{Circuit}_s(f)$.

1.1 Our Results

Our first result gives a lower bound on the degree needed to refute $\mathsf{Circuit}_s(f)$ in SoS. This lower bound is very general and in fact applies to *every* Boolean function $f : \{0,1\}^n \to \{0,1\}$.

Theorem 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* $\mathsf{Circuit}_s(f)$.

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 1.2. *Let $s \in N$ 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* $\mathsf{Circuit}_s(f)$.

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 1.3. *For all $\varepsilon > 0$ there is a $d = d(\varepsilon)$ such that the following holds. Let $n \in N$ 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 $\mathrm{Circuit}_s^{\mathsf{mon}}(f)$ of the $\mathrm{Circuit}_s(f)$ formula, which instead encodes the claim that f has a monotone circuit of size s, and prove the following theorem.

Theorem 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* $\mathrm{Circuit}_s^{\mathsf{mon}}(f)$.

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 (not necessarily monotone) Boolean circuit $C_f^{\mathsf{mon}} : \{0,1\}^n \to \{0,1,\perp\}$ of size s such that

1. for all ℓ-slice inputs $\alpha \in \binom{[n]}{\ell}$ it holds that if $C_f^{\mathsf{mon}}(\alpha) \neq \perp$, then $C_f^{\mathsf{mon}}(\alpha) = f(\alpha)$, and
2. $C_f^{\mathsf{mon}}(\alpha) = \perp$ on at most t inputs $\alpha \in \binom{[n]}{\ell}$.

Theorem 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* $\mathrm{Circuit}_s^{\mathsf{mon}}(f)$.

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 $\mathrm{Circuit}_s(f)$ formula. The reduction is achieved through a careful restriction of the $\mathrm{Circuit}_s(f)$ 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 2.6). 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 2.2). This reduction then immediately yields Theorem 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.

It is worthwhile to point out that this reduction is not specific to the SoS proof system. In fact we just outline a very general reduction that shows that if one has a CSP lower bound of the form of Theorem 2.6 that only requires good expansion of the underlying constraint-variable incidence graph and the proof system is expressive enough so that one can replace variables by local Boolean functions, then one obtains strong lower bounds for the $\mathrm{Circuit}_s(f)$ formula.

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 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 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 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 1.3.

1.3 Organization

In Section 2, we provide the necessary background material. In Section 3 we set up the general framework for our lower bounds with some preliminary definitions and lemmas. Then in Section 4 we prove the main degree Theorem 1.1 and size Theorem 1.3 lower bounds. We prove the monotone lower bounds Theorem 1.4 and Theorem 1.5 in Section 5. In Section 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 7 we give some concluding remarks.

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 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 $\mathsf{poly}(\log n + \log |U| + \log d)$.

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 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 $\mathbf{Neigh}(u, v)$ indicating whether $(u, v) \in E$ rather than the neighbor function as in Definition 2.1, but this is an immediate consequence of being able to compute the neighbor function.

Claim 2.3. *If $G = (U, V, E)$ is explicit then the neighbor relation $\mathbf{Neigh} : U \times V \to \{0,1\}$ is computable by a circuit of size $d \cdot \big(\mathsf{poly}(\log n + \log |U| + \log d) + 2 \log |V| + 1\big)$.*

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 $\mathcal{C}(f)$ 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 $\mathcal{C}_{\mathsf{mon}}(f)$ 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 2.4 ([Ber82]). *If g is any slice function on n bits, then $\mathcal{C}_{\mathsf{mon}}(g) \leq 2\,\mathcal{C}(g) + O(n^2 \log n)$.*

Finally we also rely on the following simple claim.

Claim 2.5. *Let $p : \mathbb{R}^n \to \mathbb{R}$ be a degree d polynomial such that $p(x) = 0$ for all $x \in \{0,1\}^n$. Then p can be written as*

$$p(x) = \sum_{i \in [n]} q_i(x) \cdot (x_i^2 - x_i)$$

where each term in the sum has degree at most d.

Proof sketch. We take the polynomial p and multilinearize it, using the appropriate polynomial $x_i^2 - x_i$. Eventually we are left with a sum of polynomials of the form $q_i(x) \cdot (x_i^2 - x_i)$ and a multilinear polynomial $\tilde{p}(x)$ which is 0 on all Boolean inputs. As multilinear polynomials are a basis for Boolean functions this implies that $\tilde{p}(x)$ is equal to the 0 polynomial and hence the claim follows. □

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

$$\sum_{i \in [m]} t_i p_i + \sum_{i \in [a]} s_i^2 = f \quad . \tag{1}$$

The *degree* of a proof π is

$$\mathsf{Deg}(\pi) = \max\{\max_{i \in [m]} \deg(t_i) + \deg(p_i), \max_{i \in [a]} 2\deg(s_i)\} \quad . \tag{2}$$

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 $\mathsf{Deg}(\mathcal{P} \vdash_{\mathsf{SoS}} \bot) = \min_\pi \mathsf{Deg}(\pi)$.

The size of an SoS refutation π, $\mathsf{Size}(\pi)$, 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 $\mathsf{Size}(\mathcal{P} \vdash_{\mathsf{SoS}} \bot) = \min_\pi \mathsf{Size}(\pi)$.

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 encode 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 2.6 ([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 2.7 ([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* $\exp(\Omega((d-k)^2/n))$.

2.2 Restrictions

Let $\mathcal{P} = \{p_1 = 0, \ldots, p_m = 0\}$ be a system of polynomial equations over the set of Boolean variables $X = \{x_1, \ldots, x_n, \bar{x}_1, \ldots, \bar{x}_n\}$. For a map $\rho : \{x_1, \ldots, x_n\} \to \{0, 1, x_1, \ldots, x_n, \bar{x}_1, \ldots, \bar{x}_n\}$ denote by $\mathcal{P}\big|_\rho$ the system of polynomial equations $\mathcal{P}$ restricted by ρ, i.e.,

$$\begin{aligned} \mathcal{P}\big|_\rho = \{ & p_1(\rho(x_1), \ldots, \rho(x_n)) = 0, \\ & p_2(\rho(x_1), \ldots, \rho(x_n)) = 0, \\ & \vdots \\ & p_m(\rho(x_1), \ldots, \rho(x_n)) = 0\} \quad , \end{aligned}$$

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 $\rho : \{x_1, \ldots, x_n\} \to \{0, 1, x_1, \ldots, x_n, \bar{x}_1, \ldots, \bar{x}_n\}$ such that $\mathcal{P}' = \mathcal{P}\big|_\rho$, 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 2.8. *Let $\mathcal{P}, \mathcal{P}'$ be systems of polynomial equations such that $\mathcal{P}'$ is an affine restriction of $\mathcal{P}$. Then,*

(i) $\mathsf{Deg}(\mathcal{P} \vdash_{SoS} \bot) \geq \mathsf{Deg}(\mathcal{P}' \vdash_{SoS} \bot)$*, and*

(ii) $\mathsf{Size}(\mathcal{P} \vdash_{SoS} \bot) \geq \mathsf{Size}(\mathcal{P}' \vdash_{SoS} \bot)$.

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 $\rho : \{x_1, \ldots, x_n\} \to \mathbb{R}[x]_{\leq k}$ that map variables to polynomials of degree at most k. We call such restrictions "polynomial substitutions" to differentiate them from affine restrictions.

For such polynomial substitutions we have the following well-known lemma.

Lemma 2.9. *Let $\mathcal{P}$ be a system of polynomial equations and let ρ be a polynomial substitution mapping variables to polynomials of degree at most k. Then,* $\mathsf{Deg}(\mathcal{P} \vdash_{SoS} \bot) \geq \mathsf{Deg}(\mathcal{P}\big|_\rho \vdash_{SoS} \bot)/k$*.*

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 $\mathcal{P}\big|_\rho$, whose degree is at most a factor k larger than the degree of the refutation of $\mathcal{P}$.

2.3 The Circuit Size Formula

The formula $\text{Circuit}_s(f)$ 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, Raz04b] 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 Appendix A 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 (see Proposition A.1). 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 $\text{Circuit}_s(f)$ denoted by $\text{Circuit}_s^{\mathsf{mon}}(f)$. The later is a restriction of the former with the $\mathsf{IsNeg}(v)$ (see below) variable, for all $v \in [s]$, 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 $\mathsf{IsNeg}(v), \mathsf{IsOr}(v), \mathsf{IsAnd}(v)$ indicating the operation computed at v. Similarly for a gate $v \in [s]$ and a wire $a \in \{1, 2\}$ we have variables $\mathsf{IsFromConst}(v, a), \mathsf{IsFromInput}(v, a), \mathsf{IsFromGate}(v, a)$ indicating whether the input wire a of v is connected to a constant, a variable or a gate.

