6.S977 Project: Can SoS Prove Circuit Lower Bounds?

Brian Liu
brliu@mit.edu

Nathan Sheffield
shefna@mit.edu

William Wang
wmwang@mit.edu

December 14, 2024

1 Introduction

A central problem in complexity theory is to show that SAT does not have polynomial-size circuits — that is, to show that NP $\not\subseteq$ P/poly. This problem saw a lot of enthusiasm in the 80s, with work showing exponential lower bounds against restricted classes of circuits (such as AC^0 — i.e. constant-depth arbitrary fan-in circuits of ANDs, ORs and NOTs, and $\mathsf{AC}^0[\oplus]$ — i.e. constant-depth arbitrary fan-in circuits of ANDs, ORs, NOTs and XORs). However, by the early 90s the well seemed to have dried up: although there has been some progress since, finding explicit lower bounds against many slightly larger classes of circuits appears hopelessly out of reach.

To a complexity theorist, the obvious question upon being stuck trying to solve something for 30 years is "is there some reason this is a hard problem?". And indeed, there have been a number of attempts at formalizing the intuition that circuit lower bounds are inherently difficult to prove: for instance, the Razborov-Rudich "natural proofs" barrier conditionally ruling out lower bounds that are constructive for a large fraction of truth tables, or results in "hardness magnification" showing that weak circuit lower bounds in fact imply stronger circuit lower bounds. One especially concrete approach to formalizing the difficulty of circuit lower bounds, which we will consider in this exposition, is to rule out some restricted form of proof. Specifically, one might be interested in ruling out a *non-uniform* notion of a proof of e.g. NP $\not\subseteq$ P/poly: given a particular propositional proof system, one would like to show that proof system requires proofs of length super-polynomial in n to show that SAT on n-bit inputs requires circuits of super-polynomial size.

As this paper was written as the final project for a class on sum of squares, we will focus on the sum-of-squares (a.k.a. SoS or Positivstellensatz) proof system. In Section 2, we explain a recent paper by Austrin and Risse [AR23] which formulates the minimum circuit size problem as a system of low-degree polynomial equalities, and shows that SoS requires high-degree proofs to refute the existence of a small circuit for *any* function. In Section 3, we then discuss whether this result can be extended to rule out sum-of-squares lower bounds against smaller classes of circuits. In section Section 4, we consider ***feasible interpolation***, a property of propositional proof systems which is known to imply certain lower bounds, including (under some assumptions) one against proving circuit lower bounds. We survey the paper of Hakoniemi that identifies a form of feasible interpolation posessed by sum of squares [Hak20], which we note may possibly provide an alternative route to (conditionally) ruling out SoS proofs of lower bounds. Finally, in Section 5 we step beyond SoS, and ask whether *any* propositional proof system can prove lower bounds against most functions. Here, we outline a meta-complexity result of Pich and Santhanam [PS19]: there exists some propositional proof system (with advice) such that no *other* propositional proof system can disprove that *that* system gives short circuit lower-bound proofs for most functions.

2 MCSP is hard for SoS

In this section, we provide a reformulation of work by Austrin and Risse [AR23] which shows that SoS needs high degree in order to provide circuit size lower bounds. Specifically, we will show the following theorem:

Theorem 1. *For all $\epsilon > 0$, the exists d st: $\forall n \in \mathbb{N}$, $s \geq n^d$, and any $f : \{0,1\}^n \to \{0,1\}$, SoS requires* deg $= \Omega_\epsilon(s^{1-\epsilon})$ *to refute the assertion that f has circuits of size $\leq s$.*

As a sketch for how this proof will work, we have a classic result by Grigoriev [Gri01] that XOR-CSPs require high degrees to refute:

Theorem 2. *For all $n \in \mathbb{N}$, $k, r = k(n), r(n)$, let $G = ((U, V), E)$ be an $(r, k, 2)$-expander. Then $\forall b \in \{0,1\}^{|U|}$, SoS requires degree $\Omega(r)$ to refute a satisfying assignment to the system*

$$\bigoplus_{v \in N(u)} x_v = b_u, \quad \forall u \in U$$

We will first express MCSP via a polynomial system, and construct a reduction to an XOR-CSP on an expander such that SoS proofs are preserved through the reduction. In this way, lower bounds given by Theorem 2 can be easily translated to MCSP.

2.1 Constructing a Polynomial System

We wish to encode the statement that $f(x)$ has circuits of size $\leq s$ in terms of polynomials. So, we begin by formulating a polynomial system whose solutions correspond to circuits of size s which compute f. Our polynomial system consists of two types of variables: *structural variables*, which determine the circuit itself, and *evaluation variables*, which ensure that computation on the circuit is consistent. For simplicity, we will assume that all variables are boolean, so for any variable x, we have the equation $x^2 = x$ in our system.

To define the circuit, take a topological sorting of the gates, and label them from 1 to s. For each, gate $u \in [s]$, we have the three variables $\text{IsOr}(u), \text{IsAnd}(u), \text{IsNeg}(u)$ and equation

$$\text{IsOr}(u) + \text{IsAnd}(u) + \text{IsNeg}(u) = 1, \quad \forall u \in [s]$$

so every gate is given exactly 1 of 3 possible types. To finish specifying the circuit, we also need to specify the connections between gates, namely where the inputs for each gate come from. Each gate has ≤ 2 inputs, and so for each $u \in [s], i \in \{1, 2\}$, we have the following variables for the ith input of gate u: first, we specify where the input came from with $\text{FromConst}(u, i), \text{FromInput}(u, i), \text{FromGate}(u, i)$ which correspond to the input being from a constant, from the input to the circuit, or from the output of another gate $v < u$. Each input corresponds to exactly one of these cases, so we have

$$\text{FromConst}(u, i) + \text{FromInput}(u, i) + \text{FromGate}(u, i) = 1, \quad \forall u \in [s], \quad i \in \{1, 2\}$$

If the input is a constant, we specify its value with variable $\text{ConstVal}(u, i)$. If the input is from an input, we specify which index of the input with indicator variable $\text{InputIndex}(u, i, j)$ where $j \in [n]$. This gives constraint

$$\text{FromInput}(u, i) \cdot \left(1 - \sum_{j=1}^{n} \text{InputIndex}(u, i, j)\right) = 0, \quad \forall u \in [s], \quad i \in \{1, 2\}$$

