Semi-algebraic Proofs, IPS Lower Bounds, and the τ-Conjecture: Can a Natural Number Be Negative?*

Yaroslav Alekseev
Steklov Institute of Mathematics and Chebyshev
Laboratory at St. Petersburg State University
St. Petersburg, Russia

Dima Grigoriev†
CNRS, Mathematiques, Universite de Lille
Villeneuve d'Ascq, 59655
Lille, France

Edward A. Hirsch‡
Steklov Institute of Mathematics at St. Petersburg
St. Petersburg, Russia

Iddo Tzameret§
Department of Computer Science
Royal Holloway, University of London
Egham, UK

ABSTRACT

We introduce the *binary value principle* which is a simple subset-sum instance expressing that a natural number written in binary cannot be negative, relating it to central problems in proof and algebraic complexity. We prove conditional superpolynomial lower bounds on the Ideal Proof System (IPS) refutation size of this instance, based on a well-known hypothesis by Shub and Smale about the hardness of computing factorials, where IPS is the strong algebraic proof system introduced by Grochow and Pitassi [*J. ACM*, 65(6):37:1–55, 2018]. Conversely, we show that short IPS refutations of this instance bridge the gap between sufficiently strong algebraic and semi-algebraic proof systems. Our results extend to unrestricted IPS the paradigm introduced in Forbes, Shpilka, Tzameret and Wigderson [*Proceedings of CCC*, 2016, pp. 32:1–32:17] whereby lower bounds against subsystems of IPS were obtained using restricted algebraic circuit lower bounds, and demonstrate that the binary value principle captures the advantage of semi-algebraic over algebraic reasoning, for sufficiently strong systems. Specifically, we show the following:

(1) **Conditional IPS lower bounds**: The Shub-Smale hypothesis [*Duke Math. J.*, 81:47-54, 1995] implies a superpolynomial lower bound on the size of IPS refutations of the binary value principle over the rationals defined as the unsatisfiable linear equation $\sum_{i=1}^{n} 2^{i-1}x_i = -1$, for boolean x_i's. Further, the related τ-conjecture [*Duke Math. J.*, 81:47-54, 1995] implies a superpolynomial lower bound on the size of IPS refutations of a variant of the binary value principle over the ring of rational functions. No prior conditional lower bounds were known for IPS or for apparently much weaker propositional proof systems such as Frege.

(2) **Algebraic vs. semi-algebraic proofs**: Admitting short refutations of the binary value principle is necessary for any algebraic proof system to fully simulate any known semi-algebraic proof system, and for strong enough algebraic proof systems it is also *sufficient*. In particular, we introduce a very strong proof system that simulates all known semi-algebraic proof systems (and most other known concrete propositional proof systems), under the name Cone Proof System (CPS), as a semi-algebraic analogue of the ideal proof system: CPS establishes the unsatisfiability of collections of polynomial equalities and inequalities over the reals, by representing *sum-of-squares* proofs (and extensions) as algebraic circuits. We prove that IPS is polynomially equivalent to CPS iff IPS admits polynomial-size refutations of the binary value principle (for the language of systems of equations that have no 0/1-solutions), over both $\mathbb{Z}$ and $\mathbb{Q}$.

CCS CONCEPTS

• **Theory of computation** → **Proof complexity**; **Algebraic complexity theory**; **Circuit complexity**; Problems, reductions and completeness.

KEYWORDS

Proof complexity, Semi-algebraic proofs, sum-of-squares proofs, algebraic proofs, algebraic complexity, tau-conjecture

ACM Reference Format:
Yaroslav Alekseev, Dima Grigoriev, Edward A. Hirsch, and Iddo Tzameret. 2020. Semi-algebraic Proofs, IPS Lower Bounds, and the τ-Conjecture: Can a Natural Number Be Negative?. In *Proceedings of the 52nd Annual ACM SIGACT Symposium on Theory of Computing (STOC '20), June 22–26, 2020, Chicago, IL, USA.* ACM, New York, NY, USA, 14 pages. https://doi.org/10.1145/3357713.3384245

1 INTRODUCTION

This work connects three separate objects of study in computational complexity: algebraic proof systems, semi-algebraic proof systems

[*] A preliminary report on parts of this work was delivered at Dagstuhl Proof Complexity Workshop 2018. https://materials.dagstuhl.de/files/18/18051/18051.IddoTzameret.Slides.pptx. The research presented in Section 3 is supported by Russian Science Foundation (project 16-11-10123).

[†] http://en.wikipedia.org/wiki/Dima_Grigoriev

[‡] http://logic.pdmi.ras.ru/~hirsch

[§] Iddo.Tzameret@gmail.com http://www.cs.rhul.ac.uk/home/tzameret

STOC '20, June 22–26, 2020, Chicago, IL, USA

ACM ISBN 978-1-4503-6979-4/20/06…$15.00
https://doi.org/10.1145/3357713.3384245

and algebraic circuit complexity. The connecting point is a subset-sum instance expressing that the value of a natural number given in binary is nonnegative. We will show that this instance captures the advantage of semi-algebraic reasoning over algebraic reasoning in the regime of sufficiently strong proof systems, and is expected to be hard even for very strong algebraic proof systems. We begin with a general discussion about proof complexity, and then turn to algebraic and semi-algebraic proofs, their inter-relations, and the connection between circuit lower bounds and proof-size lower bounds.

Narrowly construed, proof complexity can be viewed as a stratification of the NP vs. coNP question, whereby one aims to understand the complexity of stronger and stronger propositional proof systems as a gradual approach towards separating NP from coNP (and hence, also P from NP). This mirrors circuit complexity in which different circuit classes are analyzed in the hope to provide general super-polynomial circuit lower bounds. Broadly understood however, proof complexity serves as a way to study the computational resources required in different kind of reasoning, different algorithmic techniques and constraint solvers, as well as providing propositional analogues to weak first-order theories of arithmetic.

Algebraic proof systems have attracted immense amount of work in proof complexity, due to their simple nature, being a way to study the complexity of computer-algebra procedures such as the Gröbner basis algorithm, and their connection to different fragments of propositional proof systems with counting gates. Beginning with the fairly weak Nullstellensatz refutation system by Beame et al. [4] and culminating in the very strong Ideal Proof System by Grochow and Pitassi [25], many algebraic proof systems and variants have been studied. In such systems one basically operates with polynomial equations over a field using simple algebraic derivation rules such as additions of equations and multiplication of an equation by a variable, where variables are usually meant to range over $\{0, 1\}$ values.

Impagliazzo, Pudlák and Sgall [30], following Razborov [43], showed that the polynomial calculus, which is the standard dynamic algebraic proof system introduced in [13], requires exponential-size refutations (namely, those using an exponential number of monomials) for the simple symmetric unsatisfiable subset-sum instance $x_1 + \cdots + x_n = n + 1$. Note that refuting (that is, showing the unsatisfiability of) a linear equation $\sum_i \alpha_i x_i = \beta$ in which the variables x_i are boolean, establishes that there is no subset of the α_i numbers that sums up to β, and hence is considered to be a refutation of a subset-sum instance. Forbes *et al.* [18] showed that variants of this symmetric subset-sum instance are hard for different subsystems of the very strong IPS algebraic proof system, that is, when IPS refutations are written using various restricted algebraic circuit classes. Loosely speaking, IPS is a static Nullstellensatz refutation in which proof-size is measured by algebraic circuit complexity instead of sparsity (that is, monomial size). In other words, IPS proofs are written as algebraic circuits, and thus can tailor the advantage that algebraic circuits have over sparse polynomials (somewhat reminiscent to the way Extended Frege can tailor the full strength of boolean circuits in comparison to resolution which operates merely with clauses).

The realm of semi-algebraic proof systems has emerged as an equally fruitful subject as algebraic proofs. Semi-algebraic proofs have been brought to the attention of complexity theory from optimization [34, 35] by the works of Pudlák [41] and Grigoriev and Vorobojov [24] (cf. [23]), and more recently, through their connection to approximation algorithms with the work of Barak *et al.* [3] (see for example [36] and the new excellent survey by Fleming, Kothari and Pitassi [17]). While algebraic proofs derive polynomials in the ideal of a given initial set of polynomials, semi-algebraic proofs extend it to allow deriving polynomials also in the cone of the initial polynomials (informally a cone is an "ideal that preserves positive signs"), hence potentially utilizing a stronger kind of reasoning. In particular [3] considered the *sum-of-squares* (SoS) refutation system. What makes SoS important, for example to polynomial optimization, is the fact that the existence of a degree-d SoS certificate can be formulated as the feasibility of a semidefinite program (SDP), and hence can be solved in polynomial time.

Berkholz [5] showed interestingly that in the regime of *weak* proof systems, even static semi-algebraic proofs, such as SoS, can simulate dynamic algebraic proof systems such as polynomial calculus. Grigoriev [22] showed that in this weak regime semi-algebraic proofs are in fact strictly stronger (with respect to degrees and size) than algebraic proofs, where the separating instances are simple polynomials (for example, symmetric subset sum instances). However, it was not known in general (e.g., for strong systems) whether semi-algebraic reasoning is strictly stronger than algebraic reasoning.

Another established tradition in proof complexity is to seek synergies between proofs and circuit lower bounds. In particular, *proofs-to-circuits* transformations in the form of feasible interpolation, and other close concepts have been pivotal in the search for proof complexity lower bounds, as well as in circuit lower bounds themselves (see Göös, Kamath, Robere and Sokolov [20] for a recent example). In fact, the conception of IPS itself was motivated by the attempt to show that very strong proof complexity lower bounds would result in algebraic complexity class separations such as VP $\neq$ VNP (see [25] and the survey [40]). Li, Tzameret and Wang [33] as well as Forbes *et al.* [18] went in the other direction and showed that certain restricted algebraic circuit lower bounds imply size lower bounds on subsystems of IPS. In particular, [18] devised a simple framework by which lower bounds on (subsystems of) IPS refutations are reduced to algebraic circuit lower bounds. [18] used this framework to establish lower bounds on subsystems of IPS refutations of variants of symmetric subset-sum instances when the IPS refutations are written as read once algebraic branching programs and multilinear formulas. But lower bounds on the size of unrestricted IPS refutations were not known.

2 PRELIMINARIES

2.1 Notation

For a natural number we let $[n] = \{1, \ldots, n\}$. Let R be a ring. Denote by $R[x_1, \ldots, x_n]$ the ring of multivariate polynomials with coefficients from R and variables $x_1, \ldots, x_n$. We usually denote by $\overline{x}$ the vector of variables $x_1, \ldots, x_n$. We treat polynomials as *formal* linear combination of monomials, where a monomial is a product of variables. Hence, when we talk about the *zero polynomial* we mean the polynomial in which the coefficients of all monomials are zero. Similarly, two polynomials are said to be *identical* if their monomials have the same coefficients. The *number of monomials* in a polynomial f is the number of monomials with nonzero coefficients denoted $|f|_{\#\text{monomials}}$. The *degree* of a multivariate polynomial (or total degree) is the maximal sum of variable powers in a monomial with a nonzero coefficient in the polynomial. We write $\text{poly}(n)$ to denote a polynomial growth in n, namely a function that is upper bounded by cn^c, for some constant c independent of n. Similarly, $\text{poly}(n_1, \ldots, n_s)$ for some constant s, means a polynomial growth that is at most $kn_1^{c_1} \cdots n_s^{c_s}$, for k and c_{ji}'s that are constants independent of $n_1, \ldots, n_s$.

For S a set of polynomials from $R[x_1, \ldots, x_n]$, we denote by $\langle S \rangle$ the *ideal generated by* S, namely the minimal set containing S such that if $f, g \in \langle S \rangle$ then also $\alpha f + \beta g \in \langle S \rangle$, for any $\alpha, \beta \in R$.