Further, we have the variables $\mathsf{ConstantValue}(v, a)$, $\mathsf{IsInput}(v, a, i)$ and $\mathsf{IsGate}(v, a, u)$, 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 $\mathsf{Out}_\alpha(v)$ indicating the value computed at gate v on input α. The Boolean variable $\mathsf{In}_\alpha(v, a)$ 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

$$\mathsf{IsFromConst}(v, a) + \mathsf{IsFromInput}(v, a) + \mathsf{IsFromGate}(v, a) = 1 \quad \forall\, v \in [s] \ , \tag{3}$$

and similarly the next axioms make sure that each gate is of precisely one kind

$$\mathsf{IsNeg}(v) + \mathsf{IsOr}(v) + \mathsf{IsAnd}(v) = 1 \quad \forall\, v \in [s] \ . \tag{4}$$

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)

$$\sum_{i=1}^{n} \mathsf{IsInput}(v, a, i) = 1 \quad \forall v \in [s], \text{ and} \tag{5}$$

$$\sum_{u=1}^{v-1} \mathsf{IsGate}(v, a, u) = 1 \quad \forall v \in [s] \setminus \{1\} \ . \tag{6}$$

We further strengthen our encoding by adding the axioms

$$\mathsf{IsInput}(v, a, i)\, \mathsf{IsInput}(v, a, j) = 0 \quad \forall v \in [s],\ i < j \in [n], \text{ and} \tag{7}$$

$$\mathsf{IsGate}(v, a, u)\, \mathsf{IsGate}(v, a, u') = 0 \quad \forall v \in [s] \setminus \{1\},\ u < u' < v \ . \tag{8}$$

Note that Axioms 7 and 8 are implied by Axioms 5 and 6. 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

$$\mathsf{IsFromConst}(v, a) \cdot \big(\mathsf{In}_\alpha(v, a) - \mathsf{ConstantValue}(v, a)\big) = 0 \ , \tag{9}$$

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

$$\mathsf{IsFromInput}(v,a) \cdot \mathsf{IsInput}(v,a,i) \cdot \big(\mathsf{In}_\alpha(v,a) - \alpha_i \big) = 0 \ , \tag{10}$$

$$\mathsf{IsFromGate}(v,a) \cdot \mathsf{IsGate}(v,a,u) \cdot \big(\mathsf{In}_\alpha(v,a) - \mathsf{Out}_\alpha(u) \big) = 0 \ . \tag{11}$$

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

$$\mathsf{IsNeg}(v) \cdot \mathsf{Out}_\alpha(v) = \mathsf{IsNeg}(v) \cdot \overline{\mathsf{In}_\alpha(v,1)} \ , \tag{12}$$

$$\mathsf{IsOr}(v) \cdot \mathsf{Out}_\alpha(v) = \mathsf{IsOr}(v) \cdot \big(1 - \overline{\mathsf{In}_\alpha(v,1)} \cdot \overline{\mathsf{In}_\alpha(v,2)}\big) \ , \tag{13}$$

$$\mathsf{IsAnd}(v) \cdot \mathsf{Out}_\alpha(v) = \mathsf{IsAnd}(v) \cdot \mathsf{In}_\alpha(v,1) \cdot \mathsf{In}_\alpha(v,2) \ . \tag{14}$$

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

$$\mathsf{Out}_\alpha(s) = f(\alpha) \ . \tag{15}$$

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 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

$$\mathcal{F}_G = \{ f_{G,\beta} \mid \beta \in \{0,1\}^m \} \ .$$

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 $\mathsf{Circuit}_s(f)$ 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

$$\mathcal{C}_\emptyset = \{ T : \{0,1\}^n \to \{0,1\} \text{ such that } \mathsf{Circuit}_s(T) \text{ is satisfiable} \} \ .$$

More generally, if we restrict $\mathsf{Circuit}_s(f)$ by a restriction ρ, then the proof system must prove that f is not a member of the family of truthtables

$$\mathcal{C}_\rho = \{ T : \{0,1\}^n \to \{0,1\} \text{ such that } \mathsf{Circuit}_s(T)\big|_\rho \text{ is satisfiable} \} \ .$$

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 restricted formula $\mathsf{Circuit}_s(f)\big|_\rho$, SoS needs 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 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 next 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 some of the evaluation variables). Some set of m structure variables remains undetermined. Let us denote these variables by $y_1, \ldots, y_m$. 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 in the y variables.

Recall from Section 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 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 $Y = \{y_1, \ldots, y_m\}$ unset. Then ρ is *k-determined* if for every $v \in [s]$ and $\alpha \in \{0,1\}^n$ there are multilinear polynomials

$$g^{\mathrm{out}}_{v,\alpha}, g^{\mathrm{in}_1}_{v,\alpha}, g^{\mathrm{in}_2}_{v,\alpha} : \{0,1\}^m \to \{0,1\}$$

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 $\mathrm{Circuit}_s(T)\big|_\rho$ it holds that

$$\mathsf{Out}_\alpha(v)\big|_{\rho\cup\sigma} = g^{\mathrm{out}}_{v,\alpha}(\beta) \ , \quad \mathsf{In}_\alpha(v,1)\big|_{\rho\cup\sigma} = g^{\mathrm{in}_1}_{v,\alpha}(\beta) \ , \text{ and } \quad \mathsf{In}_\alpha(v,2)\big|_{\rho\cup\sigma} = g^{\mathrm{in}_2}_{v,\alpha}(\beta) \ , \qquad (16)$$

where $\beta \subseteq \sigma$ is the assignment to Y.

However, Definition 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 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 $Y = \{y_1, \ldots, y_m\}$ unset, and for every assignment $\beta \in \{0,1\}^Y$ it holds that $|\mathcal{C}_{\rho\cup\beta}| = 1$.

With these definitions at hand we can state the lemma that drives all our lower bounds.

Lemma 3.4. *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 3.2. If there is an SoS refutation of* $\mathrm{Circuit}_s(f)\big|_\rho$ *of degree d, then there is a degree $d \cdot k$ SoS refutation of the system of polynomial equations*

$$\{g^{\mathrm{out}}_{s,\alpha}(Y) = f(\alpha) \mid \alpha \in \{0,1\}^n\} \cup \{y_i^2 = y_i \mid i \in [m]\} \ . \qquad (17)$$

For proving this lemma, we consider the natural extension of ρ which substitutes all evaluation variables by appropriate degree-k polynomials as indicated by Definition 3.2.

Definition 3.5. For a k-determined restriction ρ (with associated polynomials $g^{\mathrm{out}}_{v,\alpha}, g^{\mathrm{in}_1}_{v,\alpha}, g^{\mathrm{in}_2}_{v,\alpha}$) of $\mathrm{Circuit}_s(f)$, we denote by $\hat{\rho}$ the polynomial substitution that extends ρ by first substituting any bar variable $\bar{x}$ by $1 - x$ and then substituting all evaluation variables as follows:

$$\hat{\rho}(\mathsf{Out}_\alpha(v)) = g^{\mathrm{out}}_{v,\alpha}(Y) \ , \qquad \hat{\rho}(\mathsf{In}_\alpha(v,1)) = g^{\mathrm{in}_1}_{v,\alpha}(Y) \ , \text{ and } \qquad \hat{\rho}(\mathsf{In}_\alpha(v,2)) = g^{\mathrm{in}_2}_{v,\alpha}(Y) \ .$$

Note that the formula $\mathrm{Circuit}_s(f)\big|_{\hat{\rho}}$ is defined only over Y. Let us stress that there are no "bar" variables left in the formula. The main observation used to prove Lemma 3.4 is the following claim, which establishes that the formula (17) is in fact essentially the same as $\mathrm{Circuit}_s(f)\big|_{\hat{\rho}}$.

Claim 3.6. *Let ρ be a natural m-independent k-determined affine restriction of* $\mathrm{Circuit}_s(f)$. *Then* $\mathrm{Circuit}_s(f)\big|_{\hat{\rho}}$ *can be written as*

$$\mathrm{Circuit}_s(f)\big|_{\hat{\rho}} = \mathcal{P} \cup \mathcal{Q} \ ,$$

where $\mathcal{P}$ is the formula (17) *and $\mathcal{Q}$ only consists of axioms that are satisfied for all assignments $\beta \in \{0,1\}^Y$.*

Proof. Note that the set of output axioms (15) of $\mathrm{Circuit}_s(f)$ under $\hat{\rho}$ equals the first part of (17), and that the Boolean axioms on the Y variables in $\mathrm{Circuit}_s(f)\big|_{\hat{\rho}}$ are exactly the second part of (17).