Finally, if the input is a gate, we specify which gate it came from with indicator $\text{GateIndex}(u, i, v)$. Note that we can only get output from v if $v < u$ since the gates are topologically sorted, so we have

$$\text{FromGate}(u, i) \cdot \left(1 - \sum_{v=1}^{u-1} \text{GateIndex}(u, i, v)\right) = 0$$

These variables and constraints completely specify circuits of size s with AND, OR, and NOT gates. Now, for evaluation variables, on each input $\alpha \in \{0,1\}^n$, and each gate $u \in [s]$, we define variables $\text{Out}_\alpha(u), \text{In}_{\alpha,1}(u), \text{In}_{\alpha,2}(u)$ to be the output and inputs of gate u on input α. These values must be consistent with circuit structure, so we must check that they compute the right things. For the output to be consistent with input, we have

$$\text{IsNeg}(u) \cdot (\text{Out}_\alpha(u) + \text{In}_{\alpha,1}(u) - 1) = 0, \quad \text{IsAnd}(u) \cdot (\text{Out}_\alpha(u) - \text{In}_{\alpha,1}(u)\text{In}_{\alpha,2}(u)) = 0,$$

$$\text{IsOr}(u) \cdot (\text{Out}_\alpha(u) - (1 - (1 - \text{In}_{\alpha,1}(u))(1 - \text{In}_{\alpha,2}(u)))) = 0$$

For the inputs to be consistent, they must come from where the wire structural variables say they come from, so

$$\text{FromConst}(u,i)\cdot(\text{In}_{\alpha,i}(u)-\text{ConstVal}(u,i))=0,\quad \text{FromInput}(u,i)\cdot\text{InputIndex}(u,i,j)\cdot(\text{In}_{\alpha,i}(u)-\alpha_j)=0,$$

$$\text{FromGate}(u,i)\cdot\text{GateIndex}(u,i,v)\cdot(\text{In}_{\alpha,i}(u)-\text{Out}_{\alpha}(v))=0$$

Finally, we wish for our circuit to compute f, so we specify that the outputs of gate s must be the correct values:

$$\text{Out}_{\alpha}(s)=f(\alpha),\quad \forall\alpha\in\{0,1\}^n$$

We now have a system of polynomials specified on $O(s)+O\left(\binom{s}{2}\right)+O(s2^n)=O(s2^n+s^2)$ total variables which corresponds exactly to circuits of size s which compute f. We will denote this system as $\text{Circuit}_s(f)$.

2.2 Polynomial Substitutions

Now that we have our polynomial system, we want to relate it to the XOR-CSP so we can apply Theorem 2. We first develop some theory on polynomial substitutions.

Definition 2.1. *A polynomial substitution ρ of degree $\leq k$ is defined by a mapping $\rho:\{x_1,\ldots,x_n\}\to\mathbb{R}_{\geq k}[X]$ which sends variables to multivariate polynomials of bounded degree.*

Definition 2.2. *Given polynomial system $\mathcal{P}$ and substitution ρ on the same set of variables, the restriction $\mathcal{P}|_\rho$ is defined by substituting all variables x_i in $\mathcal{P}$ with $\rho(x_i)$.*

With these definitions, sum of squares proofs are preserved under restriction:

Fact 2.3. *If $\mathcal{P}$ has a degree d SoS refutation, and ρ is a polynomial substitution of degree $\leq k$, then $\mathcal{P}|_\rho$ has a degree dk SoS refutation.*

The above is clear since we can get a refutation for the restricted system by just substituting variables in the SoS refutation of $\mathcal{P}$ with ρ. So, if we have a low-degree polynomial substitution whose restriction on $\text{Circuit}_s(f)$ produces a XOR-CSP-like instance, then we can finish by Theorem 2.

2.3 Constructing the Restriction

We want to somehow encode an expander into our circuit to apply Grigoriev's result. In order to do this, the exapnder should have some succinct circuit representation so it can be efficiently put into our construction. Luckily, such expanders do exist, by a result of Guruswami, Umans and Vadhan [GUV09]:

Fact 2.4. *For all $\gamma>0$, $M\in\mathbb{N}$, $r\leq M$, $\epsilon>0$, there is an $N\leq d^2r^{1+\gamma}$ such that we have $(r,d,(1-\epsilon)d)$ expander $G=((U,V),E)$ with $|U|=M,|V|=N$ and $d=O\left(((\log M\log r)/\epsilon)^{1+1/\gamma}\right)$ such that the neighbor relation $U\times V\to\{0,1\}$ is computable by a circuit of size*

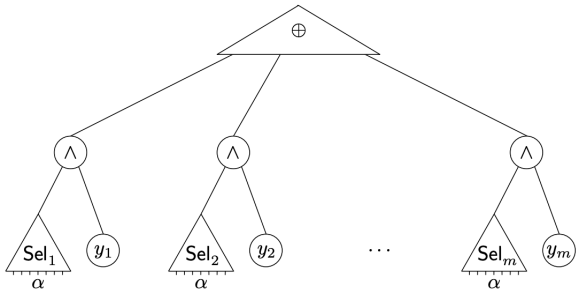
$$d(\text{poly}(\log M+\log d)+2\log N+1)$$

Fix $G=((U,V),E)$ to be an $(r,k,2)$ expander guaranteed by Fact 2.4 with $|U|=2^n,|V|=m$. We will now construct our substitution to restrict $\text{Circuit}_s(f)$ to an XOR-CSP on G. First, restrict the first m gates to be ORs. For each $i\in[m]$, set first input to constant 0 and keep the second unset as some variable $y_i\in\{0,1\}$. The output of the ith OR gate should therefore be y_i as well, and we will interchangeably refer to this gate as y_i for ease of notation. These unset constants will be the only unset structural variables in our construction, and everything else will follow a rigid, predetermined structure.