2.2 Algebraic Circuits

Algebraic circuits over some fixed chosen field or ring R compute polynomials in $R[x_1, \ldots, x_n]$ via addition and multiplication gates, starting from the input variables $\overline{x}$ and constants from the field. More precisely, an *algebraic circuit* C is a finite directed acyclic graph where edges are directed from leaves (that is, in-degree 0 nodes) towards the output nodes (that is out-degree 0 nodes). By default, there is a single output node. *Input nodes* are in-degree 0 nodes that are labeled with a variable from $x_1, \ldots, x_n$; every other in-degree zero node is labelled with a scalar element in R. All the other nodes have in-degree two (unless otherwise stated) and are labeled with either + or ×. An in-degree 0 node is said to *compute* the variable or scalar that labels itself. A + (or ×) gate is said to compute the addition (product, resp.) of the polynomials computed by its incoming nodes. The ***size*** of an algebraic circuit C is the number of nodes in it denoted $|C|$, and the *depth* of a circuit is the length of the longest directed path in it. Note that the size of a field coefficient in this setting is 1 irrespective of the value of the coefficient. Sometimes it is important to consider the size of the coefficients appearing in the circuit (for instance, when we are concerned with the computational complexity of problems pertaining to algebraic circuits we need to have an efficient way to represent the circuits as bit strings). For this purpose it is standard to define a ***constant-free*** algebraic circuit to be an algebraic circuit in which the only constants used are $0, 1, -1$. Other constants must be built up using algebraic operations, which then count towards the size of the circuit.

An algebraic circuit is said to be a *multi-output* circuit if it has more than one output node, namely, more than one node of out-degree zero. Given a single-output algebraic circuit $F(\overline{x})$ we denote by $\widehat{F}(\overline{x}) \in R[\overline{x}]$ the *polynomial* computed by $F(\overline{x})$. We define the *degree* of a circuit C (similarly a node) as the total degree of the polynomial $\hat{C}$ computed by C, denoted $\deg(C)$.

We will also use circuits that have division gates; when we need them, we define them explicitly.

Algebraic Complexity Classes. We now recall some basic notions from algebraic complexity (for more details see [45, Sec. 1.2]). Over a ring R, VP_R (for "Valiant's P") is the class of families $f = (f_n)_{n=1}^{\infty}$ of formal polynomials f_n such that f_n has $\text{poly}(n)$ input variables, is of $\text{poly}(n)$ degree, and can be computed by algebraic circuits over R of $\text{poly}(n)$ size. VNP_R (for "Valiant's NP") is the class of families g of polynomials g_n such that g_n has $\text{poly}(n)$ input variables and is of $\text{poly}(n)$ degree, and can be written as $g_n\left(x_1, \ldots, x_{\text{poly}(n)}\right) = \sum_{\overline{e} \in \{0,1\}^{\text{poly}(n)}} f_n(\overline{e}, \overline{x})$ for some family $(f_n) \in \mathsf{VP}_R$. A major question in algebraic complexity theory is whether the permanent polynomial can be computed by algebraic circuits of polynomial size. Since the permanent is complete for VNP (under a suitable concept of algebraic reductions that are called p-projections), showing that no polynomial-size circuit can compute the permanent amounts to showing VP≠VNP (cf. [50–52]).

Similarly, we can consider the *constant-free* versions of VP and VNP: we denote by VP^0 and VNP^0 the class of polynomial-size and polynomial-degree *constant-free* algebraic circuits and the class of VNP polynomials as above in which the family of polynomials $(f_n) \in \mathsf{VP}^0$. In these definitions of VP^0 and VNP^0 we assume also that no division gate occurs in the circuits, hence VP^0 and VNP^0 compute polynomials over $\mathbb{Z}$. The permanent is still complete for VNP^0. We will also consider in 3 constant-free circuits *over* $\mathbb{Q}$: these will be constant-free circuits in which constant sub-circuits (and *only* constant sub-circuits) may contain division gates.

2.3 The τ-Conjecture and Shub-Smale Hypothesis

Here we explain several important assumptions and conjectures that are known to lead to strong complexity lower bounds and complexity class separations, all of which are relevant to our work. See for example Smale's list of "mathematical problems for the next century" [47] for a short description and discussion about these problems.

Definition 1 (τ-function [46])**.** *Let $f \in \mathbb{Z}[\overline{x}]$ be a multivariate polynomial over $\mathbb{Z}$. Then $\tau(f)$ is the minimal size of a constant-free algebraic circuit that computes f (that is, a circuit where the only possible constants that may appear on leaves are* $1, 0, -1$*).*

When we focus on constant polynomials, that is, numbers $n \in \mathbb{Z}$, $\tau(n)$ is the minimal-size circuit that can construct n from 1 using additions, subtractions and multiplications (but not divisions; note that a subtraction of a term A can be constructed by $-1 \cdot A$).

We say that a family of (possibly constant) polynomials $(f_n)_{n \in \mathbb{N}}$ is ***easy*** if $\tau(f_n) = \log^{O(1)} n$, for every $n > 2$, and ***hard*** otherwise.[1]

The following are some known facts about $\tau(\cdot)$:

- $(2^n)_{n \in \mathbb{N}}$ is *easy*. For instance, if n is a power of 2 then $\tau(2^n) = \log n + 3$, where log denotes the logarithm in the base 2. We start with 3 nodes to build $2 = 1+1$ and then by $\log n$ repeated squaring we arrive at $((2^2)^2)^2 \ldots)^2 = 2^{2^{\log n}} = 2^n$.
- $(2^{2^n})_{n \in \mathbb{N}}$ is *hard*. This stems from the straightforward upper bound on the largest integer that can be computed with k multiplication/addition/subtraction gates.
- A simple known upper bound on τ is this [16]: for every integer $m > 2$, $\tau(m) \leq 2 \log m$. This is shown by considering the binary expansion of m.
- For every integer m, the following lower bound is known $\tau(m) \geq \log \log m$ [16].

[1] We put the condition $n > 2$ instead of $n \geq 1$, because unlike [46] we do not add the constant 2 to the constants available in the circuit, hence to keep the same known upper bounds for τ we skip the cases $n = 1, 2$.

While $(2^n)_{n\in\mathbb{N}}$ is easy and $(2^{2^n})_{n\in\mathbb{N}}$ is hard, it is not known whether $(n!)_{n\in\mathbb{N}}$ is easy or hard, and as seen below, showing the hardness of $\tau(m_n \cdot n!)$, for every sequence $(m_n \cdot n!)_{n\in\mathbb{N}}$ with any $m_n \in \mathbb{Z}$ nonzero integers, has very strong consequences.

Blum, Shub and Smale [8] introduced an algebraic version of Turing machines that has access to a field K (Poizat observed that their model can be defined as algebraic circuits in which *selection* gates $s(z, x, y)$ can be used; where a selection gate outputs x in case $z = 0$ and y in case $z = 1$). In this model one can formalise and study a variant of the P vs. NP problem for languages solvable by polynomial-time machines with access to K, denoted P_K, versus nondeterministic polynomial-time machines with access to K, denoted NP_K.

The following is a condition put forth by Shub and Smale [46] (cf. [47]) towards separating $\mathsf{P}_\mathbb{C}$ from $\mathsf{NP}_\mathbb{C}$, for $\mathbb{C}$ the complex numbers:

Shub-Smale Hypothesis ([46, 47]). *For every nonzero integer sequence $(m_n)_{n\in\mathbb{N}}$, the sequence $(m_n \cdot n!)_{n\in\mathbb{N}}$ is hard.*

Shub and Smale, as well as Bürgisser, showed the following consequences of the Shub-Smale hypothesis:

Theorem 2.1 ([9, 46]). *(1) If the Shub-Smale hypothesis holds then $\mathsf{P}_\mathbb{C} \neq \mathsf{NP}_\mathbb{C}$.*
(2) If the Shub-Smale Hypothesis holds then $\mathsf{VP}^0 \neq \mathsf{VNP}^0$. In other words, Shub-Smale Hypothesis implies that the permanent does not have polynomial size constant-free algebraic circuits over $\mathbb{Z}$.

It is open whether the Shub-Smale hypothesis holds. What is known is that if Shub-Smale hypothesis does *not* hold then factoring of integers can be done in (nonuniform) polynomial time (cf. Blum et al. [7, p.126] and [12]).

Another related important assumption in algebraic complexity is the *τ-conjecture*. Let $f \in \mathbb{Z}[x]$ be a univariate polynomial with integer coefficients, denote by $z(f)$ the number of distinct integer roots of f.

τ-Conjecture ([46, 47]). *There is a universal constant c, such that for every univariate polynomial $f \in \mathbb{Z}[x]$:*

$$(1 + \tau(f))^c \geq z(f)\,.$$

The consequences of the τ-conjecture are similar to the Shub-Smale Hypothesis:

Theorem 2.2 ([9, 46]). *If the τ-conjecture holds then both $\mathsf{P}_\mathbb{C} \neq \mathsf{NP}_\mathbb{C}$ and $\mathsf{VP}^0 \neq \mathsf{VNP}^0$ hold.*

2.4 Basic Proof Complexity

In the standard setting of propositional proof complexity, a *propositional proof system* [15] is a polynomial-time predicate $V(\pi, x)$ that verifies purported proofs π (encoded naturally in, say, binary) for propositional formulas x (also encoded naturally in binary), such that $\exists\pi\ (V(\pi, x) = \text{true})$ iff x is a tautology. Hence, a propositional proof system is a complete and sound proof system for propositional logic in which a proof can be checked for correctness in polynomial time (though, note that a proof π may be exponentially larger than the tautology x it proves).

When considering algebraic proof systems that operate with algebraic circuits, such as IPS, it is common to relax the notion of a propositional proof system, so to require that the relation $V(\pi, x)$ is in probabilistic polynomial time, instead of deterministic polynomial time (since polynomial identities can be verified in coRP, while not known to be verified in P).

Furthermore, the language that a given proof system proves, namely the set of instances that the proof system proves to be tautological, or always satisfied, can be different from the set of propositional tautologies. First, we can consider a propositional proof system to be a *refutation system* in which a proof establishes that the initial set of axioms (e.g., clauses) is *unsatisfiable*, instead of always satisfied (i.e., tautological). For most cases, considering a propositional proof system to be a refutation system preserves all properties of the proof system, and thus the notions of refutation and proofs are used as synonyms. Second, we can define a proof system to be complete and sound for languages different or larger than unsatisfiable propositional formulas. For instance, in algebraic proof systems we usually consider proof systems that are sound and complete for *the language of unsatisfiable sets of polynomial equations*. This language includes propositional logic by a simple encoding, but it is strictly greater than propositional logic.

For the purpose of comparing the relative complexity of different proof systems we have the concept of *simulation*: given two proof systems P, Q for the *same* language, we say that P ***simulates*** Q if there is a function f that maps Q-proofs to P-proofs of the same instances with at most a polynomial blow-up in size. If f can be computed in polynomial time, this is called a ***p-simulation***. If P and Q simulate each other we say that P and Q are *polynomially-equivalent*. If P and Q are two proof systems for *different* languages, prima facie we cannot compare their strength via the notion of simulation. However, if both P and Q prove (or refute) propositional instances like formulas in conjunctive normal form, or boolean tautologies, while encoding them in different ways (namely, they use different representations for essentially the same propositional formulas), we can fix a polynomial-time computable translation from one representation to the other. Under this translation we can consider P and Q to be proof systems for the *same* language, allowing us to use the notion of simulation between P and Q.