The remaining axioms of $\mathrm{Circuit}_s(f)\big|_{\hat{\rho}}$, which are not present in (17), are the Boolean axioms on the variables outside Y, the negation axioms, as well as Axioms 3 to 14.

The Boolean axioms may turn into polynomials of degree at most $2k$. Because the polynomials we substitute the variables with are Boolean valued, we see that these substituted axioms are satisfied for all assignments $\beta \in \{0,1\}^Y$ and we can thus put them into the set $\mathcal{Q}$.

The negation axioms all become "$0 = 0$" under $\hat{\rho}$ since $\hat{\rho}(\bar{x}) = 1 - \hat{\rho}(x)$.

Finally we need to argue that the Axioms 3 to 14 are also of the form $p(Y) = 0$ for a polynomial p which is identically 0 on all of $\{0,1\}^m$. This in turn follows immediately from the assumption that ρ is m-independent: for every $\beta \in \{0,1\}^Y$, there exists some T such that the complete assignment $\hat{\rho}(\beta) \cup \beta$ satisfies $\mathrm{Circuit}_s(T)$. But since none of the remaining Axioms 3 to 14 depends on T, they must then all be satisfied for every $\beta \in \{0,1\}^Y$. □

Using this claim we can easily prove Lemma 3.4.

Proof of Lemma 3.4. Suppose $\mathrm{Circuit}_s(f)\big|_{\rho}$ has a refutation in degree d. By Lemma 2.9, there then exists a degree $d \cdot k$ refutation of $\mathrm{Circuit}_s(f)\big|_{\hat{\rho}}$.

By Claim 3.6, this new refutation is *almost* a refutation of (17), except that $\mathrm{Circuit}_s(f)\big|_{\hat{\rho}}$ has an additional set $\mathcal{Q}$ of axioms that the refutation may use. However, each of these additional axioms is of the form $p(Y) = 0$ for a polynomial which is identically 0 on the entire Boolean cube. By Claim 2.5, such an axiom can be rewritten as a linear combination of the Boolean axioms. Since the Boolean axioms are present in (17), this yields a refutation of that formula in degree $d \cdot k$. □

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 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*

$$g^{\mathrm{out}}_{s,\alpha}(Y) = \begin{cases} f(\alpha), & \text{if } C_f(\alpha) \neq \perp, \\ \bigoplus_{i \in N(\alpha)} y_i, & \text{otherwise} \end{cases}$$

for all $\alpha \in \{0,1\}^n$ and $g^{\mathrm{out}}_{s,\alpha}$ and Y as in Definition 3.2. Furthermore, the formula $\mathrm{Circuit}_s(f)\big|_\rho$ *is over $O\big(t \cdot k + m\big)$ variables.*

For the degree lower bound (Theorem 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^1_f and C^2_f, where $C^1_f(\alpha) = 1$ if and only if $C^2_f(\alpha) = f(\alpha)$.

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 $\mathsf{Sel}_1, \ldots, \mathsf{Sel}_m$, where given any $\alpha \in \{0,1\}^n$, $\mathsf{Sel}_i(\alpha)$ is 1 if and only if the vertex $y_i \in Y$ is a neighbor of the vertex α. By Claim 2.3 these circuits Sel_i are each of size

$$k \cdot (\mathsf{poly}(n + \log k) + 2\log m + 1) \leq \mathsf{poly}(n,k) \ .$$

The restriction ρ restricts some structural variables such that a part of the circuit computes $\mathsf{Sel}_1, \ldots, \mathsf{Sel}_m$. We connect each output of the Sel_i circuit by an *and* gate to the negation of C^1_f. Denote the resulting circuits by $\mathsf{Sel}'_1, \ldots, \mathsf{Sel}'_m$. Observe that the circuits Sel'_i output 0

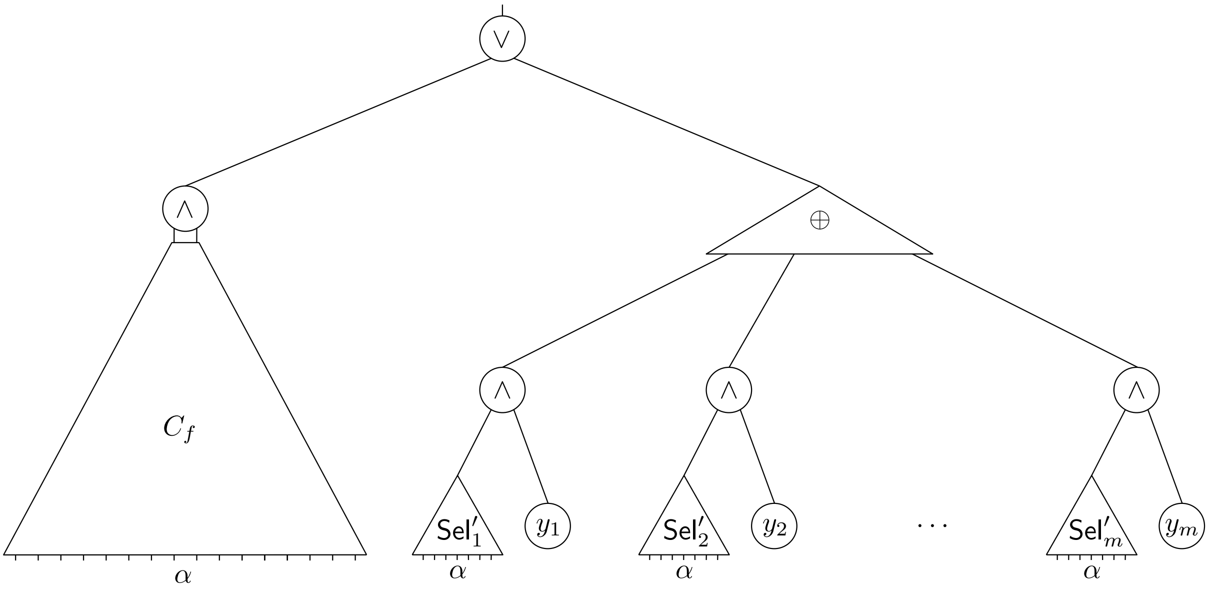

Figure 1: A schematic depiction of the formula after hitting it with the described restriction.

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 $x_1, \ldots, x_n$ 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 connected 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 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

$$\bigoplus_{i \in N(\alpha)} y_i \ , \tag{18}$$

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 $O\big(m \cdot \mathsf{poly}(n,k)\big) + s/2$.

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 $\chi_1 = \mathsf{Sel}'_1 \wedge y_1$ and $\chi_i = \chi_{i-1} \oplus (\mathsf{Sel}'_i \wedge y_i)$ for $i > 1$, and χ_m is the overall output of the $\oplus$ circuit.

The only observation required is that if the circuit $\mathsf{Sel}'_i(\alpha) = 0$, 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 $C_f(\alpha) = \bot$, 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 $\bot$ 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 ρ. The only part that remains is to verify that ρ is natural, k-determined, and m-independent. That ρ is natural is immediate – it does not substitute any structural variable by an evaluation variable. For k-determinedness, note that for a fixed input α at most k selector circuits output 1, and thus for every gate u the value of $\mathsf{Out}_\alpha(u)$ as a function of Y can be computed by a function over those k variables. Finally, each assignment to the remaining structure variables Y gives a valid circuit and thus ρ is m-independent. □

We are ready to prove the degree lower bound, restated here for convenience.

Theorem 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* $\mathsf{Circuit}_s(f)$.

Proof. Let $G = (U, V, E)$ be an explicit bipartite graph as in Theorem 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 $r \leq 2^n$ to be fixed later. Apply Lemma 4.1 with $t = 2^n$ along with $C_f = \bot$ to obtain, for $s \geq m \cdot \mathsf{poly}(n,k)$, a natural m-independent k-determined affine restriction ρ for $\mathsf{Circuit}_s(f)$ 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 3.4 to ρ to conclude that if there is an SoS refutation of $\mathsf{Circuit}_s(f)\big|_\rho$ of degree d, then there is a degree $d \cdot k$ SoS refutation of the system of polynomial equations computing

$$\mathcal{P}_G = \Big\{ \bigoplus_{i \in N(\alpha)} y_i = f(\alpha) : \alpha \in \{0,1\}^n \Big\} \cup \{ y_i^2 = y_i \mid i \in [m] \} \ .$$

As the graph G is a strong expander, we can apply Theorem 2.6 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 $\mathsf{Circuit}_s(f)\big|_\rho$ 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 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 $k = \mathsf{poly}_\gamma(n)$, where we use that $r \leq 2^n$, and we write $\mathsf{poly}_\gamma(n)$ to denote some polynomial in n whose degree and coefficients may depend on γ. Hence we may choose

$$m = \frac{s}{\mathsf{poly}_\gamma(n)} \ , \tag{19}$$

according to the requirement on s in Lemma 4.1. From the guarantees of Theorem 2.2 we know that $r \geq (m/k^2)^{1/(1+\gamma)}$. Substituting m according to the previous equation we get that

$$r \geq \left(\frac{s}{k^2 \mathsf{poly}_\gamma(n)} \right)^{\frac{1}{1+\gamma}} = \frac{s^{1/(1+\gamma)}}{\mathsf{poly}_\gamma(n)} \ . \tag{20}$$

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 $\mathsf{poly}_\gamma(n)$ is at most $s^{\varepsilon/2}$, we obtain the claimed lower bound. □

In the following we prove the claimed size lower bound.