By Fact 2.4, for each i we can construct selector circuit Sel_i of size

$$k(\text{poly}(\log 2^n+\log k)+2\log m+1)=O(k\cdot\text{poly}(n+\log k)+\log m)=\text{poly}(n,k)$$

such that $\text{Sel}_i(\alpha)=1$ iff $(\alpha,i)\in E$. (Here, $\log m=\text{poly}(n,k)$ since $m\leq k^2(2^n)^{1+\gamma}$ by the bounds in Fact 2.4.) Hardcode these selector circuits into our restriction and AND Sel_i and y_i for each $i\in m$. Finally, XOR the resulting m values to produce our completed restricted circuit.



By construction, on input α, $\text{Sel}_i(\alpha) \wedge y_i$ is y_i if $i \in N(\alpha)$ and it is 0 otherwise, so the circuit computes the quantity $\bigoplus_{i \in N(\alpha)} y_i$, which is exactly the form of an XOR-CSP.

Let's now see how to formally specify the restriction from general circuits to this very specific type of circuit. The goal is to be able to replace all variables in $\text{Circuit}_s(f)$ to polynomials in terms of the $\{y_1, \ldots, y_m\}$ such that the resulting system is consistent with circuits of this form. First, note that all structural variables besides the unset constants are fixed, so they can be mapped to constants, meaning we only care about evaluation variables. The selector circuits are all fixed, so we map all the corresponding evaluation variables to constants. Similarly, all the evaluation variables corresponding to the y_i gates or the AND are either constants or y_i, so those are dealt with easily.

For all the remaining evaluation variables in the circuit, which are entirely within the XOR, note that for fixed α all of these variables only rely on k of the y_i. Any function over k boolean variables can be expressed as a degree-k multilinear polynomial via its Fourier Transform, so we can map all evaluation variables to their respective polynomials. This suffices for specifying our polynomial substitution.

To write out the resulting restricted system would involve taking each polynomial axiom we defined previously and use our substitution, which doesn't sound too promising, since XOR-CSPs don't have that many axioms. In fact, we will show that our restricted polynomial system is a lot simpler than one might expect, and exactly follows the form of the XOR-CSP:

Lemma 2.5. *Our restricted polynomial system can be expressed with the following axioms:*

$$\left\{ \bigoplus_{i \in N(\alpha)} y_i = f(x) \middle| \alpha \in \{0,1\}^n \right\} \cup \{y_i^2 = y_i\}$$

Proof. First, note that in $\text{Circuit}_s(f)$ we have that the y_i are boolean which gives the second set of axioms, and $\text{Out}_s(\alpha)$ is mapped to the degree-k multilinear XOR polynomial, which gives the first set of axioms. It suffices to show that all remaining axioms are encapsulated by the above. We first prove a fact about boolean functions:

Fact 2.6. *If polynomial $P(x_1, \ldots, x_n)$ satisfies $P(x_1, \ldots, x_n) = 0$ for all $(x_i) \in \{0,1\}^n$, then we can write*

$$P(x_1, \ldots, x_n) = \sum_{i=1}^{n} (x_i^2 - x_i) P_i(x_1, \ldots, x_n)$$

Proof. We will prove this with induction on n, where the base case of $n = 1$ is clear, since if $P(0) = P(1) = 0$, then $x(x-1) | P(x)$. For the inductive step, we do polynomial division wrt $x_n^2 - x_n$ to get

$$P(x_1, \ldots, x_n) = (x_n^2 - x_n) P_n(x_1, \ldots, x_{n-1}) + x_n Q(x_1, \ldots, x_{n-1}) + R(x_1, \ldots, x_{n-1})$$

Plugging in $x_n = 0$ shows that R is uniformly 0 over $(x_1, \ldots, x_{n-1}) \in \{0,1\}^{n-1}$, and then plugging in $x_n = 1$ shows $Q + R = Q$ is also uniformly 0 over $(x_1, \ldots, x_{n-1}) \in \{0,1\}^{n-1}$. So, by inductive hypothesis,

$$P(x_1, \ldots, x_n) = (x_n^2 - x_n) P_n(x_1, \ldots, x_{n-1}) + \sum_{i=1}^{n-1} (x_i^2 - x_i)(x_n Q_i(x_1, \ldots, x_{n-1}) + R_i(x_1, \ldots, x_{n-1}))$$

which suffices for the proof. □

This fact tells us that if a polynomial equality holds true over $(y_1, \dots, y_m) \in \{0,1\}^m$, then it can be derived via SoS from the boolean axioms on y_i, and therefore does not need to be included in our system. But, by construction, we have that any setting of the y_i over $\{0,1\}^m$ will produce a valid, consistent circuit, meaning that all of our structural axioms, intermediate evaluation axioms for consistency, and boolean axioms should hold true. Therefore, all of these axioms can be derived from the boolean axioms on the y_i, and the listed axioms are indeed all we need to fully specify the restricted polynomial system, as desired. □

Indeed, Lemma 2.5 shows that our restricted polynomial system exactly corresponds with the polynomial system of XOR-CSP, and therefore Theorem 2 shows that the restricted system requires SoS refutation of degree $\Omega(r)$. The restriction was of degree $\leq k$, so $\text{Circuit}_s(f)$ requires SoS refutation of degree $\Omega(r/k)$. What remains to be done to prove Theorem 1 is to maximize r/k while ensuring that our construction uses at most s gates.

In total, the construction uses $m \cdot \text{poly}(n,k) \leq O(mn^{C_1}k^{C_2-2}) \leq O(r^{1+\gamma}n^{C_1}k^{C_2})$ gates for some C_1, C_2. We have

$$k = O((n \log r)^{1+1/\gamma}) \leq O(n^{2+2/\gamma})$$

so we just need that

$$O\left(r^{1+\gamma}n^{C_3+2C_2/\gamma}\right) \leq s$$

Now, if we just define γ sufficiently small such that $\frac{1}{1+\gamma} < 1 - \frac{\epsilon}{2}$ for the ϵ defined in Theorem 1, we get that we can set $r = s^{1-\epsilon/2} \cdot n^{-C_1(\gamma)}$, $k \leq n^{C_2(\gamma)}$, so when $s \geq n^d$ for sufficiently large d, we require SoS refutation of degree

$$\Omega(r/k) = \Omega\left(s^{1-\epsilon/2}n^{-C_1(\gamma)-C_2(\gamma)}\right) \geq \Omega\left(s^{1-\epsilon}\right)$$

as desired.