2.5 Algebraic Proofs

Grochow and Pitassi [25] suggested the following algebraic proof system which is essentially a Nullstellensatz proof system [4] written as an algebraic circuit. A proof in the Ideal Proof System is given as a *single* polynomial. We provide below the *boolean* version of IPS (which includes the boolean axioms), namely the version that establishes the unsatisfiability over 0-1 of a set of polynomial equations. In what follows we follow the notation in [18]:

Definition 2 ((boolean) Ideal Proof System (IPS), Grochow-Pitassi [25]). *Let $f_1(\overline{x}), \ldots, f_m(\overline{x}), p(\overline{x})$ be a collection of polynomials in $\mathbb{F}[x_1, \ldots, x_n]$ over the field $\mathbb{F}$. An* ***IPS proof of*** *$p(\overline{x}) = 0$* ***from*** *$\{f_j(\overline{x}) = 0\}_{j=1}^m$, showing that $p(\overline{x}) = 0$ is semantically implied from* *the assumptions* $\{f_j(\overline{x}) = 0\}_{j=1}^m$ *over* 0-1 *assignments, is an algebraic circuit* $C(\overline{x}, \overline{y}, \overline{z}) \in \mathbb{F}[\overline{x}, y_1, \ldots, y_m, z_1, \ldots, z_n]$ *such that (the equalities in what follows stand for formal polynomial identities*[2]*):*

(1) $C(\overline{x}, \overline{0}, \overline{0}) = 0$*; and*
(2) $C(\overline{x}, f_1(\overline{x}), \ldots, f_m(\overline{x}), x_1^2 - x_1, \ldots, x_n^2 - x_n) = p(\overline{x})$*.*

The ***size of the IPS proof*** *is the size of the circuit* C*. If* C *is assumed to be constant-free, we refer to the size of the proof as the* ***size of the constant-free IPS proof****. The variables* $\overline{y}, \overline{z}$ *are called the* placeholder variables *since they are used as placeholders for the axioms. An IPS proof* $C(\overline{x}, \overline{y}, \overline{z})$ *of* $1 = 0$ *from* $\{f_j(\overline{x}) = 0\}_{j\in[m]}$ *is called an* ***IPS refutation*** *of* $\{f_j(\overline{x}) = 0\}_{j\in[m]}$ *(note that in this case it must hold that* $\{f_j(\overline{x}) = 0\}_{j=1}^m$ *have no common solutions in* $\{0, 1\}^n$*).*

Notice that the definition above adds the equations $\{x_i^2 - x_i = 0\}_{i=1}^n$, called the set of ***boolean axioms*** denoted $\overline{x}^2 - \overline{x}$, to the system $\{f_j(\overline{x}) = 0\}_{j=1}^m$. This allows to refute over $\{0, 1\}^n$ unsatisfiable systems of equations. Also, note that the first equality in the definition of IPS means that the polynomial computed by C is in the ideal generated by $\overline{y}, \overline{z}$, which in turn, following the second equality, means that C witnesses the fact that 1 is in the ideal generated by $f_1(\overline{x}), \ldots, f_m(\overline{x}), x_1^2 - x_1, \ldots, x_n^2 - x_n$ (the existence of this witness, for unsatisfiable set of polynomials, stems from the Nullstellensatz theorem [4]).

In order to use IPS as a propositional proof system (namely, a proof system for propositional tautologies), we need to fix the encoding of clauses as algebraic circuits.

Definition 3 (algebraic translation of CNF formulas)**.** *Given a CNF formula in the variables* $\overline{x}$*, every clause* $\bigvee_{i\in P} x_i \vee \bigvee_{j\in N} \neg x_j$ *is translated into* $\prod_{i\in P}(1 - x_i) \cdot \prod_{j\in N} x_j = 0$*. (Note that these terms are written as algebraic circuits as displayed, where products are not multiplied out.)*

Notice that in this way a 0-1 assignment to a CNF is satisfying iff the assignment is satisfying all the equations in the algebraic translation of the CNF. Therefore, using Definition 3 to encode CNF formulas, boolean IPS is considered as a propositional proof system for the language of unsatisfiable CNF formulas, sometimes called ***propositional IPS***. We say that an IPS proof is an ***algebraic IPS*** proof, if we do not use the boolean axioms $\overline{x}^2 - \overline{x}$ in the proof. *As a default when referring to IPS we mean the boolean IPS version.*

2.5.1 Conventions and Notations for IPS Proofs. An IPS proof over a specific field or ring $\mathbb{F}$ is sometimes denoted $\text{IPS}_{\mathbb{F}}$. For two algebraic circuits F, G, we define the *size of the equation* $F = G$ to be the total circuit size of F and G, namely, $|F| + |G|$.

Let $\overline{\mathcal{F}}$ denote a set of polynomial equations $\{f_i(\overline{x}) = 0\}_{i=1}^m$, and let $C(\overline{x}, \overline{y}, \overline{z}) \in \mathbb{F}[\overline{x}, \overline{y}, \overline{z}]$ be an IPS proof of $f(\overline{x})$ from $\overline{\mathcal{F}}$ as in Definition 2. Then we write $C(\overline{x}, \overline{\mathcal{F}}, \overline{x}^2 - \overline{x})$ to denote the circuit C in which y_i is substituted by $f_i(\overline{x})$ and z_i is substituted by the boolean axiom $x_i^2 - x_i$. By a slight abuse of notation we also call $C(\overline{x}, \overline{\mathcal{F}}, \overline{x}^2 - \overline{x}) = f(\overline{x})$ an IPS proof of $f(\overline{x})$ from $\overline{\mathcal{F}}$ and $\overline{x}^2 - \overline{x}$ (that is, displaying $C(\overline{x}, \overline{y}, \overline{z})$ *after* the substitution of the placeholder variables $\overline{y}, \overline{z}$ by the axioms in $\overline{\mathcal{F}}$ and $\overline{x}^2 - \overline{x}$, respectively).

For two polynomials $f(\overline{x}), g(\overline{x})$, *an IPS proof of* $f(\overline{x}) = g(\overline{x})$ from the assumptions $\overline{\mathcal{F}}$ is an IPS proof of $f(\overline{x}) - g(\overline{x}) = 0$ (note that in case $f(\overline{x})$ and $g(\overline{x})$ are identical as polynomials this is trivial to prove).

We denote by $C : \overline{\mathcal{F}} \vdash^s_{\text{IPS}} p = 0$ (resp. $C : \overline{\mathcal{F}} \vdash^s_{\text{IPS}} p = g$) the fact that $p = 0$ (resp. $p = g$) has an IPS proof $C(\overline{x}, \overline{y}, \overline{z})$ of size s from assumptions $\overline{\mathcal{F}}$. We may also suppress "= 0" and write simply $C : \overline{\mathcal{F}} \vdash^s_{\text{IPS}} p$ for $C : \overline{\mathcal{F}} \vdash^s_{\text{IPS}} p = 0$. Whenever we are only interested in claiming the existence of an IPS proof of size s of $p = 0$ from $\overline{\mathcal{F}}$ we suppress the C from the notation. Similarly, we can suppress the size parameter s from the notation. If F is a circuit computing a polynomial $\hat{F} \in \mathbb{F}[\overline{x}]$, then we can talk about *an IPS proof* C *of* F *from assumptions* $\overline{\mathcal{F}}$, in symbols $C : \overline{\mathcal{F}} \vdash_{\text{IPS}} F$, meaning an IPS proof of $\hat{F}$. Accordingly, for two circuits F, F' such that $\hat{F} = \hat{F}'$, we may speak about an *an IPS proof* C *of* F *from assumptions* $\overline{\mathcal{F}}$ to refer to an IPS proof of F' from assumptions $\overline{\mathcal{F}}$.

2.6 Semi-algebraic Proofs

The *Positivstellensatz* proof system, as defined by Grigoriev and Vorobojov [24], is a refutation system for establishing the unsatisfiability over the reals $\mathbb{R}$ of a system consisting of both polynomial equations $\overline{\mathcal{F}} = \{f_i(\overline{x}) = 0\}_{i\in I}$ and polynomial inequalities $\overline{\mathcal{H}} = \{h_j(\overline{x}) \geq 0\}_{j\in J}$, respectively. It is based on a restricted version of the Positivstellensatz theorem [32, 48]. In order to formulate it, we need to define the notion of a cone, as in [24], which serves as a non-negative closure of a set of polynomials, or informally the notion of a "positive ideal". Usually the cone is defined as the set closed under non-negative linear combinations of polynomials (cf. [6]), but following [24] we are going to use a more general formulation which is sometimes called *the sos cone.*

Definition 4 (cone)**.** *Let* $\overline{\mathcal{H}} \subseteq R[\overline{x}]$ *be a set of polynomials over an ordered ring* R*. Then* the cone of $\overline{\mathcal{H}}$*, denoted* $\text{cone}(\overline{\mathcal{H}})$*, is defined to be the smallest set* $S \subseteq R[\overline{x}]$ *such that:*

(1) $\overline{\mathcal{H}} \subseteq S$*;*
(2) for any polynomial $s \in R[\overline{x}]$*,* $s^2 \in S$*;*
(3) for any positive constant $c > 0$*,* $c \in S$*;*
(4) if $f, g \in S$*, then both* $f + g \in S$ *and* $f \cdot g \in S$*.*

Note that we have formulated the cone for any ordered ring (item 3 would be superfluous for reals). This is because we are going to use this notion in the context of $\mathbb{Z}$ and $\mathbb{Q}$ (although the Positivstellensatz theorem does not hold for these rings, it is still possible to use Positivstellensatz refutations in the presence of the boolean axioms, namely as a refutation system for instances unsatisfiable over 0-1 value).

Note also that every sum of squares (that is, every sum of squared polynomials $\sum_i s_i^2$) is contained in every cone. Specifically, $\text{cone}(\emptyset)$ contains every sum of squares.

Similar to the way the Nullstellensatz proof system [4] establishes the unsatisfiability of sets of polynomial equations based on the Hilbert's Nullstellensatz theorem [26] from algebraic geometry, the Positivstellensatz proof system is based on the Positivstellensatz theorem from semi-algebraic geometry:

Theorem 2.3 (Positivstellensatz theorem [32, 48], restricted version)*. Let* $\overline{\mathcal{F}} := \{f_i(\overline{x}) = 0\}_{i\in I}$ *be a set of polynomial*

[2] That is, $C(\overline{x}, \overline{0}, \overline{0})$ computes the zero polynomial and $C(\overline{x}, f_1(\overline{x}), \ldots, f_m(\overline{x}), x_1^2 - x_1, \ldots, x_n^2 - x_n)$ computes the polynomial $p(\overline{x})$.

equations and let $\overline{\mathcal{H}} := \{h_j(\overline{x}) \geq 0\}_{j\in J}$ be a set of polynomial inequalities, where all polynomials are from $\mathbb{R}[x_1,\ldots,x_n]$. There exists a pair of polynomials $f \in \langle\{f_i(\overline{x})\}_{i\in I}\rangle$ and $h \in \text{cone}(\{h_j(\overline{x})\}_{j\in J})$ such that $f+h=-1$ if and only if there is no assignment that satisfies both $\overline{\mathcal{F}}$ and $\overline{\mathcal{H}}$.

The Positivstellensatz proof system is now natural to define. We shall distinguish between the *real* Positivstellensatz in which variables are meant to range over the reals and *boolean* Positivstellensatz in which variables range over $\{0,1\}$.

Definition 5 (real Positivstellensatz proof system (real PS) [24]). *Let $\overline{\mathcal{F}} := \{f_i(\overline{x}) = 0\}_{i\in I}$ be a set of polynomial equations and let $\overline{\mathcal{H}} := \{h_j(\overline{x}) \geq 0\}_{j\in J}$ be a set of polynomial inequalities, where all polynomials are from $\mathbb{R}[x_1,\ldots,x_n]$. Assume that $\overline{\mathcal{F}}, \overline{\mathcal{H}}$ have no common real solutions. A* Positivstellensatz refutation *of $\overline{\mathcal{F}}, \overline{\mathcal{H}}$ is a collection of polynomials $\{p_i\}_{i\in I}$ and $\{s_{i,\zeta}\}_{i,\zeta}$ (for $i\in\mathbb{N}$, $\zeta\subseteq J$ and $I_\zeta\subseteq\mathbb{N}$) in $\mathbb{R}[x_1,\ldots,x_n]$ such that the following formal polynomial identity holds:*

$$\sum\nolimits_{i\in I} p_i \cdot f_i + \sum\nolimits_{\zeta\subseteq J}\left(\prod\nolimits_{j\in\zeta} h_j \cdot \left(\sum\nolimits_{i\in I_\zeta} s_{i,\zeta}^2\right)\right) = -1\,. \qquad (1)$$

*The **monomial size** of a Positivstellensatz refutation is the combined total number of monomials in $\{p_i\}_{i\in I}$ and $\sum_{i\in I_\zeta} s_{i,\zeta}^2$, for all $\zeta\subseteq J$, that is, $\sum_{i\in I}|p_i|_{\#\text{monomials}} + \sum_{\zeta\subseteq J}\left|\sum_{i\in I_\zeta} s_{i,\zeta}^2\right|_{\#\text{monomials}}$.*

Note that Grigoriev, Hirsch and Pasechnik [23] defined the size of Positivstellensatz proofs slightly differently: they included in the size of proofs both the number of monomials and the size of the coefficients of monomials written in binary (this does not matter for their lower bounds). This is more natural when considering Positivstellensatz as a propositional proof system (which is meant to be polynomially verifiable).