Theorem 1.3. *For all $\varepsilon > 0$ there is a $d = d(\varepsilon)$ such that the following holds. Let $n \in N$ 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 4.1 with the graphs from Theorem 2.2 as in the proof of Theorem 1.1. We get a natural m-independent k-determined affine restriction ρ and the formula $\mathrm{Circuit}_s(f)\big|_\rho$ over $O(t \cdot k + m)$ variables. To this formula we then apply Lemma 3.4 to obtain a degree lower bound of $\Omega(r/k)$, akin to the proof of Theorem 1.1. By setting the parameters as in the aforementioned proof we get the same degree lower bound of $\Omega_\varepsilon(s^{1-\varepsilon/3})$ for the formula $\mathrm{Circuit}_s(f)\big|_\rho$. As this formula is over few variables we can apply Theorem 2.7 to obtain an SoS size lower bound of $\exp\left(\Omega_\varepsilon\left((s^{1-\varepsilon/3} - 3k)^2/(t \cdot k + m)\right)\right)$ 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 $s^{\varepsilon/3} \geq k = \mathsf{poly}_\varepsilon(n)$ and by recalling that $t \geq s \geq m$. □

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 have $|\mathcal{M}_n(\ell)| = 2^{\binom{n}{\ell}}$. 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 (not necessarily monotone) Boolean circuit $C_f^{\mathsf{mon}} : \{0,1\}^n \to \{0,1,\perp\}$ of size s such that

1. for all ℓ-slice inputs $\alpha \in \binom{[n]}{\ell}$ it holds that if $C_f^{\mathsf{mon}}(\alpha) \neq \perp$, then $C_f^{\mathsf{mon}}(\alpha) = f(\alpha)$, and
2. $C_f^{\mathsf{mon}}(\alpha) = \perp$ on at most t inputs $\alpha \in \binom{[n]}{\ell}$.

It is very convenient to work with slice functions as we have a handle on their monotone circuit complexity: by Lemma 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 4.1. The idea is to work over the ℓth slice and disregard all other inputs. By Lemma 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 2.4 will be useful to us.

Claim 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 $2s + \mathsf{poly}(n)$ 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 $\mathsf{poly}(n)$. We apply Lemma 2.4 to conclude that there is a monotone circuit C^{mon} computing the same function as C' of size $2s + \mathsf{poly}(n)$. □

Before stating the following lemma we need to adapt some terminology to the monotone setting. Observe that $\mathrm{Circuit}_s^{\mathsf{mon}}(f)$ is a restriction of $\mathrm{Circuit}_s(f)$. Let τ be such that $\mathrm{Circuit}_s(f)\big|_\tau = \mathrm{Circuit}_s^{\mathsf{mon}}(f)$. This allows us to naturally extend k-determined restrictions to the monotone setting: a restriction ρ is a k-determined restriction for $\mathrm{Circuit}_s^{\mathsf{mon}}(f)$ 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 3.4 even though we are working with the monotone formula.

Lemma 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 $\mathrm{Circuit}_s^{\mathsf{mon}}(f)$ such that*

$$g_{s,\alpha}^{\mathrm{out}}(Y) = \begin{cases} 1, & \text{if } |\alpha| > \ell, \\ 0, & \text{if } |\alpha| < \ell, \\ f(\alpha), & \text{if } |\alpha| = \ell \text{ and } C_f^{\mathsf{mon}}(\alpha) \neq \bot, \\ \bigoplus_{i \in N(\alpha)} y_i, & \text{otherwise,} \end{cases}$$

for $g_{s,\alpha}^{\mathrm{out}}$ and Y as in Definition 3.2.

Furthermore, the formula $\mathrm{Circuit}_s^{\mathsf{mon}}(f)\big|_\rho$ is over $O(t \cdot k + m)$ variables.

Proof. This proof is an adaptation of the argument of the proof Lemma 4.1. Let us describe the natural m-independent k-determined restriction ρ for the formula $\mathrm{Circuit}_s^{\mathsf{mon}}(f)$.

As in the proof of Lemma 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 5.1 we may assume that the circuit C_f^{mon} computes a monotone ℓ-slice function in both outputs $C_{f,1}^{\mathsf{mon}}, C_{f,2}^{\mathsf{mon}}$ for a mild increase in size; $|C_f^{\mathsf{mon}}| \leq s/5 + \mathsf{poly}(n) \leq s/4$ for s large enough. Recall that the first output of C_f^{mon} indicates whether the second output bit is equal to f on the ℓ-slice. Let $\overline{C}_{f,1}^{\mathsf{mon}}$ be the negation of $C_{f,1}^{\mathsf{mon}}$ on the ℓ-slice. In other words, $\overline{C}_{f,1}^{\mathsf{mon}}(\alpha) = \neg C_{f,1}^{\mathsf{mon}}(\alpha)$ if α has Hamming weight ℓ, and $\overline{C}_{f,1}^{\mathsf{mon}}(\alpha) = C_{f,1}^{\mathsf{mon}}(\alpha)$ otherwise.

The monotone circuit C_f^{mon} is of size at most $s/4$ and hence according to Lemma 2.4 there is a monotone circuit of size $s/2 + \mathsf{poly}(n) \leq 5s/8$ computing $\overline{C}_{f,1}^{\mathsf{mon}}(\alpha)$.

We restrict the formula such that a part of the circuit is equivalent to C_f^{mon} and another part is equal to $\overline{C}_{f,1}^{\mathsf{mon}}$. 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 $\mathsf{Sel}_1, \mathsf{Sel}_2, \ldots, \mathsf{Sel}_m$, each of size $\mathsf{poly}(n,k)$, 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 $\overline{\mathsf{Sel}_i} = \neg\, \mathsf{Sel}_i$ and denote by $\mathsf{Sel}_i^{\mathsf{mon}}$ (respectively $\overline{\mathsf{Sel}_i}^{\mathsf{mon}}$) the circuit obtained by applying Claim 5.1 to Sel_i (to $\overline{\mathsf{Sel}_i}$ respectively). By the guarantees of Claim 5.1 all these $2m$ circuits are of size $\mathsf{poly}(n,k)$.

We restrict the formula such that a part of the circuit computes the functions

$$\mathsf{Sel}_1^{\mathsf{mon}}, \ldots, \mathsf{Sel}_m^{\mathsf{mon}}, \overline{\mathsf{Sel}_1}^{\mathsf{mon}}, \ldots, \overline{\mathsf{Sel}_m}^{\mathsf{mon}} \quad . \tag{21}$$

From these ℓ-slice selector circuits we can then define selector circuits that take C_f^{mon} into account. Namely, we connect $\mathsf{Sel}_i^{\mathsf{mon}}$ by an *and* gate to the output of $\overline{C}_{f,1}^{\mathsf{mon}}$ to obtain the circuit $\mathsf{Sel}_i'^{\mathsf{mon}}$ and similarly connect $\overline{\mathsf{Sel}_i}^{\mathsf{mon}}$ by an *or* gate to $C_{f,1}^{\mathsf{mon}}$ to obtain the circuit $\overline{\mathsf{Sel}_i}'^{\mathsf{mon}}$.

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 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 $\bar{y}_i^{\mathsf{mon}}$). It is well-known [Val84, BW06, Gol20] that threshold circuits have montone circuits of size $\mathsf{poly}(n)$ and we can thus restrict the formula such that a part of the circuit computes y_i^{mon} and $\bar{y}_i^{\mathsf{mon}}$.