3 Lower bounds against restricted circuit classes

Perhaps the natural question one is left with after that result is "is this really saying something about why new lower bounds are difficult, or just about why SoS is weak?". One way of asking this question is to consider whether SoS can prove circuit lower bounds if we restrict the class of circuits to one in which we *do* know unconditional lower bounds against explicit functions. If so, then the result of Section 2 would correspond to identifying a large class of methods which are known to be useful in some restricted settings, but provably not in general. If not, then perhaps it's instead telling us that SoS is just fundamentally not good at reasoning about circuits (at least in the encoding we've chosen). In this setting, we'll prove a couple of minor results in the direction of understanding this question, and briefly discuss potential ways to proceed.

3.1 Polynomial systems encoding weaker circuit classes

In order to talk about whether SoS can prove lower bounds against some circuit class $\mathcal{C}$, we'll need a different system of polynomial equalities than the one presented in Section 2, since we'll want the structural variables to be forced to describe a circuit belonging to $\mathcal{C}$. Let's note here a couple of modifications one could make.

i) **Bounded depth**: To represent a circuit with bounded depth, there are a couple options for how to proceed. If we're interested in circuits of only constant depth, as we will be in this section, the simplest thing to do would just be to have each gate assigned a layer of the circuit initially, and only include the variables $\text{FromGate}(u,i,v)$ when u appears in an later layer than v. However, in order to encode *any* circuit of size s and depth d, this would require us to have s gate variables at each layer, since it's possible that a circuit has most of its gates in a single layer. So this has multiplied the number of gate variables by d — note that if we care about this factor in the size, we should now also include a new Boolean variable for each gate specifying whether that gate is present in the circuit, with the constraint that the sum of all those variables is s. All of this has only increased the number of gate variables in our system by a constant if d is constant, however if d is larger we might want a different encoding. For instance, one could imagine giving each gate an additional d associated Boolean variables with sum 1, serving as the indicator of the layer it's on, and then enforcing the constraints that no gate's IN wire is connected to a gate at a greater than or equal layer.

ii) **Modified gate set**: One modification one might want to make is to introduce different gates, such as XOR/parity gates. For a gate that takes a constant number of inputs, this is easy to do: note that we can just add e.g. an IsXor variable for each gate, and then since the gate only acts over a constant number of inputs we can still express the function it computes as a constant-degree polynomial as we did for AND, OR and NEG, which will result in a constant-degree system of polynomials.

iii) **Arbitrary fan-in**: However, the above will not work if we want to allow our gates to have superconstant fan-in. In the case of superconstant fan-in, even AND or OR have high degree in their inputs, so the most natural extension of the system in Section 2 doesn't really make sense. A workaround we propose (there may be better ones known) is to represent an arbitrary-fan gate (either AND, OR, or $\oplus$) as a collection of $O(s)$ sub-gates with wires hardwired to form a binary tree and constrained to all compute the same operation. We introduce evaluation variables in the same way we did before for each of these sub-gates, so that the circuit still has to evaluate the same way. In other words, the workaround is just to increase the size and depth of the circuit, but then group and restrict the gates so that they're actually just implementing a small number and depth of arbitrary-fan in gates.

3.2 SoS can't prove exponential lower bounds against $\mathsf{AC}^0[\oplus]$

The approach of Section 2 seemed pretty general: we just took our initial system, substituted some structural variables with constants to restrict the circuit to a special form, and then noted that the evaluation variables could be substituted upwards to yield a polynomial corresponding to a hard CSP. In particular, note that the only property we need from our circuit class is that we can restrict it to something computing the particular functions we needed. Recalling that $\mathsf{AC}^0[\oplus]$ is the class of constant-depth circuits with arbitrary fan-in AND/OR/$\oplus$ gates, we can simply plug in known results to get the following proposition:

Proposition 3.1. *There exists a depth d such that, for any constant $\epsilon > 0$, any $f\colon \{0,1\}^n \to \{0,1\}$, and any $s > 2^{n^\epsilon}$, SoS requires degree $2^{\Omega(n^\epsilon)}$ to refute that f has depth-d, size-s $\mathsf{AC}^0[\oplus]$ circuits.*

Proof. By a result of Oliveira, Santhanam and Tell ([OST18], Theorem 4.5 in the ECCC preprint or Theorem 9 in the conference version), for sufficiently large constant d there exists for every ϵ a $\left(2^{\Omega(n^\epsilon)}, \mathrm{poly}(n), 2\right)$-expander $G = \left(\{0,1\}^n, \left[2^{n^\epsilon}\right], E\right)$ with neighbour relation computable by a size-$2^{O(n^\epsilon)}$ depth-d $\mathsf{AC}^0[\oplus]$ circuit[1]. So, if we build a circuit encoding an XOR-CSP instance on this graph with the same approach as we did in Section 2, we will need 2^{n^ϵ} selectors, each of which of size $2^{O(n^\epsilon)}$, giving a total circuit size $s = 2^{O(n^\epsilon)}$. Note that the depth required is only $d+2$, since each selector is just ORed with one of the y_i, and then fed into the final $\oplus$ gate. Once we've hardwired structural variables to restrict to this circuit, applying a $\mathrm{poly}(n)$-degree substitution on the evaluation variables produces the polynomial system corresponding to an XOR-CSP with condition values given by f on graph G, which requires SoS degree $2^{\Omega(n^\epsilon)}$ to refute. So, SoS requires degree at least $2^{\Omega(n^\epsilon)}/\mathrm{poly}(n) = 2^{\Omega(n^\epsilon)}$ to refute the existence of 2^{n^ϵ}-size $\mathsf{AC}^0[\oplus]$ circuits for any function f. □

This is perhaps rather disappointing, because we *do* have techniques to prove exponential $\mathsf{AC}^0[\oplus]$ lower bounds against very explicit functions: in fact, even for MOD-3 function, i.e. computing whether the sum of the bits in the input is a multiple of 3. So it seems what we've learned is that SoS fundamentally not capable of implementing these techniques. The next question is: can SoS implement *any* interesting circuit lower bound techniques? Or even relatively uninteresting circuit lower bound techniques? $\mathsf{AC}^0[\oplus]$ is pretty simple, but there's still room to look at simpler circuit classes.