In order to use Positivstellensatz as a refutation system for collections of equations $\overline{\mathcal{F}}$ and inequalities $\overline{\mathcal{H}}$ that are unsatisfiable over 0-1 assignments, we need to include simple so-called boolean axioms. This is done in slightly different ways in different works (see for example [2, 23]). One way to do this, which is the way we follow, is the following:

Definition 6 ((boolean) Positivstellensatz proof system (boolean PS)). *A **boolean Positivstellensatz proof** from a set of polynomial equations $\overline{\mathcal{F}}$, and polynomial inequalities $\overline{\mathcal{H}}$, is an algebraic Positivstellensatz proof in which the following **boolean axioms** are part of the axioms: the polynomial equations $x_i^2 - x_i = 0$ (for all $i\in[n]$) are included in $\overline{\mathcal{F}}$, and the polynomial inequalities $x_i \geq 0,\ 1-x_i\geq 0$ (for all $i\in[n]$) are included in $\overline{\mathcal{H}}$.*

In this way, $\overline{\mathcal{F}}, \overline{\mathcal{H}}$ have no common 0-1 solutions iff there exists a boolean Positivstellensatz refutation of $\overline{\mathcal{F}}, \overline{\mathcal{H}}$. Eventually, to define the boolean Positivstellensatz as a propositional proof system for the unsatisfiable CNF formula we consider CNF formulas to be encoded as polynomial equalities according to Definition 3. This version is sometimes called ***propositional Positivstellensatz***. *As a default when referring to Positivstellensatz we mean the boolean Positivstellensatz version.* Nevertheless, we shall consider the propositional Positivstellensatz as a refutation system for unsatisfiable equations and inequalities $\overline{\mathcal{F}}, \overline{\mathcal{H}}$ over 0-1 assignments, *not* restricting it to merely CNF formulas.

In recent years, starting mainly with the work of Barak, Brandao, Harrow, Kelner, Steurer and Zhou [3], a special case of the Positivstellensatz proof system has gained much interest due to its application in complexity and algorithms (cf. [36]). This is the ***sum-of-squares*** proof system (**SoS**), which is defined as follows:

Definition 7 (sum-of-squares proof system). *A **sum-of-squares proof** (SoS for short) is a Positivstellensatz proof in which in (1) in Definition 5 we restrict the index sets $\zeta\subseteq J$ to be* singletons, *namely $|\zeta|=1$, hence, disallowing arbitrary products of inequalities within themselves. The real, boolean and propositional versions of* SoS *are defined similar to Positivstellensatz.*

For most interesting cases SoS is also complete (and sound) by a result of Putinar [42].

2.6.1 Dynamic Positivstellensatz.

Here we follow Grigoriev, Hirsch and Pasechnik [23] to define what is, to the best of our knowledge, the most general propositional Positivstellensatz- (or SoS-) based semi-algebraic proof system defined to date. It can be viewed as a generalization of (dynamic) Lovasz-Schrijver proof systems [34, 35] that have been put in the context of propositional proof complexity by Pudlák [41], and constitutes essentially a dynamic version of propositional Positivstellensatz (the proof size is measured by the total number of monomials with repetitions appearing in the proof).

The translation of propositional formulas here is different from the algebraic translation (Definition 3). For higher degree proof systems, Definition 3 and the definition that follows can be reduced to one another (within the proof system, as long as both translations can be written down efficiently); however, we provide Definition 8 for the sake of consistency with earlier work.

Definition 8 (semi-algebraic translation of CNF formulas). *Given a CNF formula in the variables $\overline{x}$, every clause $\bigvee_{i\in P} x_i \vee \bigvee_{j\in N}\neg x_j$ is translated into $\sum_{i\in P} x_i + \sum_{j\in N}(1-x_j) \geq 1$.*

Notice that in this way a 0-1 assignment to a CNF formula is satisfying iff the assignment satisfies all the inequalities in the semi-algebraic translation of the CNF formula.

Definition 9 ($\text{LS}_{*,+}^{\infty}$ [23]; "dynamic Positivstellensatz"). *We work over $\mathbb{R}$. Consider a boolean formula in conjunctive normal form and translate it into inequalities as in Definition 8. The axioms are taken to be these inequalities as well as $x\geq 0$, $1-x\geq 0$, $x^2-x\geq 0$, $x-x^2\geq 0$ for each variable x, and $h^2\geq 0$ for any polynomial h of degree at most d. An* $\text{LS}_{*,+}^{d}$ proof *of the original CNF formula is a derivation of $-1\geq 0$ from these axioms using the following rules:*

$$\frac{f\geq 0,\quad g\geq 0}{f+g\geq 0} \qquad \frac{f\geq 0}{\alpha f\geq 0}\ \text{(for } \alpha \text{ a nonnegative integer)} \qquad \frac{f\geq 0,\quad g\geq 0}{f\cdot g\geq 0},$$

where these rules mean that from the inequalitie(s) above the line we can derive the inequality below the line, assuming the latter is of degree at most d. In particular, $\text{LS}_{*,+}^{\infty}$ is such a proof without the restriction on the degree. Polynomials are written as sums of monomials (and not as circuits or formulas), so the verification of such proofs is doable in deterministic polynomial-time.

The proof of the following simulation follows by definition and we omit the details:

Proposition 2.4. $\mathrm{LS}^{\infty}_{*,+}$ *simulates boolean Positivstellensatz.*

2.7 Overview of Results and Techniques

We consider the following subset-sum instance written as an unsatisfiable linear equation with large coefficients, expressing the fact that natural numbers written in binary cannot be negative:

Definition 10 (Binary Value Principle BVP_n). *The* binary value principle *over the variables* $x_1, \ldots, x_n$, BVP_n *for short, is the following unsatisfiable (over* $\{0,1\}$ *assignments) linear equation:*

$$x_1 + 2x_2 + 4x_3 + \cdots + 2^{n-1}x_n = -1\,.$$

At times we use a more general principle denoted $\mathrm{BVP}_{n,M}$, which we call the *generalized binary value principle*: $x_1 + 2x_2 + 4x_3 + \cdots + 2^{n-1}x_n = -M,$ for a positive integer M.

2.7.1 Lower Bounds. We prove two kinds of conditional super-polynomial lower bounds against IPS refutations. The first is over $\mathbb{Q}$ and $\mathbb{Z}$ and the second is over the field $\mathbb{Q}(y)$ of rational functions of univariate polynomials in the indeterminate y denoted $\mathbb{Q}[y]$ (see full version [1]). We start with the first lower bound.

Theorem (Thm. 3.3). *Under the Shub and Smale hypothesis, there are no* $\mathrm{poly}(n)$*-size constant-free (boolean) IPS refutations of the binary value principle* BVP_n *over* $\mathbb{Q}$.

This result can be viewed as pushing forward to full IPS the proof method initiated by Forbes et al. [18] wherein proof complexity lower bound questions are reduced to algebraic circuit size lower bound questions: an IPS proof written as a circuit from a class C is obtained by showing that there are no small C-circuits computing certain polynomials. Here, by "full IPS" we simply mean that instead of using circuit lower bounds to obtain lower bounds against *subsystems* of IPS, we use a circuit lower bound, alas conditional, to obtain a lower bound against (general) IPS.

We stress that *this approach can only lead to conditional lower bounds for full (unrestricted) IPS*, as long as we do not have (explicit) super-polynomial lower bounds against general algebraic circuits, namely as long as we do not prove VP ≠ VNP.[3]

Proof sketch of Thm. 3.3: First, we show in Cor. 3.2 that it is enough to consider IPS refutations over $\mathbb{Z}$ instead of $\mathbb{Q}$. An IPS refutation over $\mathbb{Z}$ is a proof of a nonzero integer M instead of -1. Let $S_n := \sum_{i=1}^{n} 2^{i-1}x_i$ so that BVP is $S_n + 1 = 0$, and assume that the IPS refutation of BVP is written as follows (this can be assumed without loss of generality by a result of [18]):

$$Q(\overline{x}) \cdot (S_n + 1) + \sum_{i=1}^{n} H(\overline{x}) \cdot (x_i^2 - x_i) = M\,. \tag{2}$$

Since the IPS refutation is over $\mathbb{Z}$ we know in particular that $Q(\overline{x})$ is an integer polynomial. Let us consider now only $\{0,1\}$ assignments to eq. 2. Since under $\{0,1\}$ assignments the boolean axioms $x_i^2 - x_i$ vanish we get from eq. 2:

$$Q(\overline{x}) \cdot (S_n + 1) = M\,. \tag{3}$$

Observe that the image of $S_n + 1$ under boolean assignments is the set of all possible natural numbers between 1 to 2^n. In other words, for every number $b \in [2^n]$, there exists an assignment $\overline{\alpha} \in \{0,1\}^n$, such that $(S_n + 1)(\overline{\alpha}) = b$. Since $Q(\overline{x})$ is an integer polynomial, it evaluates to an integer under every $\{0,1\}$ assignment. Therefore, by eq. 3 M is a product of every natural number between 1 to 2^n. This already brings us close to the conditional lower bound: we assume contra-positively that there is a polynomial-size constant-free circuit that computes $Q(\overline{x})$, which implies that there exists a polynomial-size constant-free and variable-free circuit that computes M (because fixing any boolean assignment to the variables we get such a circuit over $\mathbb{Z}$ computing M). We then show that if there exists a $\mathrm{poly}(n)$-size constant-free circuit for $M \in \mathbb{Z}$, such that M is divisible by every number in $[2^n]$, then there exists a $\mathrm{poly}(n)$-size circuit that computes $(2^n)!$.

Consider the $\mathrm{poly}(n)$-size circuit for M^{2^n} that is obtained by n repeated squaring of M. Since M is divided by every natural number in $[2^n]$ it is in particular divisible by every *prime* number in $[2^n]$. It is possible to show (Claim 3.5) that the power of every prime number in the prime factorisation of $(2^n)!$ is at most 2^n (and that every such prime is $< 2^n$), from which we can conclude that M^{2^n} is an integer product of $(2^n)!$. We thus obtain a constant-free $\mathrm{poly}(n)$-size circuit for a nonzero integer product of $(2^n)!$. From this it is easy to show that for every m with $2^{n-1} \le m \le 2^n$ there is a $\mathrm{poly}(n)$-size constant-free circuit computing a nonzero integer product of $m!$, hence there is a sequence $(c_m \cdot m!)_{m=1}^{\infty}$ with c_m a nonzero integer for all m that admits a $\log^{O(1)} m$-size family of constant-free circuits, in contrast to the Shub-Smale hypothesis. □

Rational field lower bounds. We can consider IPS operating over the field of rational functions in the (new) indeterminate y, denoted $\mathbb{Q}(y)$. This allows us to formulate a very interesting version of the binary value principle. Roughly speaking, this version expresses the fact that the BVP is "almost always" unsatisfiable. More precisely, consider the linear equation $\sum_{i=1}^{n} a_i x_i = y$, for integer a_i's, and y the new indeterminate. This equation is unsatisfiable for most y's, when y is substituted by an element from $\mathbb{Q}$. We show that once we have an IPS refutation over $\mathbb{Q}(y)$ of this equation we can substitute y by any rational number but a finite number of rational numbers and get a valid IPS refutation over $\mathbb{Q}$ of the original BVP. Thus *an IPS refutation over* $\mathbb{Q}(y)$ *of* $\sum_{i=1}^{n} a_i x_i = y$ *can be viewed as a single refutation for all but finitely many values of* $y \in \mathbb{Q}$.