Finally we connect y_i^{mon} by an *and* gate to the selector circuit $\mathsf{Sel}_i'^{\mathsf{mon}}$. 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 $\bar{y}_i^{\mathsf{mon}}$ similarly: connect $\bar{y}_i^{\mathsf{mon}}$ by an *or* gate to the negated selector circuit $\overline{\mathsf{Sel}_i}'^{\mathsf{mon}}$. 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 $\mathsf{Sel}_i'^{\mathsf{mon}}(\alpha) \wedge y_i$ while the negative circuit outputs $\overline{\mathsf{Sel}_i}'^{\mathsf{mon}}(\alpha) \vee \bar{y}_i$. 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 $\mathsf{Sel}_i'^{\mathsf{mon}}(\alpha) \wedge y_i$ 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 $\mathsf{Sel}_i'^{\mathsf{mon}} \wedge y_i$. Whenever the negation of $\mathsf{Sel}_i'^{\mathsf{mon}}(\alpha) \wedge y_i$ is needed, we do one last step of De Morgan and replace it by $\overline{\mathsf{Sel}_i}'^{\mathsf{mon}}(\alpha) \vee \bar{y}_i$.

To ensure that the circuit outputs $f(\alpha)$ whenever $C_f^{\mathsf{mon}}(\alpha) \neq \perp$, 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 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.

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 4.1 and we thus omit it here. □

Let us prove our degree lower bound for monotone circuits, restated here for convenience.

Theorem 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* $\mathsf{Circuit}_s^{\mathsf{mon}}(f)$.

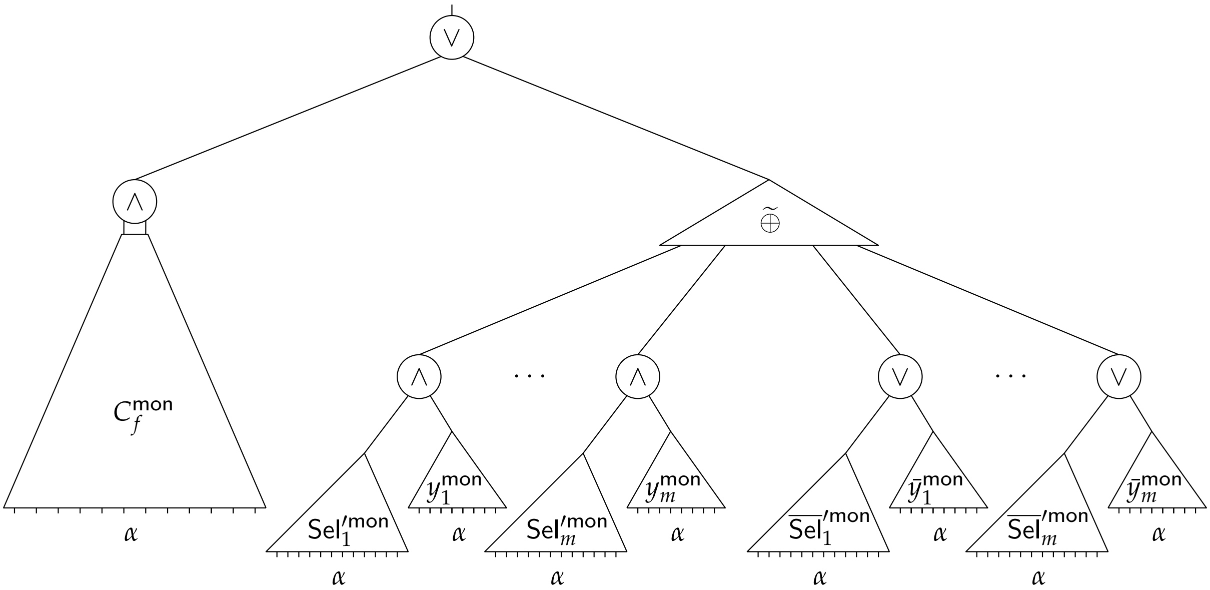
∨
∧
C_f^{mon}
α
$\widetilde{\oplus}$
∧
∧
∨
∨
. . .
. . .
$\mathsf{Sel}_1'^{\mathsf{mon}}$
y_1^{mon}
$\mathsf{Sel}_m'^{\mathsf{mon}}$
y_m^{mon}
$\overline{\mathsf{Sel}}_1'^{\mathsf{mon}}$
$\bar{y}_1^{\mathsf{mon}}$
$\overline{\mathsf{Sel}}_m'^{\mathsf{mon}}$
$\bar{y}_m^{\mathsf{mon}}$
α

Figure 2: A depiction of the monotone circuit, where $\widetilde{\oplus}$ is the $\oplus$ circuit with the negations pushed down.

Proof of Theorem 1.4. As in the proof of Theorem 1.1, we use the graphs from Theorem 2.2, with $U = \{0,1\}^n$, $k = O_\gamma\big((n \log r)^{1+1/\gamma}\big)$, and $|V| \le k^2 r^{1+\gamma}$ for parameters $\gamma > 0$ and $r \le 2^n$. We apply Lemma 5.2 with above graph and $t = 2^n$ along with $C_f^{\mathsf{mon}} = \bot$ to obtain, for $s \ge m \cdot \mathsf{poly}(n,k)$, an appropriate natural m-independent k-determined affine restriction ρ for $\mathrm{Circuit}_s^{\mathsf{mon}}(f)$. In particular ρ satisfies

$$g_{s,\alpha}^{\mathrm{out}}(Y) = \begin{cases} 1, & \text{if } |\alpha| > \ell, \\ 0, & \text{if } |\alpha| < \ell, \\ \bigoplus_{i \in N(\alpha)} y_i, & \text{otherwise,} \end{cases}$$

for $g_{s,\alpha}^{\mathrm{out}}$ and Y as in definition Definition 3.2.

Recall that there is a restriction τ such that $\mathrm{Circuit}_s^{\mathsf{mon}}(f) = \mathrm{Circuit}_s(f)\big|_\tau$ and we can thus apply Lemma 3.4 with $\tau\rho$ to conclude that if there is an SoS refutation of $\mathrm{Circuit}_s^{\mathsf{mon}}(f)\big|_\rho$ in degree d, then there is a degree $d \cdot k$ SoS refutation of the system of polynomial equations computing

$$\Big\{ \bigoplus_{i \in N(\alpha)} y_i = f(\alpha) \mid \alpha \in \binom{[n]}{\ell} \Big\} \ . \tag{22}$$

As the graph G is a strong expander, we can apply Theorem 2.6 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 $\mathrm{Circuit}_s^{\mathsf{mon}}(f)\big|_\rho$ formula and hence also for the unrestricted formula.

Regarding the parameters, as in the proof of Theorem 1.1 we choose $m = s/\mathsf{poly}_\gamma(n)$. Repeating the calculations from the aforementioned proof we obtain that $r \ge s^{1/(1+\gamma)}/\mathsf{poly}_\gamma(n)$. Thus by choosing γ small enough such that $\frac{1}{1+\gamma} > 1 - \varepsilon/2$ and s large enough such that the final $\mathsf{poly}_\gamma(n) \le s^{\varepsilon/2}$ 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 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* $\mathsf{Circuit}_s^{\mathsf{mon}}(f)$.

Proof. Analogous to the proof of Theorem 1.3. □

6 Degree Upper Bound

In this section we give a simple upper bound on the SoS refutation degree for $\mathsf{Circuit}_s(f)$. Specifically, for functions f that have no circuit of size s, we show that there is an SoS refutation of $\mathsf{Circuit}_s(f)$ of degree $O(s)$, essentially matching our $\Omega(s^{1-\epsilon})$ lower bound.

At a high level, the logic behind the refutation is as follows: first, we show that SoS in degree $O(s)$ can derive that the $\Theta(s^2)$ structure variables of $\mathsf{Circuit}_s(f)$ must uniquely describe a circuit of size s. Then, we also show that for any fixed circuit, SoS can in degree $O(s)$ derive that the circuit does not compute f by "evaluating" the circuit on some (non-deterministically chosen) input where the output differs from f.