3.3 What can we say about even weaker classes that $\mathsf{AC}^0[\oplus]$

Perhaps the weakest class which can reasonably be said to do anything at all[2] is NC^0: constant-depth circuits with *fan-in 2*. Note that we have a trivial circuit lower bound against NC^0: no matter the size, the output of any NC^0 circuit of depth d can only depend on at most 2^d bits of the input, so any function which depends on more

[1] They did not state the theorem with these parameters so if you go to the original paper you will need to set variables appropriately. It took me an embarrassingly long time to figure out how to match up the parameters here though (not sure how long but the units were hours) so I don't recommend doing it — unless you don't trust me, which might be a good call.

[2] And indeed probably many reasonable people would hold that this class is not even at the level of doing anything at all.

bits than that has no circuit of that depth. However, it's not immediately obvious[3] that SoS is able to make this argument given the polynomial system we're using to encode the circuit size problem. Austrin and Risse do show that, for general circuits, any f with no size-s circuit has an SoS proof of this fact with degree $O(s)$. But their proof relies on deriving a polynomial of large enough degree that the monomials correspond exactly to the valid size-s circuits, and here we would like to do something better. It would be quite a let down if it turned out SoS still needed $\text{poly}(s)$ size even to refute existence of size-s NC^0 circuits, because then the extra size ought not to help at all.

Fortunately, not all hope is lost! SoS does prove these NC^0 lower bounds in constant-degree.

Proposition 3.2. *For any d, if $f\colon \{0,1\}^n \to \{0,1\}$ doesn't have depth-d NC^0 circuits, then there exists a degree-$2^{O(d)}$ SoS refutation of the corresponding polynomial system for any s.*

Proof. Suppose there exists no such SoS refutation. Then, for any constant C, by duality there exists a degree-$(C \cdot 2^d)$ pseudodistribution over Boolean assignments to the variables that satisfies all constraints. We will generate the relevant portion of the circuit by choosing variables one at a time to condition on. Specifically, starting with the output gate we'll look at the local distribution for IsOr, IsAnd, and IsNeg. At least one of these must have nonzero probability, so we'll condition on that value. Similarly, we'll look at the all the possible inputs to the left input — there must be at least one with nonzero probability, so we'll condition on that. Once we've got concrete values pinned down for both of the input gates, we move on and do the same for each of them. Overall, if it's a valid pseudodistribution, we should always have an option with nonzero probability, and since each input gate must be at a lower layer by definition this means that we will only have to condition on $O(2^d)$ variables before we've determined the entire circuit.

Now, since this circuit does not in fact compute f, there is some particular input x for which it does not correctly compute $f(x)$. We can now in degree $O(2^d)$ derive all of the evaluation variables for input x, which will find a contradiction. So, we cannot in fact have started with a valid pseudodistribution. □

If we want to understand the limits of what circuit lower bounds SoS can prove, there's a couple of classes we could now look at in-between NC^0 and $\mathsf{AC}^0[\oplus]$. One slightly unusual option would be $\mathsf{NC}^0[\oplus]$ — that is, the class of constant-depth circuits with fan-in 2 AND and OR gates, and arbitrary fan-in $\oplus$ gates. These circuits compute low-degree functions over $\mathbf{F}_2$, so it's pretty easy to show that they can't compute something like the AND of all inputs. However, this reasoning about $\mathbf{F}_2$ seems like the sort of thing that SoS might have trouble with. It's conceivable that one might be able to rule this out with a construction analogous to what we did for $\mathsf{AC}^0[\oplus]$, but note that to do this we would need to be able to compute the neighbour relation of some appropriate unbalanced expander in $\mathsf{NC}^0[\oplus]$, which it is not currently known how to do. We do know that it's possible to compute the neighbour function of some *balanced* expander in NC^0 [VW18], but it's unclear if even with unbounded parity gates one can do this for unbalanced expanders. Oliviera, Santhanam and Tell have formulated this as an assumption relevant for security of some particular cryptographic scheme, so there has been at least some interest [OST18]. Note that if you could get low-degree SoS proofs of exponential circuit lower bounds against $\mathsf{NC}^0[\oplus]$ this would therefore rule out that assumption.

Another somewhat more traditional class to consider in-between NC^0 and $\mathsf{AC}^0[\oplus]$ would be AC^0: constant-depth arbitrary fan-in circuits with AND gates and OR gates, but no $\oplus$. Observe that we know that AC^0 can't compute the parity of n bits with $2^{o(n)}$ size — so, in particular, this means that the reduction we've been using to XOR-CSP will definitely not be implementable in AC^0. On the other side, we could ask whether known explicit lower bounds against AC^0 seem like they could be possible to translate into low-degree SoS. There are a number of different approaches to showing lower bounds against AC^0, but the primary two are probablistic polynomials (i.e. show by a probablistic argument that any AC^0 circuit has output well approximated by a certain low-degree polynomial over $\mathbb{F}_2$) or switching lemmas (i.e. show that a DNF under a random restriction is likely to have a small decision tree, and use this to swap a pair of layers from ORs of ANDs to ANDs of ORs and hence repeatedly shrink the circuit depth). The former of these approaches is almost certainly doomed in SoS-land: the exact same arguments work to show exponential lower bounds against $\mathsf{AC}^0[\oplus]$, and we know these should not be

[3] Or wasn't to me, at least. If you told me that actually this ought to have been obvious, I would believe you. Also if you told me my proof was wrong and it still wasn't obvious, I would unfortunately probably still believe you.

possible. For the latter, we don't currently have a formal way of ruling out, but the path ahead seems potentially fraught. The proof relies pretty heavily on having a particular circuit in front of you and being able to analyze how things simplify under random restrictions, which might be hard to try to capture when all you have is a pseudodistribution over circuits[4].