We show that while for polynomially bounded coefficients a_i there are small $\mathbb{Q}(y)$-IPS refutations of $\sum_{i=1}^{n} a_i x_i = y$, for $\sum_{i=1}^{n} 2^{i-1}x_i = y$, there are no small refutations, assuming the τ-conjecture:

Theorem 2.5. *Suppose a system of polynomial equations* $F_0(\overline{x}) = F_1(\overline{x}) = F_2(\overline{x}) = \cdots = F_n(\overline{x}) = 0$, $F_i \in \mathbb{Q}(y)[x_1, \ldots, x_n]$, *where* $F_0(\overline{x}) = y + \sum_{i=1}^{i=n} 2^{i-1}x_i$ *and* $F_i(\overline{x}) = x_i^2 - x_i$, *has an IPS-LIN*$_{\mathbb{Q}(y)}$ *certificate* $H_0(\overline{x}), \ldots, H_n(\overline{x})$, *where each* $H_i(\overline{x})$ *can be computed by a* $\mathbb{Q}(y)[x_1, \ldots, x_n]$*-algebraic circuit of size* $\mathrm{poly}(n)$ *. Then, the* τ*-conjecture is false.*

Roughly speaking, the lower bound proof extracts denominators from the refutation and obtains a small circuit that has all n-bit

[3] Though, it should be mentioned that in proof complexity even non-explicit lower bounds are not known, and will constitute a breakthrough in the field; hence moving from non-explicit (and thus *known*) circuit lower bounds to (possibly also non-explicit) proof complexity lower bounds cannot be ruled out entirely.

nonnegative integers as its roots and thus cannot exist under the τ-conjecture (see [1] for more details).

2.7.2 *Algebraic versus Semi-Algebraic Proofs.*

We exhibit the importance of the binary value principle by showing that it captures in a manner made precise the strength of semi-algebraic reasoning in the regime of strong (to very strong) proof systems, and formally those systems that can efficiently reason about bit arithmetic. Note that already Frege system can reason about bit arithmetic (see [19] following [10]); however, this alone is not sufficient to simulate semi-algebraic systems: one needs also to be able to prove the BVP. Specifically, we show that short refutations of the binary value principle would bridge the gap between very strong algebraic reasoning captured by the ideal proof system and its semi-algebraic analogue that we introduce in this work, which we call the Cone Proof System (CPS for short).

Whereas IPS is devised to capture derivations in the *ideal* of initial given polynomials, CPS is defined so to exhibit derivations in the *cone* (Definition 4) of these polynomials. The cone proof system establishes that a collection of polynomial equations $\overline{\mathcal{F}} := \{f_i = 0\}_i$ and polynomial inequalities $\overline{\mathcal{H}} := \{h_i \geq 0\}_i$ are unsatisfiable over 0-1 assignments (or over real-valued assignments, when desired). In the spirit of IPS [25] we define a refutation in CPS as a *single* algebraic circuit. This circuit computes a polynomial that results from positive-preserving operations such as addition and product applied between the inequalities $\overline{\mathcal{H}}$ and themselves, as well as the use of nonnegative scalars and arbitrary squared polynomials. In order to simulate in CPS the free use of equations from $\overline{\mathcal{F}}$ we incorporate in the set of inequalities $\overline{\mathcal{H}}$ the inequalities $f_i \geq 0$ and $-f_i \geq 0$ for each $f_i = 0$ in $\overline{\mathcal{F}}$. We show that this enables one to add freely products of the polynomial f_i in CPS proofs, namely working in the ideal of $\overline{\mathcal{F}}$ in addition to working in the cone of $\overline{\mathcal{H}}$ (even *without* the use of Boolean axioms—an observation that may be interesting by itself; see Prop. 4.1).

We first formalize the concept of a cone as an algebraic circuit. Let C be a circuit and v be a node in C. We call v a ***squaring gate*** if v is a product gate of which two incoming edges are emanating from the *same* node z, that is $v = z^2$.

Definition 11 ($\overline{y}$-conic circuit)**.** *Let R be an ordered ring. We say that an algebraic circuit C computing a polynomial over $R[\overline{x}, \overline{y}]$ is a* ***conic circuit with respect to*** *$\overline{y}$, or $\overline{y}$**-conic** for short, if for every negative constant or variable $x_i \in \overline{x}$, that appears as a leaf u in C, the following holds: every path from u to the output gate of C contains a squaring gate.*

Informally, a $\overline{y}$-conic circuit is a circuit in which we assume that the $\overline{y}$-variables are nonnegative, and any other input that may be negative (that is, a negative constant or an $\overline{x}$-variable) must be part of a squared sub-circuit. Here are examples of $\overline{y}$-conic circuits (over $\mathbb{Z}$): y_1, $y_1 \cdot y_2$, $3+2y_1$, $(-3)^2$, x_1^2, $(3\cdot -x_1+1)^2$, $(x_1y_2+y_1)^2$, $y_1+\cdots+y_n$. On the other hand, -1, x_1, $x_1 \cdot y_2$, $-1 \cdot y_1 + 4$ are examples of non $\overline{y}$-conic circuits.

Note that if the $\overline{y}$-variables of a $\overline{y}$-conic circuit are assumed to take on non-negative values, then a $\overline{y}$-conic circuit computes only non-negative values. It is evident that $\overline{y}$-conic circuits can compute all and only polynomials that are in the cone of the $\overline{y}$ variables. In other words, if $\overline{y}$ are the variables $y_1, \ldots, y_m$, then there exists a $\overline{y}$-conic circuit $C(\overline{x}, \overline{y})$ that computes the polynomial $p(\overline{x}, \overline{y})$ iff $p(\overline{x}, \overline{y}) \in \mathrm{cone}(y_1, \ldots, y_m) \subseteq R[\overline{x}, \overline{y}]$. Similarly, if $\overline{f}(\overline{x})$ is a sequence of polynomials $f_1(\overline{x}), \ldots, f_m(\overline{x})$, then there exists a $\overline{y}$-conic circuit $C(\overline{x}, \overline{y})$ such that $C(\overline{x}, \overline{f}(\overline{x})) = p(\overline{x})$ iff $p(\overline{x})$ computes a polynomial in $\mathrm{cone}(\overline{f}(\overline{x})) \subseteq R[\overline{x}]$.

CPS is defined roughly in the same way as IPS only that instead of circuits we use conic circuits:

Definition 12 ((boolean) Cone Proof System (CPS))**.** *Consider a collection of polynomial equations $\overline{\mathcal{F}} := \{f_i(\overline{x}) = 0\}_{i=1}^m$, and a collection of polynomial inequalities $\overline{\mathcal{H}} := \{h_i(\overline{x}) \geq 0\}_{i=1}^\ell$, where all polynomials are from $\mathbb{R}[x_1, \ldots, x_n]$. Assume that the following* ***boolean axioms*** *are included in the assumptions: $\overline{\mathcal{F}}$ includes $x_i^2 - x_i = 0$, and $\overline{\mathcal{H}}$ includes the inequalities $x_i \geq 0$ and $1 - x_i \geq 0$, for every variable $x_i \in \overline{x}$. Suppose further that $\overline{\mathcal{H}}$ includes (among possibly other inequalities) the two inequalities $f_i(\overline{x}) \geq 0$ and $-f_i(\overline{x}) \geq 0$ for every equation $f_i(\overline{x}) = 0$ in $\overline{\mathcal{F}}$ (including the equations $x_i^2 - x_i = 0$). A* ***CPS proof of*** *$p(\overline{x})$* ***from*** *$\overline{\mathcal{F}}$* ***and*** *$\overline{\mathcal{H}}$, showing that $\overline{\mathcal{F}}, \overline{\mathcal{H}}$ semantically imply the polynomial inequality $p(\overline{x}) \geq 0$ over* 0-1 *assignments, is an algebraic circuit $C(\overline{x}, \overline{y})$ computing a polynomial in $\mathbb{R}[\overline{x}, y_1, \ldots, y_\ell]$, such that:*[4]

(1) $C(\overline{x}, \overline{y})$ is a $\overline{y}$-conic circuit; and

(2) $C(\overline{x}, \overline{\mathcal{H}}) = p(\overline{x})$,

where equality 2 above is a formal polynomial identity (that is, $C(\overline{x}, \overline{\mathcal{H}})$ computes the polynomial $p(\overline{x})$) in which the left hand side means that we substitute $h_i(\overline{x})$ for y_i, for all $i = 0, \ldots, \ell$. The ***size*** *of a CPS proof is the size of the circuit C. The variables $\overline{y}$ are the* placeholder variables *since they are used as a placeholder for the axioms. A CPS proof of -1 from $\overline{\mathcal{F}}, \overline{\mathcal{H}}$ is called a* ***CPS refutation of $\overline{\mathcal{F}}, \overline{\mathcal{H}}$****.*

To refute CNF formulas in CPS we use the algebraic translation of CNFs (Definition 3) into a set of polynomial equalities (we can equally express CNFs as inequalities). The real version of CPS, called ***real CPS***, is defined similar to CPS only without the boolean axioms.

Remarks about CPS.

(1) CPS should be thought of as a way to derive valid polynomial inequalities from a set of polynomial equations and inequalities from $\mathbb{R}[\overline{x}]$. Loosely speaking, it is a circuit representation of the Positivstellensatz proof system (Definition 5), though in CPS the assumptions $\overline{\mathcal{F}}, \overline{\mathcal{H}}$ (more precisely, placeholder variables of which) may have powers greater than one. That is, whereas 1 is *multilinear* in the h_i variables, CPS is not.

(2) We add the boolean axioms $x_i^2 - x_i \geq 0$, $x_i - x_i^2 \geq 0$, $x_i \geq 0$ and $1 - x_i \geq 0$ to $\overline{\mathcal{H}}$ as a default. Hence, the system can refute any set of inequalities (and equalities) that is unsatisfiable over 0-1 assignments.

(3) Formally, CPS proves only consequences from an initial set of inequalities $\overline{\mathcal{H}}$ and not equalities $\overline{\mathcal{F}}$. However, we are not losing any power doing this. First, observe that:

[4] Note that formally we do not make use of the assumptions $\overline{\mathcal{F}}$ in CPS, as we assume always that the inequalities that correspond to the equalities in $\overline{\mathcal{F}}$ are present in $\overline{\mathcal{H}}$. Thus, the indication of $\overline{\mathcal{F}}$ is done merely to maintain clarity and distinguish (semantically) between two kinds of assumptions: equalities and inequalities.

An assignment satisfies $\overline{\mathcal{F}}, \overline{\mathcal{H}}$ iff it satisfies $\overline{\mathcal{H}}$ (in the case of boolean CPS an assignment that satisfies either $\overline{\mathcal{F}}$ or $\overline{\mathcal{H}}$ must be a 0-1 assignment).

Second, we encode equalities $f_i(\overline{x}) = 0 \in \overline{\mathcal{F}}$ using the two inequalities $f_i(\overline{x}) \geq 0$ and $-f_i(\overline{x}) \geq 0$ in $\overline{\mathcal{H}}$. As shown in Thm. 4.2 *this way we can derive any polynomial in the* ideal *of* $\overline{\mathcal{F}}$*, and not merely in the cone of* $\overline{\mathcal{F}}$*,* as is required for equations (and similar to the definition of SoS), with at most a polynomial increase in size (when compared to IPS).

To derive polynomials in the ideal of $\overline{\mathcal{F}}$ we need to be able to multiply f_i and $-f_i$ (from $\overline{\mathcal{H}}$) by *any* (positive) polynomial in the $\overline{x}$ variables. There are two ways to achieve this in boolean CPS: the first, is to use the boolean axiom $x_i \geq 0$ in $\overline{\mathcal{H}}$. This allows to product f_i and $-f_i$ by any polynomial in the $\overline{x}$-variables. The second way, the one we use in Prop. 4.1 to show that CPS simulates IPS in Thm. 4.2, is different and does not necessitate the addition of the axiom $x_i \geq 0$ to $\overline{\mathcal{H}}$ (or any other boolean axioms). Since the second way does not use the boolean axiom $x_i \geq 0$ in $\overline{\mathcal{H}}$ we can use it in real CPS, hence allowing the derivation of polynomials in the ideal of $\overline{\mathcal{F}}$ within real CPS.