To make this precise, we first define a set of monomials, which 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. exactly one of the variables $\mathsf{IsNeg}(v), \mathsf{IsOr}(v)$ or $\mathsf{IsAnd}(v)$ occurs in m,
2. for $a \in \{1,2\}$ exactly one of the variables $\mathsf{IsFromConst}(v,a)$, $\mathsf{IsFromInput}(v,a)$ or $\mathsf{IsFromGate}(v,a)$ occurs in m,
3. for $a \in \{1,2\}$ exactly one of the variables $\mathsf{ConstantValue}(v,a)$ or $\overline{\mathsf{ConstantValue}(v,a)}$ occurs in m,
4. for $a \in \{1,2\}$ exactly one of the variables $\{\mathsf{IsInput}(v,a,i) \mid i \in [n]\}$ occurs in m, and
5. for $a \in \{1,2\}$ and $v > 1$, exactly one of the variables $\{\mathsf{IsGate}(v,a,u) \mid u < v\}$ occurs in m, and
6. 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 $\sum_{m \in \mathcal{M}_s} m - 1$; this corresponds to SoS proving that the structure variables uniquely describe a circuit and does not use anything about f. Then in a second step we show that for every $m \in \mathcal{M}_s$, SoS can derive $-m$ in degree $O(s)$; this corresponds to SoS proving that the circuit described by m does not compute f correctly. Summing these two parts up yields an SoS derivation of -1, i.e., a refutation of the $\mathsf{Circuit}_s(f)$ 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

$$m \cdot \left(\mathsf{IsNeg}(v) + \mathsf{IsOr}(v) + \mathsf{IsAnd}(v) - 1\right) \tag{23}$$

to the derivation (note that the second term is Axiom 4). This gives us an SoS derivation of $\sum_{m \in \mathcal{M}'_s} m - 1$, where

$$\mathcal{M}'_s = \bigcup_{m \in \mathcal{M}_s} \{m \cdot \mathsf{IsNeg}(v), m \cdot \mathsf{IsOr}(v), m \cdot \mathsf{IsAnd}(v)\} \ .$$

We can continue in the same manner with Axiom 3 and Axiom 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 $-m$ for $m \in \mathcal{M}_s$. Let C be the circuit that corresponds to the monomial m and let $\alpha \in \{0,1\}^n$ be such that $f(\alpha) \neq C(\alpha)$ (by the assumption that f does not have a circuit of size s, such an α exists). Suppose $C(\alpha) = b$ but $f(\alpha) = 1 - b$.

We construct a degree $O(s)$ SoS proof of the fact that $C(\alpha) = b$. That is, we are going to show that the polynomial $p_s = m \cdot (\mathsf{Out}_\alpha(s) - b)$ can be written as

$$p_s = \sum_{i=1}^{t} r_i \cdot q_i \ , \tag{24}$$

for some parameter t, axioms $q_1, \ldots, q_t$ and some polynomials $r_1, \ldots, r_t$, such that $\deg(r_i \cdot q_i) = O(s)$ for all i. Note that given this, we can then easily derive $-m$ by subtracting the polynomial $m \cdot (\mathsf{Out}_\alpha(s) - f(\alpha)) = m \cdot (\mathsf{Out}_\alpha(s) - 1 + b)$ (this is m multiplied by Axiom 15), yielding a derivation of $m \cdot (1 - 2b)$ which is either m or $-m$ depending on b; in the former case it can be multiplied by -1 to yield $-m$. Note that here it is important that the derivation (24) is a Nullstellensatz derivation, not using any Sum-of-Squares part, since otherwise it would not be possible to multiply it by -1.

Let us thus see how to derive $m \cdot (\mathsf{Out}_\alpha(s) - b)$. We do this by structural induction over the circuit: for every gate v we are going to construct an SoS proof of the fact that the circuit rooted at v outputs the bit b_v on input α. In other words, an SoS derivation of the polynomial $m \cdot (\mathsf{Out}_\alpha(v) - b_v)$.

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 differently. As a first step let us construct SoS proofs q_1 and q_2 of the fact that on input α the bit c_a, $a \in \{1,2\}$, is carried on wire a to the gate v. That is, the polynomial q_a should simplify to $m \cdot \big(\mathsf{In}_\alpha(v,a) - c_a\big)$. 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 m is of the form $m = m' \cdot \mathsf{IsFromConst}(v,a) \cdot \mathsf{ConstantValue}(v,a)$ for some monomial m', i.e., wire a is connected to the constant $c_a = 1$ in the circuit described by m, then we can derive $q_a = m \cdot \big(\mathsf{In}_\alpha(v,a) - 1\big)$ by the identity

$$\begin{aligned} q_a = m' \cdot \mathsf{ConstantValue}(v,a) \cdot \mathsf{IsFromConst}(v,a) \cdot \big(\mathsf{In}_\alpha(v,a) - \mathsf{ConstantValue}(v,a)\big) + \\ m' \cdot \mathsf{IsFromConst}(v,a) \cdot \big(\mathsf{ConstantValue}(v,a)^2 - \mathsf{ConstantValue}(v,a)\big) \ , \end{aligned} \tag{25}$$

a linear combination of Axiom 9 and the Boolean axiom on $\mathsf{ConstantValue}(v,a)$. Similarly if $m = m' \cdot \mathsf{IsFromConst}(v,a) \cdot \overline{\mathsf{ConstantValue}(v,a)}$ (i.e., $c_a = 0$) we can derive $q_a = m \cdot \mathsf{In}_\alpha(v,a)$ by

$$\begin{aligned} q_a = m' \cdot \overline{\mathsf{ConstantValue}(v,a)} \cdot \mathsf{IsFromConst}(v,a) \cdot \big(\mathsf{In}_\alpha(v,a) - \mathsf{ConstantValue}(v,a)\big) + \\ m' \cdot \mathsf{IsFromConst}(v,a) \cdot \mathsf{ConstantValue}(v,a) \cdot \\ \big(1 - \mathsf{ConstantValue}(v,a) - \overline{\mathsf{ConstantValue}(v,a)}\big) + \\ m' \cdot \mathsf{IsFromConst}(v,a) \cdot (\mathsf{ConstantValue}(v,a)^2 - \mathsf{ConstantValue}(v,a)) \ , \end{aligned} \tag{26}$$

where we additionally use the negation axiom on $\mathsf{ConstantValue}(v,a)$.

Next, if a is connected to an input i and m is of the form $m = m' \cdot \mathsf{IsFromInput}(v,a) \cdot \mathsf{IsInput}(v,a,i)$ (so that $c_a = \alpha_i$), then

$$q_a = m \cdot (\mathsf{In}_\alpha(v,a) - \alpha_i) = m' \cdot \mathsf{IsFromInput}(v,a) \cdot \mathsf{IsInput}(v,a,i) \cdot \big(\mathsf{In}_\alpha(v,a) - \alpha_i\big) \ , \tag{27}$$

which is a multiple of Axiom 10. Lastly, if a is connected to a gate u and m is of the form $m = m' \cdot \mathsf{IsFromGate}(v,a) \cdot \mathsf{IsGate}(v,a,u)$ (i.e., $c_a = b_u$), then

$$\begin{aligned} q_a = m \cdot (\mathsf{In}_\alpha(v,a) - b_u) = m' \cdot \mathsf{IsFromGate}(v,a) \cdot \mathsf{IsGate}(v,a,u) \cdot \big(\mathsf{In}_\alpha(v,a) - \mathsf{Out}_\alpha(u)\big) \\ + m \cdot (\mathsf{Out}_\alpha(u) - b_u) \ , \end{aligned} \tag{28}$$

where the first term is a multiple of Axiom 11, and the second term is the polynomial p_u which, by induction, we assume has already been derived in degree $O(s)$.

Given the two SoS proofs q_1 and q_2 we are ready to construct the SoS proof $p_v = m \cdot (\mathsf{Out}_\alpha(v) - b_v)$. As mentioned earlier we do a case distinction over the funtion computed at v.

1. *v is a* not *gate ($m = m' \cdot \mathsf{IsNeg}(v)$ and $b_v = 1 - c_1$).* We have the derivation

$$\begin{aligned} p_v = m' \cdot \mathsf{IsNeg}(v) \cdot \big(\mathsf{Out}_\alpha(v) - \overline{\mathsf{In}_\alpha(v,1)}\big) \\ + m \cdot \big(\overline{\mathsf{In}_\alpha(v,1)} - 1 + \mathsf{In}_\alpha(v,1)\big) \\ - q_1 \ , \end{aligned} \tag{29}$$

where the first line uses Axiom 12 and the second line uses the negation axiom for $\mathsf{In}_\alpha(v,1)$.

2. *v is an* or *gate ($m = m' \cdot \mathsf{IsOr}(v)$, and $b_v = 1 - (1 - c_1)(1 - c_2)$.* We have the derivation

$$\begin{aligned} p_v =& m' \cdot \mathsf{IsOr}(v) \cdot \Big(\mathsf{Out}_\alpha(v) - \big(1 - \overline{\mathsf{In}_\alpha(v,1)} \cdot \overline{\mathsf{In}_\alpha(v,2)}\big)\Big) \\ &- m \cdot \overline{\mathsf{In}_\alpha(v,1)} \cdot \big(\overline{\mathsf{In}_\alpha(v,2)} - 1 + \mathsf{In}_\alpha(v,2)\big) \\ &+ m \cdot \big(\mathsf{In}_\alpha(v,2) - 1\big) \cdot \big(\overline{\mathsf{In}_\alpha(v,1)} - 1 + \mathsf{In}_\alpha(v,1)\big) \\ &+ \big(1 - \mathsf{In}_\alpha(v,1)\big) \cdot q_2 \\ &+ \big(1 - c_2\big) \cdot q_1 \ , \end{aligned} \tag{30}$$

where the first line uses Axiom 13, and the following two lines uses negation axioms.