4 Feasible interpolation

In this section, we will briefly discuss the property of feasible interpolation. A propositional proof system is said to satisfy a feasible interpolation principle if, for any formulas $\phi(x, z)$ and $\psi(y, z)$ overlapping only on variables z, given a disproof of $\phi(x, z) \wedge \psi(y, z)$ and a particular assignment z' to the z variables, it is possible in polynomial time to compute a disproof of either $\phi(x, z')$ or $\phi(y, z')$.

4.1 Feasible interpolation and the provability of circuit lower bounds

The reason this is at all pertinent to the discussion at hand is that we know in a number of cases lower bounds on propositional proof complexity for certain statements against any propositional proof system satisfying a feasible interpolation property. In particular, *something* of the form "proof systems with feasible interpolations probably can't prove any lower bounds against general circuits" seems to be known. A paper where Steve Rudich states the following:

Theorem 3 (Rudich [Rud97]; paraphrased)**.** *If there exist pseudorandom generators computable in* P/poly *and fooling circuits of size* 2^{n^ϵ}*, then any "reasonable" propositional proof system satisfying feasible interpolation cannot prove a quasipolynomial circuit lower bound against any boolean function.*

However, Rudich does not give a formal definition of what "reasonable" means, except to say that it entails properties "common to all studied propositional proof systems", and attributes the proof of this theorem to Razborov in [Raz95]. The results in [Raz95] do not explicitly make any statements about general propositional proof systems — rather, Razborov rules out proofs of circuit lower bounds in various particular theories of bounded arithmetic. The authors of this project have unfortunately not been able to understand how these results translate, and so will not be able to give a formal definition of what "reasonable" means at the moment — however, if one takes Rudich on faith that the statement of Theorem 3 does follow from Razborov's results, and assumes that sum-of-squares presumably satisfies this "reasonableness" condition, then this would give an alternate approach to (conditionally) establishing similar results to those shown in Section 2: simply prove a feasible interpolation theorem for SoS. Of course, we don't really *need* to prove such a statement conditionally, given that that the results of Austrin and Risse give it unconditionally, but this implication could perhaps be interesting for the purposes of showing variations of the result. For instance, maybe this gives us an alternative route towards ruling out lower bounds against other circuit classes. (Although we note that this would still likely require pretty new arguments — for instance, Razborov's paper uses his techniques against theories of bounded arithmetic which *are* capable of proving lower bounds against AC^0, so his approach shouldn't be black-box capable of disproving AC^0 lower bounds.)

4.2 Feasible interpolation for SoS

For the remainder of this section, we'll sketch the result of Hakoniemi [Hak20] that SoS does satisfy a notion of feasible interpolation. Specifically, they show the following:

Theorem 4. *For any sets $P(x, z)$ and $Q(y, z)$ of multilinear polynomials, there is a polynomial-time (in the bit-complexity of the input proof) algorithm that takes an SoS refutation of $P(x, z) \cup Q(y, z)$ over the Boolean hypercube and an assignment z', and outputs either an SoS refutation of $P(x, z')$ or an SoS refutation of $Q(y, z')$.*

The proof follows two steps: first, show that a polynomial-sized refutation *exists* for either $P(x, z')$ or $Q(y, z')$, and then show that it can be found using the ellipsoid algorithm. The basic outline of the argument is as follows

[4] However I don't have great intuition for these things yet I think, so it's possible that you, the reader, are the better judge. If you, the reader, currently happen to be Sam Hopkins, then it is not just possible but in fact 100% true. All I know is that I couldn't figure out how to say anything.

(more care than we present here is needed to actually ensure that coefficients stay bounded).

Let's fix and ignore the z variables. Observe that an SoS refutation of $P \cup Q$ consists of a polynomial equality of the form

$$\sum_{i \in [k]} r_i^2 + \sum_{q \in Q \cup Q} t_q q = -1$$

for some polynomials r_i, t_i. Consider the set S of all monomials appearing in any of the r_i and t_i, and look at the "projections" S_x and S_y consisting of terms involving only x variables and only y variables, respectively. We will show that we can get a refutation for one of the two disjuncts using only those monomials. That is, we have the following:

Lemma 4.1. *If there is an SoS refutation of $P(x) \cup Q(y)$ with monomials S and coefficients of bounded bit-complexity, then there exists either an SoS refutation with bounded coefficients for $P(x)$ using only monomials S_x, or for $Q(y)$ using only monomials S_y.*

Proof. The first thing to observe is that, when we restrict the set of monomials an SoS proof is allowed to use, there still exists a notion of duality with pseudoexpectations. That is, you can define a pseudoexpectation *over monomial set* S to be a linear map from the degree-2 polynomials over S — that is, polynomials that can be written as a weighted sums of products of two monomials from S — to $\mathbb{R}$. We'll ask this to act like a degree-2 pseudoexpectation over variable set S (i.e. it's a pseudoexpectation defined over these monomials as opposed to the actual variables). That is, we require $\widetilde{\mathbb{E}}[1] = 1$, $\widetilde{\mathbb{E}}[m^2] \geq 0$ for any m that's a linear combination of monomials from S, and that $\widetilde{\mathbb{E}}[mq] = 0$ for any $m \in S$ and $q \in P \cup Q$. With this definition, we again get duality: for any $P \cup Q$ and any set S of monomials, either there exists such a pseudoexpectation for S, or there exists an SoS proof using only monomials S. (And, if you want an SoS proof with small coefficients, this corresponds to just relaxing the $\widetilde{\mathbb{E}}[mq] = 0$ condition to $\widetilde{\mathbb{E}}[mq] \leq \epsilon$.)