In contrast to IPS where a short refutation for BVP_n would imply strong computational consequences, the binary value principle is trivially refutable in CPS (as well as in SoS):

Proposition 2.6. *CPS admits a linear size refutation of the binary value principle* BVP_n*.*

Proof: To simplify notation we put $S := \sum_{i=1}^{n} 2^{i-1} \cdot x_i + 1$. Let $\overline{\mathcal{F}} := \left\{S = 0, x_1^2 - x_1 = 0, \ldots, x_n^2 - x_n = 0\right\}$. Then by the definition of CPS $\overline{\mathcal{H}}$ contains the following correspondent $4n + 2$ axioms ($4n$ boolean axioms, and two axioms for the single non-boolean axiom in $\overline{\mathcal{F}}$):

$$\overline{\mathcal{H}} := \{x_1 \geq 0, \ldots,\ x_n \geq 0,\ -S \geq 0,\ S \geq 0,\ x_1^2 - x_1 \geq 0, \ldots, x_n^2 - x_n \geq 0,\ -(x_1^2 - x_1) \geq 0, \ldots,\ -(x_n^2 - x_n) \geq 0,\ 1 - x_1 \geq 0, \ldots,\ 1 - x_n \geq 0\Big\}\,.$$

Therefore, the CPS refutation of the binary value principle is defined as the following $\overline{y}$*-conic* circuit:

$$C(\overline{x}, \overline{y}) := \left(\sum_{i=1}^{n} 2^{i-1} \cdot y_i\right) + y_{n+1}, \tag{4}$$

where the placeholder variables $y_1, y_2, \ldots, y_{4n+2}$ correspond to the axioms in $\overline{\mathcal{H}}$ in the order they appear above. Observe indeed that $C(\overline{x}, \overline{\mathcal{H}}) = C(\overline{x}, x_1, \ldots, x_n, -S, \ldots) = \left(\sum_{i=1}^{n} 2^{i-1} \cdot x_i\right) + (-S) = -1$. □

Observing the CPS refutation in 4 we see that it is in fact already an SoS refutation:

Corollary 2.7. SoS *admits a linear monomial size refutation of the binary value principle* BVP_n*.*

The following is immediate from the definitions (see full version [1]).

Corollary 2.8. *Boolean CPS simulates* SoS *and Positivstellensatz for inputs that include the boolean axioms.*

We show that IPS and CPS simulate each other if there exist small IPS refutations of the binary value principle. This provides a characterisation of semi-algebraic reasoning in terms of the binary value principle. In what follows, $\mathrm{IPS}_{\mathbb{Z}}^{\star}$ and $\mathrm{CPS}_{\mathbb{Z}}^{\star}$ stand for *boolean* versions of IPS and CPS, where both are proof systems for refuting unsatisfiable sets of polynomial equalities (not necessarily CNFs) and where the '$\star$' superscript means that possible values that are computed along the IPS or CPS proofs (as circuits) are not super-exponential (when the input variables range over $\{0,1\}$), namely, that the bit-size of these values are polynomial in the proof size.

Corollary 2.9 (BVP characterizes the strength of boolean CPS)**.**

(1) Constant-free $\mathrm{IPS}_{\mathbb{Z}}^{\star}$ *is polynomially equivalent to constant-free* $\mathrm{CPS}_{\mathbb{Z}}^{\star}$ *iff constant-free* $\mathrm{IPS}_{\mathbb{Z}}^{\star}$ *admits* $\mathrm{poly}(t)$*-size refutations of* BVP_t*.*

(2) Constant-free $\mathrm{IPS}_{\mathbb{Q}}^{\star}$ *is polynomially equivalent to constant-free* $\mathrm{CPS}_{\mathbb{Q}}^{\star}$ *iff for every positive integer M constant-free* $\mathrm{IPS}_{\mathbb{Q}}^{\star}$ *admits* $\mathrm{poly}(t, \tau(M))$*-size refutations of* $\mathrm{BVP}_{t,M}$*.*

Proof idea. In Sect. 4 we prove that CPS simulates IPS. Here we sketch the proof of part 1 ($\Leftarrow$). For full proofs see the full version [1].

To show that $\mathrm{IPS}_{\mathbb{Z}}^{\star}$ simulates $\mathrm{CPS}_{\mathbb{Z}}^{\star}$ assuming short refutations of BVP_t we proceed as follows: let $C(\overline{x}, \overline{\mathcal{F}}) = -1$ be the $\mathrm{CPS}_{\mathbb{Z}}^{\star}$ refutation of $\overline{\mathcal{F}}$. Then, as a polynomial identity $C(\overline{x}, \overline{\mathcal{F}}) = -1$ is basically freely provable in $\mathrm{IPS}_{\mathbb{Z}}^{\star}$. We now use the ability of IPS to do efficient bit arithmetic, that we demonstrate formally as follows. Define $\mathrm{VAL}(\overline{w}) = w_1 + 2w_2 + \ldots 2^{n-2} w_{n-1} - 2^{n-1} w_n$ to be the value of an integer number given by the n boolean bits $\overline{w}$ in the two's complement scheme (where w_n is the sign bit). Our *main novel technical contribution here* is the following result connecting the value of a polynomial to its bit vector expressed as a function of the variables:

Lemma 2.10 (informal)**.** *For any circuit f, IPS has a* $\mathrm{poly}(|f|)$*-size proof of*

$$\mathrm{VAL}\left(\mathrm{BIT}_1(f) \cdots \mathrm{BIT}_n(f)\right) = f, \tag{5}$$

where $\mathrm{BIT}_i(f)$ *is the polynomial that computes the ith bit of the number computed by f as a function of the variables $\overline{x}$ to f that range over* $\{0,1\}$ *values.*

Denote $C(\overline{x}, \overline{\mathcal{F}})$ by C for short. By eq. 5 we have $C = \mathrm{VAL}\left(\mathrm{BIT}_1(C) \cdots \mathrm{BIT}_n(C)\right) = -1$. Since C is a conic circuit and thus preserves positive signs we can prove that the sign bit $\mathrm{BIT}_n(C) = 0$. We are thus left with the need to refute that the value of a positive number written in binary $\mathrm{BIT}_1(C) \cdots \mathrm{BIT}_{n-1}(C)$ is non-negative, which is efficiently provable in $\mathrm{IPS}_{\mathbb{Z}}^{\star}$ by assumption. □

The relative strength of proof systems. Figure 1 provides an illustrative picture of the relative strength of algebraic and semi-algebraic proof systems, which gives context to our results. Note that CPS is among the strongest concrete proof systems for boolean tautologies to be formalized to date: it simulates IPS (4.2) which is already very strong (note that *constant-free* IPS simulates Extended Frege [25]). Like IPS it can prove freely polynomial identities, and so it "subsumes" in this sense such identities (accordingly, CPS proofs needs the full power of coRP to be verified). It is unclear whether even ZFC (namely, Zermelo-Fraenkel set theory with the axiom of choice) can simulate CPS (it is not hard to show that this would imply that polynomial identity testing is in NP)[5]. Indeed, we are unaware of any concrete propositional proof system (even those that are merely coRP-verifiable) that can simulate CPS.

Grigoriev [22] showed that algebraic proofs like PC cannot simulate semi-algebraic proofs like SoS because symmetric subset-sum instances such as $x_1 + \cdots + x_n = -1$ require linear degrees (and exponential monomial size) [30], and Forbes et al. [18] extended these lower bounds on symmetric subset-sum instances to stronger algebraic proof systems, namely to subsystems of IPS. Our work (3.3) extends this gap further, showing that even the strongest algebraic proof system known to date IPS cannot fully simulate even a weak proof system like SoS, assuming Shub-Smale hypothesis.

Exponential size lower bounds for semi-algebraic proof systems are known since [23], and such bounds for propositional versions of static Lovasz-Schrijver and constant degree Positivstellensatz systems were proved in [31]. Beame, Pitassi and Segerlind [38] started the study of lower bounds for semantic threshold systems, that include in particular tree-like Lovász-Schrijver systems. This line of research culminated in [21], where strong lower bounds were proved using critical block sensitivity, a notion introduced in [28].

3 CONDITIONAL IPS LOWER BOUNDS

3.1 IPS Lower Bounds under Shub-Smale Hypothesis

Here we provide the full proof of the super-polynomial conditional lower bound on the size of (boolean) IPS refutations of the binary value principle over the rationals based on the Shub-Smale Hypothesis (Sect. 2.3).

The conditional lower bound is first established for constant-free IPS proofs over $\mathbb{Z}$ and then we extract a lower bound over $\mathbb{Q}$ as a corollary using Cor. 3.2 below. Notice that we can consider IPS proofs also over rings, and not only fields. In the case of IPS over $\mathbb{Z}$ we cannot anymore assume that *refutations* are proofs of the polynomial 1, rather we define refutations in IPS over $\mathbb{Z}$ to be proofs of any nonzero constant polynomial (cf. [11, Definition 2.1]):

Definition 13 ($\mathrm{IPS}_{\mathbb{Z}}$). *An* $\mathrm{IPS}_{\mathbb{Z}}$ proof *of* $g(\overline{x}) \in \mathbb{Z}[\overline{x}]$ *from a set of assumptions* $\overline{\mathcal{F}} \subseteq \mathbb{Z}[\overline{x}]$ *is an IPS proof of* $g(\overline{x})$ *from* $\overline{\mathcal{F}}$*, as in Definition 2, where* $\mathbb{F} = \mathbb{Z}$ *and all the constants in the IPS proof are from* $\mathbb{Z}$*. An* $\mathrm{IPS}_{\mathbb{Z}}$ refutation *of* $\overline{\mathcal{F}}$ *is a proof of M, for* $M \in \mathbb{Z} \setminus \{0\}$*. (The definition is similar for the boolean and algebraic IPS versions.)*

We will need to define a constant-free circuit over $\mathbb{Q}$.

Definition 14. *A* constant-free circuit over $\mathbb{Q}$ *is a constant-free algebraic circuit as in Sect. 2.2 that has additionally division gates* $\div$*, where* $u \div v$ *means that the polynomial computed by u is divided by the polynomial computed by v, such that for every division gate* $u \div v$ *in C the circuit v contains* no variables *and computes a nonzero constant. A* constant-free IPS proof over $\mathbb{Q}$ *is an IPS proof written with a constant-free circuit over* $\mathbb{Q}$.

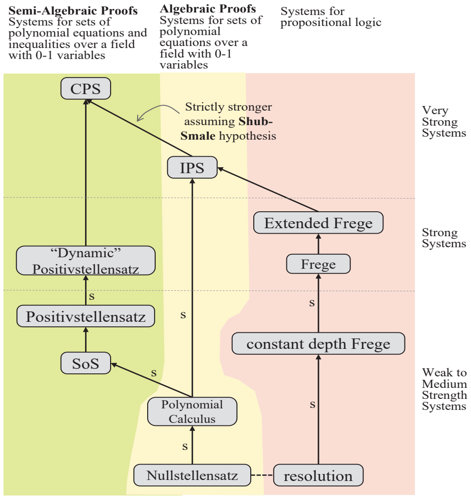
Semi-Algebraic Proofs
Systems for sets of polynomial equations and inequalities over a field with 0-1 variables
Algebraic Proofs
Systems for sets of polynomial equations over a field with 0-1 variables
Systems for propositional logic
CPS
Strictly stronger assuming Shub-Smale hypothesis
IPS
Very Strong Systems
Extended Frege
Strong Systems
"Dynamic" Positivstellensatz
Frege
s
s
s
Positivstellensatz
constant depth Frege
SoS
s
Weak to Medium Strength Systems
Polynomial Calculus
s
s
Nullstellensatz
resolution

Figure 1: Relative strength of propositional proof systems (partial). An arrow $Q \rightarrow P$ means that P simulates Q. While $Q \overset{s}{\rightarrow} P$ means "strictly stronger", i.e., P simulates Q but Q does not simulate P. Dashed line $Q - - - P$ means that Q and P are incomparable: P cannot simulate Q and Q cannot simulate P. The three colored-shaded vertical blocks indicate proof systems for languages of increasing expressiveness (from right to left): systems for propositional logic, for polynomial equations with 0/1 variables (including encodings of propositional logic) and both polynomial equations and inequalities with 0/1 variables. The *informal* qualifications of strength mean roughly the following: weak systems are those we know super-polynomial lower bounds against, and their strength and limitations are quite well understood via feasible interpolation results and random CNFs lower bounds. Medium strength systems are those with some known lower bounds, but their strength is less well understood; e.g., feasible interpolation is not known for them. Strong systems are those with no known lower bounds. Very strong proof systems are those strong systems whose verification is done in coRP, and they can prove freely any polynomial identity (written as an algebraic circuit).