3. *v is an* and *gate ($m = m' \cdot \mathsf{IsAnd}(v)$, and $b_v = c_1 \cdot c_2$.* We have

$$\begin{aligned} p_v =& m' \cdot \mathsf{IsAnd}(v) \cdot \big(\mathsf{Out}_\alpha(v) - \mathsf{In}_\alpha(v,1) \cdot \mathsf{In}_\alpha(v,2)\big) \\ &+ \mathsf{In}_\alpha(v,1) \cdot q_2 \\ &+ c_2 \cdot q_1 \ , \end{aligned} \tag{31}$$

where the first line uses Axiom 14.

This completes the description of the SoS derivation of $\mathsf{Out}_\alpha(s) = b$. 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.

7 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

[AH19] Albert Atserias and Tuomas Hakoniemi. Size-Degree Trade-Offs for Sums-of-Squares and Positivstellensatz Proofs. In Amir Shpilka, editor, *34th Computational Complexity Conference (CCC 2019)*, volume 137 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 24:1–24:20, Dagstuhl, Germany, 2019. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik. 4, 6

[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)*, pages 689–708, Los Alamitos, CA, USA, oct 2015. IEEE Computer Society. 1

[Ber82] S. J. Berkowitz. On some relationships between monotone and non-monotone circuit complexity. Technical report, Technical Report, University of Toronto, 1982. 5

[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)*, pages 428–437, 2016. 2

[BT06] Andrej Bogdanov and Luca Trevisan. Average-case complexity. *Foundations and Trends in Theoretical Computer Science*, 2(1):1–106, 2006. 3

[BW06] Amos Beimel and Enav Weinreb. Monotone circuits for monotone weighted threshold functions. *Inf. Process. Lett.*, 97(1):12–18, jan 2006. 17

[GHP02] Dima Grigoriev, Edward A. Hirsch, and Dmitrii V. Pasechnik. Complexity of semi-algebraic proofs. In *STACS 2002*, pages 419–430, Berlin, Heidelberg, 2002. Springer Berlin Heidelberg. 2

[Gol20] Oded Goldreich. *On (Valiant's) Polynomial-Size Monotone Formula for Majority*, pages 17–23. Springer International Publishing, Cham, 2020. 17

[Gri01] Dima Grigoriev. Linear lower bound on degrees of positivstellensatz calculus proofs for the parity. *Theoretical Computer Science*, 259(1):613 – 622, 2001. 1, 6

[GUV09] Venkatesan Guruswami, Christopher Umans, and Salil Vadhan. Unbalanced expanders and randomness extractors from Parvaresh–Vardy codes. *Journal of the ACM*, 56(4):20:1–20:34, July 2009. Preliminary version in *CCC '07*. 4, 5

[GW95] Michel X. Goemans and David P. Williamson. Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming. *J. ACM*, 42(6):1115–1145, 1995. 1

[Hir18] Shuichi Hirahara. Non-black-box worst-case to average-case reductions within np. In *2018 IEEE 59th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 247–258, 2018. 1

[KC00] Valentine Kabanets and Jin-Yi Cai. Circuit minimization problem. In *Proceedings of the Thirty-Second Annual ACM Symposium on Theory of Computing*, STOC '00, page 73–79, New York, NY, USA, 2000. Association for Computing Machinery. 1

[KMOW17] Pravesh K. Kothari, Ryuhei Mori, Ryan O'Donnell, and David Witmer. Sum of squares lower bounds for refuting any csp. In *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing*, STOC 2017, page 132–145, New York, NY, USA, 2017. Association for Computing Machinery. 2

[KMS98] David Karger, Rajeev Motwani, and Madhu Sudan. Approximate graph coloring by semidefinite programming. *J. ACM*, 45(2):246–265, mar 1998. 1

[MPW15] Raghu Meka, Aaron Potechin, and Avi Wigderson. Sum-of-squares lower bounds for planted clique. In *Proceedings of the 47th Annual ACM Symposium on Theory of Computing (STOC '15)*, pages 87–96, June 2015. 2

[MW15] Cody D. Murrayand and R. Ryan Williams. On the (Non) NP-Hardness of Computing Circuit Complexity. In David Zuckerman, editor, *30th Conference on Computational Complexity (CCC 2015)*, volume 33 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 365–380, Dagstuhl, Germany, 2015. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik. 1

[PR04] Toniann Pitassi and Ran Raz. Regular resolution lower bounds for the weak pigeonhole principle. *Combinatorica*, 24(3):503–524, 2004. Preliminary version in *STOC '01*. 2

[Raz98] Alexander A. Razborov. Lower bounds for the polynomial calculus. *Computational Complexity*, 7(4):291–324, December 1998. 2, 7, 25

[Raz04a] Ran Raz. Resolution lower bounds for the weak pigeonhole principle. *J. ACM*, 51(2):115–138, 2004. 2

[Raz04b] Alexander A. Razborov. Resolution lower bounds for perfect matching principles. *Journal of Computer and System Sciences*, 69(1):3–27, August 2004. Preliminary version in *CCC '02*. 2, 7

[Raz15] Alexander A. Razborov. Pseudorandom generators hard for k-DNF resolution and polynomial calculus resolution. *Annals of Mathematics*, 181(2):415–472, March 2015. 2, 4

[Raz21] Alexander Razborov. P, NP and Proof Complexity. `https://youtu.be/ZVL_HsPC4xE?t=2646`, 2021. Accessed April 2022. 2

[Raz22] Alexander Razborov. Open problems. `https://people.cs.uchicago.edu/~razborov/teaching/index.html`, 2022. Accessed April 2022. 2

[RRS17] Prasad Raghavendra, Satish Rao, and Tselil Schramm. Strongly refuting random csps below the spectral threshold. In *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing*, STOC 2017, page 121–131, New York, NY, USA, 2017. Association for Computing Machinery. 1

[RWY02] Alexander A. Razborov, Avi Wigderson, and Andrew Chi-Chih Yao. Read-once branching programs, rectangular proofs of the pigeonhole principle and the transversal calculus. *Combinatorica*, 22(4):555–574, 2002. Preliminary version in *STOC '97*. 2

[Val84] Leslie G. Valiant. Short monotone formulae for the majority function. *J. Algorithms*, 5:363–366, 1984. 17

A On Encodings of the $\mathsf{Circuit}_s(f)$ Tautology

Let us introduce a possible constant width CNF encoding of $\mathsf{Circuit}_s(f)$ as proposed by Razborov [Raz98].

The formula is defined over the same variables as introduced in Section 2.3, but in order to keep the fan-in bounded, we further introduce the extension variables $\mathsf{IsInput}^{\leq}(v,a,i)$ and $\mathsf{IsGate}^{\leq}(v,a,u)$ 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 2.3. First we have the structure axioms. The first axioms encode that each wire is connected to a single kind

$$\begin{aligned}
&\big(\mathsf{IsFromConst}(v,a) \vee \mathsf{IsFromInput}(v,a) \vee \mathsf{IsFromGate}(v,a)\big)\wedge\\
&\neg\big(\mathsf{IsFromConst}(v,a) \wedge \mathsf{IsFromInput}(v,a)\big)\wedge\\
&\neg\big(\mathsf{IsFromInput}(v,a) \wedge \mathsf{IsFromGate}(v,a)\big)\wedge\\
&\neg\big(\mathsf{IsFromConst}(v,a) \wedge \mathsf{IsFromGate}(v,a)\big) \ .
\end{aligned} \tag{32}$$

The next set of axioms similarly ensures that each gate computes precisely one function

$$\begin{aligned}
&\big(\mathsf{IsNeg}(v) \vee \mathsf{IsOr}(v) \vee \mathsf{IsAnd}(v)\big)\wedge\\
&\neg\big(\mathsf{IsNeg}(v) \wedge \mathsf{IsOr}(v)\big) \wedge \neg\big(\mathsf{IsOr}(v) \wedge \mathsf{IsAnd}(v)\big) \wedge \neg\big(\mathsf{IsNeg}(v) \wedge \mathsf{IsAnd}(v)\big) \ .
\end{aligned} \tag{33}$$

Last, we need to make sure that each wire is connected to a single input or a gate.