So, if we assume for contradiction that we have neither an SoS refutation for P over S_x nor an SoS refutation for Q over S_y, we get corresponding pseudoexpectations $\widetilde{\mathbb{E}}_x$ and $\widetilde{\mathbb{E}}_y$, respectively. We'll now define a new pseudoexpectation $\widetilde{\mathbb{E}}$ over all of S, by letting $\widetilde{\mathbb{E}}[m] = \widetilde{\mathbb{E}}_x[m_x] \cdot \widetilde{\mathbb{E}}_y[m_y]$, where by m_x and m_y we denote the projections onto S_x and S_y respectively. Observe that this satisfies $\widetilde{\mathbb{E}}[1] = 1$, and that $\widetilde{\mathbb{E}}[mq] = 0$ for $m \in S$, $p \in P$ because we'll have $\widetilde{\mathbb{E}}_x[m_x p_x] = 0$ (similarly for Q). To get that $\widetilde{\mathbb{E}}[m^2] \geq 0$ for all $m \in S$, we can just write in monomial basis and then expand out the sum, which will allow applying positive semi-definiteness of $\widetilde{\mathbb{E}}_x$ and $\widetilde{\mathbb{E}}_y$ separately. But now this pseudoexpectation contradicts our assumption that $P(x) \cup Q(x)$ had an SoS refutation over S. □

Proof of Theorem 4. Just as how we normally find low-degree SoS proofs efficiently, observe that the set of pseudoexpectations over S satisfying constraints $P \cup Q$ over the Boolean hypercube is convex, and so we can find a feasible point if one exists using the ellipsoid algorithm. □

5 Circuit lower bounds in other proof systems

We've seen in Section 2 that SoS requires large degree to prove lower bounds against general circuits for any function, and then in Section 4 that perhaps SoS's "feasible interpolation" property gives an explanation for why this should hold. Now, let's step beyond SoS and ask whether these sorts of circuit lower bounds are inherently hard for propositional proof systems more generally, even those which do not possess a feasible interpolation property. Specifically, we'll be covering a result of Pich and Santhanam [PS19] which, while not specifically about sum of squares, is nevertheless sufficiently delightful that we didn't think that singular deficiency should be enough to preclude inclusion.

If we're allowing sufficiently general propositional proof systems, we probably shouldn't expect something so strong as ruling out circuit lower bounds for *all* functions, but maybe we can rule out circuit lower bounds for *most* functions. Rudich has conjectured this to be the case, even if the proof system is allowed to have some nonuniform advice:

Definition 5.1. *A **propositional proof system** is a polynomial-time computable function that takes in a Boolean formula ϕ and a string π representing a proof and outputs whether or not the proof is valid, such that the statements with valid proofs are exactly the tautologies. A propositional proof system **with advice** additionally allows the proof verifier to depend on a polynomial-length fixed advice string fixed for each input length.*

Conjecture 5.2 (Rudich's conjecture)**.** *For any propositional proof system R with advice, for most (i.e. all but a negligible — less than $|f|^{-\omega(1)}$ — fraction of) inputs f, R requires proofs of super-polynomial length to show that f requires circuits of super-polynomial size.*

Here, we're using a propositional encoding of the question of whether a circuit exists computing a given truth table — the exact form of the encoding doesn't particularly matter; it can be taken as anything roughly analogous to the polynomial system we presented in Section 2. For a function $f\colon \{0,1\}^n \to \{0,1\}$, we'll denote by "HasCircuit(f, s)" the proposition encoding that f has a circuit of size s. For a propositional proof system R and a proposition ϕ, we can similarly write "HasProof(R, ϕ, s)" to denote the proposition encoding that R has a proof of ϕ of size s. The main result of Pich and Santhanam [PS19] is the unconditional result that Rudich's conjecture *itself* does not have short proofs in any propositional proof system S. That is, they show the following:

Theorem 5. *There exists a constant c and a propositional proof system R with advice, such that any propositional proof system S with advice S lacks polynomial-size proofs of $\neg$HasProof$(R, \neg\text{HasCircuit}(f, n^c), 2^n)$ for all but a negligible fraction of functions $f\colon \{0,1\}^n \to n$.*

Rudich's conjecture would hold that $\neg$HasProof$(R, \neg\text{HasCircuit}(f, n^c), 2^n)$ is a tautology for the vast majority of f. However, Theorem 5 tells us that nevertheless, for the vast majority of f, your favourite propositional proof system fails to prove this tautology.

If Rudich's conjecture is false, Theorem 5 is not so surprising: of course S shouldn't be able to prove these statements if they're not even true most of the time. Theorem 5 is not quite immediate from falsehood of Rudich's conjecture, but it's pretty close, and this is definitely not the interesting case. The more exciting case is if Rudich's conjecture is true. In this case, the idea will be to use the statement of Rudich's conjecture to get a hitting set against nondeterministic circuits, and then design a proof system R with "planted" proofs of circuit lower bounds corresponding to the elements of that hitting set. First, recall the definition of a hitting set, stated here specifically for nondeterministic circuits:

Definition 5.3. *A set $H_N \subseteq \{0,1\}^N$ is an ϵ-hitting set against size-s nondeterministic circuits if any nondeterministic circuit of size s which accepts at least an ϵ fraction of $\{0,1\}^N$ accepts at least one element of H_N.*

Note that Rudich's theorem implies (and is roughly equivalent to) the following statement:

Conjecture 5.4 (Rudich's conjecture, equivalent form)**.** *There exists a c such that for any constant k, the set of truth tables of size-n^c circuits is a $(1/(2^n)^k)$-hitting set against size-$(2^n)^k$ nondeterministic circuits.*

Proof of Conjecture 5.2 $\implies$ Conjecture 5.4. Suppose we had a nondeterministic circuit M on N-bit inputs for $N = 2^n$, such that M accepted at least a $1/N^k$ fraction of inputs, but didn't accept the truth table of any circuit of size n^c. Since M never accepts the truth table of a circuit of size n^c, we could get a sound proof system by taking your favourite propositional proof system, and then gluing on to it a subroutine that says "if the proposition you're trying to prove is of the form $\neg$HasCircuit(f, n^c), also try treating the proof as input to M and accept the proof if M accepts". But now this proof system has polynomial-size proofs of $\neg$HasCircuit(f, n^c) for a $1/\text{poly}(N)$ fraction of f. □

We will use this hitting set to design a tricky proof system R.

Proof of Theorem 5. We'll start by assuming that Rudich's conjecture is true; we can handle the case where Rudich's conjecture is false at the end. By Conjecture 5.4, we therefore know that the set H_N of truth tables of n^c-size circuits on n-bit inputs is a $(1/N^k)$-hitting set for size-N^k nondeterministic circuits for every k (where we're again denoting $N = 2^n$). Let's let $z_N \in \{0,1\}^N$ denote the truth table of some function $\{0,1\}^n \to \{0,1\}$ with maximal Boolean complexity. We claim that $H' = \{x \oplus z_N \mid x \in H\}$ is also a $(1/N^k)$-hitting set for size-N^k nondeterministic circuits — this is because given a circuit breaking the hitting property of H', we could get a

circuit breaking the hitting property of H just by hardcoding in z_N and having the circuit flip the appropriate bits.