The following proposition is proved by a simple induction on the circuit size, using sufficient products to cancel out the denominators in the circuit over $\mathbb{Q}$, turning it into a circuit over $\mathbb{Z}$.

Proposition 3.1 (from $\mathbb{Q}$-circuit to $\mathbb{Z}$-circuit)**.** *Let C be a size-s constant-free circuit over* $\mathbb{Q}$*. Then there exists a size* $\leq 4s$ *constant-free circuit computing* $M \cdot \widehat{C}$*, for some* $M \in \mathbb{Z} \setminus \{0\}$*, with* $\tau(M) \leq 4s$*.*

Proof: We choose any topological order $g_1, g_2, \ldots, g_i, \ldots, g_{|C|}$ on the gates of the constant-free circuit C over $\mathbb{Q}$ (that is, if g_j has a directed path to g_k in C then $j < k$) and proceed by induction on $|C|$ to eliminate rationals from the circuit (identifying the gate g_i with the sub-circuit of C for which g_i is its root).

Induction statement: Let $g_1, \ldots, g_s$ be the topologically ordered gates of a constant-free circuit C over $\mathbb{Q}$, where $s = |C|$. Then, there exists a division-free constant-free circuit

[5] Indeed, if ZFC simulates CPS, and hence IPS, as a proof system for $\overline{F} \models p = 0$, where $\overline{F}$ is a set of polynomial equations over $\mathbb{Q}$ written as algebraic circuits and $\overline{F} \models p = 0$ means that the $\mathbb{Q}$ polynomial p written as an algebraic circuit is in the ideal generated by $\overline{F}$ (equivalently, that every $\mathbb{Q}$-assignment that nullifies $\overline{F}$ nullifies also p), then PIT is in NP. The proof proceeds as follows: let C be an algebraic circuit over $\mathbb{Q}$. Then there exists a size $|C|$ IPS proof of $\models C = 0$ iff C is identically zero. Hence, there exists a size poly$(|C|)$ ZFC proof of $\models C = 0$ iff C is identically zero. Since ZFC is verifiable in poly-time (we assume here some efficient encoding of circuits over $\mathbb{Q}$), we conclude that any algebraic circuit C over $\mathbb{Q}$ is identically zero iff it has a poly-time verifiable witness (where the witness is the ZFC proof of $\models C = 0$).

consisting of the corresponding topologically ordered gates $g_{11}, \dots, g_{1a_1}, g_{21}, \dots, g_{2a_2}, \dots, g_{s1}, \dots, g_{sa_s}$, such that for *every* $i \le s$:

(1) $a_i \le 4$ and g_{ia_i} computes the polynomial $M_i \cdot g_i$, for some nonzero integer M_i (again, identifying the gate g_{ia_i} with the sub-circuit for which it is a root);
(2) The integer M_i is constructed as a part of the circuit (except for the trivial case $M_i = 1$). More precisely, there exists a division-free constant-free and variable-free (sub-)circuit $g_{j,\ell}$, for $j \le i, \ell \le 4$ that computes M_i. In particular $\tau(M_i) \le 4i$;
(3) Sub-circuits with no division gates remain intact: if g_i is a division-free constant-free circuit then $M_i = 1$ and g_i is a sub-circuit of the new circuit. That is, $g_{i1} = g_i$, $a_i = 1$, and all gates in $g_1, \dots, g_i$ that are part of the sub-circuit g_i in C occur also as gates $g_{j\ell}$ (for some $j \le i, \ell \le 4$) in the new circuit g_{i1}.

Base case: g_i is a variable or a constant in $\{-1, 0, 1\}$. Hence, we put $g_{i1} := g_i$, $a_i = 1$, and $M_i = 1$.

Induction step: In the case of a binary gate $g_i = g_j \circ g_\ell$, for $\circ \in \{\times, +, \div\}$ (where $j, \ell < i$), by induction hypothesis we already have division-free constant-free circuits g_{ja_j} and $g_{\ell a_\ell}$ computing the polynomials $M_j g_j$ and $M_\ell g_\ell$, respectively, for some integers M_j, M_ℓ that are also computed as part of the circuit.

Case 1: g_i is a division gate computing g_j / g_ℓ, where, by definition of circuits over $\mathbb{Q}$, g_ℓ is a division-free constant-free circuit computing a nonzero constant that contains no variables.

By induction hypothesis item 1 we have already constructed the two gates g_{ja_j} and $g_{\ell a_\ell}$, where g_{ja_j} computes the polynomial $M_j g_j$ for some nonzero integer M_j. By item 2, M_j is already computed by one of the gates in the circuit. Finally, since g_ℓ does not have division gates by definition, item 3 means that $g_{\ell a_\ell} = g_\ell$ (and $a_\ell = a_1$ and $M_\ell = 1$), and specifically g_ℓ is a constant-free variable-free circuit.

We put $g_{i2} := g_{ja_j}$ and $g_{i1} := M_j \cdot g_{\ell a_\ell} = M_j \cdot g_\ell$ (that is, g_{i1} is a product gate that connects to the two previously constructed gates that compute the two integers M_j and g_ℓ), $a_i = 2$ and $M_i = M_j g_\ell$ is an integer. Hence, g_{i2} computes the polynomial $g_{ja_j} = M_j g_j = g_\ell \cdot ((M_j g_j)/g_\ell) = g_\ell \cdot M_j \cdot (g_j/g_\ell) = (g_\ell \cdot M_j) \cdot g_i = M_i \cdot g_i$ and $M_i = M_j g_\ell$ is an integer that is computed (as a constant-free variable-free circuit) by the gate g_{i1} as required.

Case 2: $g_i = g_j \cdot g_\ell$. In this case $a_i = 2$ and $M_i = M_j M_\ell$, and we put $g_{i2} := g_{ja_j} \cdot g_{\ell a_\ell}$ and $g_{i1} := M_i \cdot M_j$, where M_i, M_j are two integers that are computed already in the circuit (with a constant-free division-free and variable-free sub-circuit).

Case 3: $g_i = g_j + g_\ell$. In this case $a_i = 4$, $M_i = M_j M_\ell$, and we put $g_{i4} := M_\ell \cdot g_{ja_j} + M_j \cdot g_{\ell a_\ell}$, namely, we add three gates g_{i2}, g_{i3}, g_{i4} (two products, both of which connects to previous gates, and one addition to add these two products). Finally, we put $g_{i1} := M_i \cdot M_j$, where M_i, M_j are two integers that are computed already in the circuit (with a constant-free division-free and variable-free sub-circuit). □

An immediate corollary of Prop. 3.1 is:

Corollary 3.2 (from $\text{IPS}_\mathbb{Q}$ to $\text{IPS}_\mathbb{Z}$). *Boolean $\text{IPS}_\mathbb{Z}$ simulates boolean $\text{IPS}_\mathbb{Q}$, in the following sense: if there exists a size-s constant-free boolean IPS proof over $\mathbb{Q}$ from $\overline{\mathcal{F}}$ of H, for $\overline{\mathcal{F}}$ a set of assumptions written as constant-free algebraic circuits over $\mathbb{Z}$ and H a constant-free algebraic circuit over $\mathbb{Z}$, then there exists a size $\le 4s$ constant-free boolean $\text{IPS}_\mathbb{Z}$ proof of $M \cdot H$, for some $M \in \mathbb{Z} \setminus \{0\}$, such that $\tau(M) \le 4s$.*

Theorem 3.3. *Under the Shub and Smale Hypothesis, there are no* $\text{poly}(n)$*-size constant-free (boolean) IPS refutations of the binary value principle* BVP_n *over* $\mathbb{Q}$.

Proof: Given Cor. 3.2, it suffices to prove the statement for constant-free (boolean) $\text{IPS}_\mathbb{Z}$. We proceed to prove the contrapositive. Suppose that the binary value principle $1 + \sum_{i=1}^{i=n} 2^{i-1} x_i = 0$ has a constant-free $\text{IPS}_\mathbb{Z}$ refutation (using only the boolean axioms) of size $\text{poly}(n)$. We will show that there is a sequence of nonzero natural numbers c_m such that $\tau(c_m m!) \le (\log m)^c$, for all $m \ge 2$, where c is a constant independent of m. In other words, we will show that $(c_m m!)_{m=1}^{\infty}$ is easy.

Assume that $C(\overline{x}, y, \overline{z})$ is the polynomial-size constant-free boolean $\text{IPS}_\mathbb{Z}$ refutation of $1 + \sum_{i=1}^{i=n} 2^{i-1} x_i = 0$ (here we only have a single placeholder variable y for the single non-boolean axiom, that is, the binary value principle). For simplicity, denote $G(\overline{x}) = 1 + \sum_{i=1}^{i=n} 2^{i-1} x_i$, $F_i(\overline{x}) = x_i^2 - x_i$, and $\overline{F}(\overline{x}) = \overline{x}^2 - \overline{x}$.

We know that there exists an integer constant $M \ne 0$ such that

$$C\left(\overline{x}, G(\overline{x}), \overline{F}(\overline{x})\right) = M\,. \tag{6}$$

For every integer $0 \le k < 2^n$ we denote by $\overline{b}_k := (b_{k1}, \dots, b_{kn}) \in \{0,1\}^n$ its (positive, standard) binary representation, that is, $k = \sum_{i=1}^{i=n} b_{ki} 2^{i-1}$. Then, $F_i(\overline{b}_k) = 0$ and $G(\overline{b}_k) = 1 + k$, for all $1 \le i \le n$, $0 \le k < 2^n$. Hence, by eq. 6:

$$C(b_{k1}, \dots, b_{kn}, 1 + k, \overline{0}) = M, \text{ for every integer } 0 \le k < 2^n. \tag{7}$$

Claim 3.4. *M is divisible by every prime number less than 2^n.*

Proof of claim: For a fixed $0 \le k < 2^n$ and its binary representation $b_{k1}, \dots, b_{kn}$, consider $g(y) = C(b_{k1}, \dots, b_{kn}, y, \overline{0})$ as a univariate polynomial in $\mathbb{Z}[y]$. Then, $g(1+k) = M$ by eq. 7, and $g(0) = 0$ holds since $C(b_{k1}, \dots, b_{kn}, 0, \overline{0}) = 0$, by the definition of IPS. Because $g(0) = 0$, we know that $g(y) = y \cdot g^\star(y)$, for some $g^\star(y) \in \mathbb{Z}[y]$, meaning that $g(1+k) = (1+k) \cdot g^\star(1+k) = M$. Since $g^\star(y)$ is an integer polynomial, this implies that M is a product of $1 + k$.

Overall, this argument shows that for every $1 \le p \le 2^n$, M is divisible by p, and in particular M is divisible by every prime number less than 2^n. $\blacksquare_{\text{Claim}}$

Note that once we substitute the all-zero assignment $\overline{0}$ into eq. 6, we obtain a constant-free algebraic circuit of size $\text{poly}(n)$ with no variables computing M, thus $\tau(M) = \text{poly}(n)$. Then we can compute M^{2^n} using a constant-free algebraic circuit of size $\text{poly}(n)$ by taking M to the power 2, n many times (that is, using n repeated squaring), yielding $\tau(M^{2^n}) = \text{poly}(n)$.