$$\begin{aligned}
&\mathsf{IsInput}^{\leq}(v,a,n) \wedge \bigwedge_{i\neq j} \neg\big(\mathsf{IsInput}(v,a,i) \wedge \mathsf{IsInput}(v,a,j)\big)\wedge\\
&\bigwedge_{i\in[n]} \Big(\mathsf{IsInput}^{\leq}(v,a,i) \equiv \big(\mathsf{IsInput}^{\leq}(v,a,i-1) \vee \mathsf{IsInput}(v,a,i)\big)\Big) \ ,\\
&\qquad \text{where } \mathsf{IsInput}^{\leq}(v,a,0) \overset{\text{def}}{\equiv} 0 \ ,
\end{aligned} \tag{34}$$

and similarly for $v \in [s] \setminus \{1\}$ we have that

$$\begin{aligned}
&\mathsf{IsGate}^{\leq}(v,a,v-1) \wedge \bigwedge_{u<u'<v} \neg\big(\mathsf{IsGate}(v,a,u) \wedge \mathsf{IsGate}(v,a,u')\big)\wedge\\
&\bigwedge_{u\in[v-1]} \Big(\mathsf{IsGate}^{\leq}(v,a,u) \equiv \big(\mathsf{IsGate}^{\leq}(v,a,u-1) \vee \mathsf{IsGate}(v,a,u)\big)\Big) \ ,\\
&\qquad \text{where } \mathsf{IsGate}^{\leq}(v,a,0) \overset{\text{def}}{\equiv} 0 \ .
\end{aligned} \tag{35}$$

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

$$\mathsf{IsFromConst}(v,a) \to \big(\mathsf{In}_\alpha(v,a) \equiv \mathsf{ConstantValue}(v,a)\big) \ , \tag{36}$$

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

$$\mathsf{IsFromInput}(v,a) \wedge \mathsf{IsInput}(v,a,i) \to \mathsf{In}_\alpha(v,a) \equiv \alpha_i \ , \tag{37}$$

$$\mathsf{IsFromGate}(v,a) \wedge \mathsf{IsGate}(v,a,u) \to \mathsf{In}_\alpha(v,a) \equiv \mathsf{Out}_\alpha(u) \ . \tag{38}$$

Last we need to make sure that the gates propagate the value they are supposed to compute.

$$\mathsf{IsNeg}(v) \to \big(\mathsf{Out}_\alpha(v) \equiv \neg\, \mathsf{In}_\alpha(v,1)\big) \tag{39}$$

$$\mathsf{IsOr}(v) \to \big(\mathsf{Out}_\alpha(v) \equiv \mathsf{In}_\alpha(v,1) \vee \mathsf{In}_\alpha(v,2)\big) \tag{40}$$

$$\mathsf{IsAnd}(v) \to \big(\mathsf{Out}_\alpha(v) \equiv \mathsf{In}_\alpha(v,1) \wedge \mathsf{In}_\alpha(v,2)\big) \ . \tag{41}$$

The final axioms ensure that the correct function is computed

$$\mathsf{Out}_\alpha(s) \equiv f(\alpha) \ . \tag{42}$$

This formula can be rewritten in the usual manner into a 4-CNF. Let us denote this formula by $\mathrm{Circuit}_s^{\mathrm{CNF}}(f)$.

Proposition A.1. *If there is an SoS refutation of degree d of the CNF formula $\mathrm{Circuit}_s^{CNF}(f)$, then there is an SoS refutation of degree $O(d)$ of the system of polynomials $\mathrm{Circuit}_s(f)$ as introduced in Section 2.3.*

The rest of this section is devoted to the proof of Proposition A.1.

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 $\mathsf{IsInput}^{\leq}(v,a,i)$ and $\mathsf{IsGate}^{\leq}(v,a,u)$) 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.

Let us define a polynomial substitution ρ that gets rid of the extension variables in the natural manner: the substitution ρ first replace each occurrence of a "bar" extension variable $\overline{\mathsf{IsInput}^{\leq}(v,a,i)}$ or $\overline{\mathsf{IsGate}^{\leq}(v,a,u)}$ by the polynomial $1 - \mathsf{IsInput}^{\leq}(v,a,i)$ and the polynomial $1 - \mathsf{IsGate}^{\leq}(v,a,u)$ respectively. Then, ρ replaces each occurrence of the variable $\mathsf{IsInput}^{\leq}(v,a,i)$ by $\sum_{j\leq i} \mathsf{IsInput}(v,a,j)$ and similarly $\mathsf{IsGate}^{\leq}(v,a,u)$ by $\sum_{w\leq u} \mathsf{IsGate}(v,a,w)$.

Suppose we have a degree d refutation π of $\mathrm{Circuit}_s^{\mathrm{CNF}}(f)$. Let us consider $\pi\big|_\rho$ and $\mathrm{Circuit}^{\mathrm{CNF}}(f)\Big|_\rho$. Note that $\pi\big|_\rho$ is a degree d SoS refutation of $\mathrm{Circuit}_s^{\mathrm{CNF}}(f)\Big|_\rho$.

We claim that in constant degree the axioms of $\mathrm{Circuit}_s^{\mathrm{CNF}}(f)\Big|_\rho$ can be derived from the polynomial encoding $\mathrm{Circuit}_s(f)$. As previously noted, this holds for all axioms but the ones that are over a non-constant number of variables. In other words it just remains to show that we can derive the substituted Axioms 34 and 35 from Axioms 5 to 8.

Let us consider Axiom 34. With the extension variables substituted and translated into a system of polynomials the axiom consists of the following polynomial equations.

$$1 - \sum_{j \in [n]} \mathsf{IsInput}(v, a, j) = 0 \tag{43}$$

$$\mathsf{IsInput}(v, a, i) \cdot \mathsf{IsInput}(v, a, j) = 0, \text{ for } i \neq j \tag{44}$$

$$\begin{aligned} &\left(\sum_{j \leq i} \mathsf{IsInput}(v, a, j) \right) \left(1 - \sum_{j < i} \mathsf{IsInput}(v, a, j) \right) \cdot \\ &\qquad \overline{\mathsf{IsInput}(v, a, i)} = 0, \text{ for } i \in [n] \end{aligned} \tag{45}$$

$$\left(1 - \sum_{j \leq i} \mathsf{IsInput}(v, a, j) \right) \left(\sum_{j < i} \mathsf{IsInput}(v, a, j) \right) = 0, \text{ for } i \in [n] \tag{46}$$

$$\left(1 - \sum_{j \leq i} \mathsf{IsInput}(v, a, j) \right) \mathsf{IsInput}(v, a, i) = 0, \text{ for } i \in [n] \ . \tag{47}$$

Axiom 43 is equal to Axiom 5 and similarly Axiom 44 is equal to Axiom 7. In the following we show that Axioms 45 to 47 can be derived from Axiom 7, the Boolean axioms and the negation axioms in constant degree.

Consider Axiom 45. Expand and rewrite modulo the Boolean axioms and the negation axiom to obtain

$$\begin{aligned} &\mathsf{IsInput}(v, a, i) \left(\sum_{j < i} \mathsf{IsInput}(v, a, j) \right)^2 - \\ &\quad 2 \sum_{j < j' < i} \mathsf{IsInput}(v, a, j) \cdot \mathsf{IsInput}(v, a, j') - \\ &\qquad \mathsf{IsInput}(v, a, i) \sum_{j < i} \mathsf{IsInput}(v, a, j) = 0 \ . \end{aligned} \tag{48}$$

Observe that every term t left in this polynomial is of the form $t = t' \cdot \mathsf{IsInput}(v, a, j) \cdot \mathsf{IsInput}(v, a, j')$, 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 7 and we thus see that Axiom 45 can be derived in constant degree from $\text{Circuit}_s(f)$.

Let us consider Axiom 46. Rewrite modulo the Boolean axiom to obtain

$$\begin{aligned} &\mathsf{IsInput}(v, a, i) \sum_{j < i} \mathsf{IsInput}(v, a, j) + \\ &\quad 2 \sum_{j < j' < i} \mathsf{IsInput}(v, a, j) \cdot \mathsf{IsInput}(v, a, j') = 0 \ . \end{aligned} \tag{49}$$

All terms are of the form of Axiom 7 and we can thus derive Axiom 46 from $\text{Circuit}_s(f)$ in constant degree.

Last, we need to consider Axiom 47. Note that modulo the Boolean axiom we obtain the polynomial equation

$$- \mathsf{IsInput}(v, a, i) \sum_{j < i} \mathsf{IsInput}(v, a, j) = 0 \ . \tag{50}$$

Also in this polynomial every term is of the form of Axiom 7 and thus also Axiom 47 can be derived in constant degree.

What remains is to show that Axiom 35 can be derived from $\text{Circuit}_s(f)$ in constant degree. This can be checked analogous to Axiom 34 and we thus omit it here.

We conclude that all axioms of $\mathrm{Circuit}_s^{\mathrm{CNF}}(f)\Big|_\rho$ 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.