Now, we'll use H' to design a proof system R. R will consist of your favourite propositional proof system, except that if the proposition to be proved is of the form $\neg\text{HasCircuit}(f, n^c)$, and the proof consists of a circuit of size n^c, it will additionally check whether $f \oplus \text{tt} = z_N$, where tt is the truth table of that circuit, and accept if so. First, note that this verification can still be done in polynomial size: we just need to hardcode z_N, and then do $n^c \cdot 2^n$ work to compute the truth table. Also, note that we haven't harmed soundness of the proof system, since if $f \oplus \text{tt} = z_N$ it must be the case that f requires circuits of size much larger than n^c (otherwise, we could get a small circuit for z_N by taking the XOR of a circuit for f and the circuit computing tt). Finally, observe that R has small proofs — in fact, proofs of size $n^c \ll N$ — for $\neg\text{HasCircuit}(f, n^c)$ whenever $f \in H'$.

Let's fix a new propositional proof system S, and assume for contradiction that S has poly-size proofs of $\neg\text{HasProof}(R, \neg\text{HasCircuit}(f, n^c), N)$ on at least a $1/N^k$ fraction of f, for some constant k. But now, consider the nondeterministic circuit which, on input f, guesses a poly-sized S-proof of $\neg\text{HasProof}(R, \neg\text{HasCircuit}(f, n^c), N)$, runs it through the S-verifier, and accepts if the proof verifies. Since we have a poly-size verifier and are guessing a poly-size proof, this nondeterministic circuit can be implemented in time N^d for some constant d. Observe that this circuit will accept at least a $1/N^k$ fraction of all f, but that by soundness of S it will never accept any f for which R has a size-N proof of $\neg\text{HasCircuit}(f, n^c)$. But this is a contradiction: this circuit must accept an element of H' because we showed H' is a $(1/N^k)$-hitting set for nondeterministic size-N^d circuits, but it cannot accept any element of H' because they all have short R-proofs of $\neg\text{HasCircuit}(f, n^c)$.

The final case to consider is that Rudich's conjecture is false. This gives us for any d a proof system R with advice that has N^k-size proofs of $\neg\text{HasCircuit}(f, n^d)$ for a N^{-k} fraction of all f. All that remains is to boost this to a proof system R' with poly-size proofs of $\neg\text{HasCircuit}(f, n^c)$ on all but a negligible fraction of f — then, no proof system S can prove $\neg\text{HasProof}(R, \neg\text{HasCircuit}(f, n^c), N)$ more than a negligible fraction of the time simply because $\neg\text{HasProof}(R, \neg\text{HasCircuit}(f, n^c), N)$ is only true a negligible fraction of the time. To do this boosting, we just have R' split the input truth table into $N^{1-1/3k}$ equal-sized chunks, and then accept if its given R-proof ruling out circuits of size n^d for any given chunk, since this will also then rule out small circuits for the whole function. If we choose $d = 100k^{100}c$, then our guarantees on R tell us that we'll have such a proof for any given chunk with probability at least $1 - N^{-1/3}$, so since the probabilities for each chunk are independent our overall probability of having a proof is at least $1 - \left(1 - \frac{1}{N^{1/3}}\right)^{N^{2/3}} = 1 - e^{-\Omega(N^{1/3})} = 1 - N^{-\omega(1)}$. □

References

[AR23] Per Austrin and Kilian Risse. "Sum-Of-Squares Lower Bounds for the Minimum Circuit Size Problem". In: *38th Computational Complexity Conference (CCC 2023)*. Schloss Dagstuhl-Leibniz-Zentrum für Informatik. 2023.

[Gri01] Dima Grigoriev. "Linear lower bound on degrees of Positivstellensatz calculus proofs for the parity". In: *Theoretical Computer Science* 259.1-2 (2001), pp. 613–622.

[GUV09] Venkatesan Guruswami, Christopher Umans, and Salil Vadhan. "Unbalanced expanders and randomness extractors from Parvaresh–Vardy codes". In: *Journal of the ACM (JACM)* 56.4 (2009), pp. 1–34.

[Hak20] Tuomas Hakoniemi. "Feasible Interpolation for Polynomial Calculus and Sums-Of-Squares". In: *47th International Colloquium on Automata, Languages, and Programming (ICALP 2020)*. Ed. by Artur Czumaj, Anuj Dawar, and Emanuela Merelli. Vol. 168. Leibniz International Proceedings in Informatics (LIPIcs). Dagstuhl, Germany: Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2020, 63:1–63:14. ISBN: 978-3-95977-138-2. DOI: 10.4230/LIPIcs.ICALP.2020.63. URL: https://drops.dagstuhl.de/entities/document/10.4230/LIPIcs.ICALP.2020.63.

[OST18] I Oliveira, Rahul Santhanam, and Roei Tell. "Expander-based cryptography meets natural proofs". In: *Innovations in Theoretical Computer Science (ITCS)* 124 (2018).

[PS19] Jan Pich and Rahul Santhanam. "Why are Proof Complexity Lower Bounds Hard?" In: *2019 IEEE 60th Annual Symposium on Foundations of Computer Science (FOCS)*. 2019, pp. 1305–1324. DOI: 10.1109/FOCS.2019.00080.

[Raz95] Alexander A Razborov. "Unprovability of lower bounds on circuit size in certain fragments of bounded arithmetic". In: *Izvestiya: mathematics* 59.1 (1995), p. 205.

[Rud97] Steven Rudich. "Super-bits, demi-bits, and NP/qpoly-natural proofs". In: *International Workshop on Randomization and Approximation Techniques in Computer Science*. Springer. 1997, pp. 85–93.

[VW18] Emanuele Viola and Avi Wigderson. "Local expanders". In: *computational complexity* 27 (2018), pp. 225–244.