Claim 3.5. *The power of every prime factor in $(2^n)!$ is at most 2^n.*

Proof of claim: We show that for every number $k \in \mathbb{N}$, the power of every prime factor of $k!$ is at most k. Let $p_1^{t_1} \cdots p_r^{t_r}$ be the prime factorisation of $k!$, namely $k! = p_1^{t_1} \cdots p_r^{t_r}$ where each p_i is a prime number and $p_i \ne p_j$, for all $i \ne j$. To compute t_i we consider the k products $k, (k-1), \dots, 1$, in $k! = k \cdot (k-1) \cdots 1$, out of which only each p_ith number is divisible by p_i, hence only $\lfloor \frac{k}{p_i} \rfloor$ numbers are divisible by p_i. Consider now only these $\lfloor \frac{k}{p_i} \rfloor$ numbers in $k!$ which are divisible by p_i, and write them as $p_i \cdot \lfloor \frac{k}{p_i} \rfloor, p_i \cdot (\lfloor \frac{k}{p_i} \rfloor - 1), \ldots, p_i \cdot 1$. Now we need once again to factor out the p_i products in $\lfloor \frac{k}{p_i} \rfloor, \lfloor \frac{k}{p_i} \rfloor - 1, \ldots, 1$. Hence, as before, we conclude that in these $\lfloor \frac{k}{p_i} \rfloor$ numbers only $\lfloor \frac{\lfloor \frac{k}{p_i} \rfloor}{p_i} \rfloor = \lfloor \frac{k}{p_i^2} \rfloor$ are divisible by p_i. Continuing in a similar fashion we obtain the equation $t_i = \lfloor \frac{k}{p_i} \rfloor + \lfloor \frac{k}{p_i^2} \rfloor + \lfloor \frac{k}{p_i^3} \rfloor + \cdots \leq \frac{k}{p-1}$. $\blacksquare_{\text{Claim}}$

Consider the poly(n)-size circuit for M^{2^n} that exists by assumption. Since M is divisible by every prime number between 1 and 2^n, and since every prime factor of $(2^n)!$ is clearly at most 2^n, we get that M^{2^n} is divisible by the 2^n-th power of *each* prime factor of $(2^n)!$. By Claim 3.5 the power of every prime factor of $(2^n)!$ is at most 2^n, and so M^{2^n} is divisible by $(2^n)!$. We conclude that there are nonzero numbers $c_n \in \mathbb{N}$ such that the sequence $\{c_n \cdot (2^n)!\}_{n=1}^{\infty}$ is computable by a sequence of constant-free algebraic circuits of size poly(n), that is, $\tau(c_n \cdot (2^n)!) \leq n^c$ for some constant c independent of n. It remains to show that not only the multiples of factorials *of powers of* 2 are easy, but also the multiples of factorials of *all* natural numbers are easy.

For every natural number m, let $n \in \mathbb{N}$ be such that $2^{n-1} \leq m \leq 2^n$. Because $(2^n)!$ is clearly divisible by $m!$, there exists some $c_m \in \mathbb{N}$, such that $c_n \cdot (2^n)! = c_m \cdot m!$, where c_n is the natural number for which we have showed the existence of poly(n)-size constant-free circuit computing $c_n \cdot (2^n)!$. Hence, this same circuit also computes $c_m \cdot m!$, meaning that $\tau(c_m \cdot m!) \leq n^b \leq (\log(2m))^b \leq (\log m)^c$, for some constants b and c independent of m. □

4 SIMULATIONS

We now show that boolean CPS (Definition 12) simulates boolean IPS for the language of $\{0,1\}$-unsatisfiable sets of polynomial *equations* over any ordered ring. Similarly, real CPS simulates algebraic IPS over $\mathbb{Q}$. This is the easier simulation result. For the conditional simulation of CPS by IPS assuming short refutations for the BVP see the full version [1].

We translate an input equality $f_i(\overline{x}) = 0$ into a pair of inequalities $f_i(\overline{x}) \geq 0$ and $-f_i(\overline{x}) \geq 0$. Note that an IPS proof is written as a general algebraic circuit (computing an element of an ideal), while a CPS proof is written as a more restrictive algebraic circuit, namely as a $\overline{y}$-conic circuit (computing an element of a cone). This means that a priori we cannot multiply an inequality by an arbitrary polynomial in CPS. We thus demonstrate how to do it when we have opposite-sign inequalities. In order to do this, we represent an arbitrary polynomial as the difference of two nonnegative expressions.

Proposition 4.1 (minus gate normalisation)**.** *Let $G(\overline{x})$ be an algebraic circuit computing a polynomial in the $\overline{x}$ variables over $\mathbb{Q}$. Then, there is an algebraic circuit of the form $G_P(\overline{x}) - G_N(\overline{x})$ computing the same polynomial as $G(\overline{x})$ where G_P and G_N are $\emptyset$-conic. The size of G_P, G_N is at most linear in the size of G.*

Proof: This is somewhat reminiscent of Strassen's conversion of a circuit with division gate to a circuit with only a single division gate at the top [49]. We are going to break inductively each node into a pair of nodes computing the positive and negative parts of the polynomial computed in that node. Formally, we define the circuits G_P, G_N (that may have common nodes) by induction on the size of G as follows:

Case 1: $G = x_i$, for $x_i \in \overline{x}$. Then, $G_P := \frac{1}{2}(x_i^2 + 1)$, $G_N := \frac{1}{2}(x_i - 1)^2$.
Case 2: $G = \alpha$, for α a constant in the ring. Then

$$G_P := \alpha,\ G_N := 0, \quad \text{if } \alpha \geq 0;$$
$$G_P := 0,\ G_N := \alpha, \quad \text{if } \alpha < 0.$$

Case 3: $G = F + H$. Then, $G_P := F_P + H_P$ and $G_N := F_N + H_N$.
Case 4: $G = F \cdot H$. Then, $G_P := F_P \cdot H_P + F_N \cdot H_N$ and $G_N := F_P \cdot H_N + F_N \cdot H_P$.

The size of both G_P, G_N is $O(|G|)$, namely linear in the size of G. This is because we only add constantly many new nodes in G_P, G_N for any original node in G; note that since we construct a new *circuit* computing the same polynomial as G, we can re-use nodes computed already, in case 4: for example, F_P is the same node used in G_P and G_N (hence, indeed, the number of new added nodes for every original node in G is constant). □

Theorem 4.2. *Real CPS simulates algebraic IPS as a proof system for the language of unsatisfiable sets of polynomial equations over $\mathbb{Q}$. In other words, there exists a constant c such that for any polynomial $p(\overline{x})$ and a set of polynomial equations $\overline{\mathcal{F}}$, if $p(\overline{x})$ has an IPS proof of size s from $\overline{\mathcal{F}}$ then there is a CPS proof of $p(\overline{x})$ from $\overline{\mathcal{F}}$ of size at most s^c. Furthermore, boolean CPS simulates boolean IPS (for any ordered ring).*

Proof of Thm. 4.2. We are going to simulate both the boolean and the algebraic versions of IPS. The proof in both cases is the same. Assume that $C(\overline{x}, \overline{y})$ is the IPS proof of $p(\overline{x})$ from $\overline{\mathcal{F}} = \{f_i(\overline{x}) = 0\}_{i=1}^{\ell}$, of size s, and let $\overline{y} = \{y_1, \ldots, y_\ell\}$ be the placeholder variables for the equations in $\overline{\mathcal{F}}$. We assume for simplicity that if we simulate the *boolean* version of IPS the boolean axioms $\overline{x}^2 - \overline{x}$ are also part of $\overline{\mathcal{F}}$ (while if we simulate the *algebraic* version of IPS these axioms are not part of $\overline{\mathcal{F}}$). We use the following claim which is proved by a standard process that factors out the $\overline{y}$ variables one by one:

Claim 4.3. *Let $C(\overline{x}, \overline{y})$ be a circuit of size s, where $\overline{y} = \{y_1, \ldots, y_\ell\}$ and such that $C(\overline{x}, \overline{0}) = 0$. Then C can be written as a sum of circuits with only a polynomial increase in size as follows: $C(\overline{x}, \overline{y}) = \sum_{i=1}^{\ell} y_i \cdot C_i(\overline{x}, \overline{y})$.*

Proof of claim: We proceed by a standard process to factor out the $\overline{y}$ variables one by one. Beginning with y_1 we get:

$$C(\overline{y}, \overline{x}) = y_1 \cdot \big(C(1, \overline{y}', \overline{x}) - C(0, \overline{y}', \overline{x})\big) + C(0, \overline{y}', \overline{x}),$$

where $\overline{y}'$ denotes the vector of variables $(y_2, \ldots, y_\ell)$. In a similar manner we factor out the variable y_2 from $C(0, \overline{y}', \overline{x})$. Continuing in a similar fashion we conclude the claim. Notice that the size of the resulting circuit is $O(|C|^2)$, and that in the final iteration of the construction we factor out y_ℓ from $C(\overline{0}, y_\ell, \overline{x})$ it must hold that $C(\overline{0}, y_\ell, \overline{x}) = y_1 \cdot \Big(C(\overline{0}, 1, \overline{x}) - C(\overline{0}, 0, \overline{x})\Big) + C(\overline{0}, 0, \overline{x}) = y_1 \cdot C(\overline{0}, 1, \overline{x})$, because by assumption $C(\overline{0}, 0, \overline{x}) = 0$. $\blacksquare_{\text{Claim}}$

By this claim we have

$$\begin{aligned} C(\overline{x}, \overline{y}) &= \sum\nolimits_{i=1}^{\ell} y_i \cdot C_i(\overline{x}, \overline{y}) \\ &= \sum\nolimits_{i=1}^{\ell} y_i \cdot C_{i,P}(\overline{x}, \overline{y}) - \sum\nolimits_{i=1}^{\ell} y_i \cdot C_{i,N}(\overline{x}, \overline{y}), \end{aligned} \qquad (8)$$

where $C_{i,P}(\overline{x},\overline{y}), C_{i,N}(\overline{x},\overline{y})$ are the positive and negative parts of $C_i(\overline{x},\overline{y})$, respectively, that exist by Prop. 4.1, written as circuits in which no negative constants occur (we do not need to distinguish between the variables $\overline{x}$ and $\overline{y}$ here).

We wish to construct now a CPS refutation of $\overline{\mathcal{F}}$. Our corresponding set of inequalities $\overline{\mathcal{H}}$ will consist of $f_i(\overline{x}) \geq 0, -f_i(\overline{x}) \geq 0$, for every $i \in [\ell]$. In total, $|\overline{\mathcal{H}}| = 2\ell$. Accordingly, our CPS refutation of $\overline{\mathcal{F}}, \overline{\mathcal{H}}$, will have 2ℓ placeholder variables for the axioms in $\overline{\mathcal{H}}$ denoted as follows: $\overline{y}_P$ are the ℓ placeholder variables $y_{i,P}$ corresponding to $f_i(\overline{x}) \geq 0$, $i \in [l]$, $\overline{y}_N$ are the ℓ placeholder variables $y_{i,N}$ corresponding to $-f_i(\overline{x}) \geq 0$, $i \in [l]$.

Since $C_{i,P}$ and $C_{i,N}$ are $\emptyset$-conic circuits,

$$\sum_{i=1}^{\ell} y_{i,P} \cdot C_{i,P}(\overline{x},\overline{y}_P,\overline{y}_N) + \sum_{i=1}^{\ell} y_{i,N} \cdot C_{i,N}(\overline{x},\overline{y}_P,\overline{y}_N)$$

is a $(\overline{y}_P,\overline{y}_N)$-conic circuit. It constitutes a CPS proof of $p(\overline{x})$ from the assumptions $f_i(\overline{x}) \geq 0, -f_i(\overline{x}) \geq 0$, for $i \in [\ell]$ of size linear in $|C|$ (as before, we denote by $\overline{\mathcal{F}}$ the vector $f_1(\overline{x}),\dots,f_\ell(\overline{x})$):

$$\begin{aligned}\sum_{i=1}^{\ell} f_i(\overline{x}) \cdot C_{i,P}(\overline{x},\overline{\mathcal{F}}) + \sum_{i=1}^{\ell} (-f_i(\overline{x})) \cdot C_{i,N}(\overline{x},\overline{\mathcal{F}}) \\ = \sum_{i=1}^{\ell} f_i(\overline{x}) \cdot \left(C_{i,P}(\overline{x},\overline{\mathcal{F}}) - C_{i,N}(\overline{x},\overline{\mathcal{F}})\right) \\ = \sum_{i=1}^{\ell} f_i(\overline{x}) \cdot C_i(\overline{x},\overline{\mathcal{F}}) = C(\overline{x},\overline{\mathcal{F}}) = p(\overline{x}).\end{aligned}$$

□

5 CONCLUSIONS

This work demonstrates that a simple subset-sum principle, written as a linear equation, captures, in the boolean case (i.e., when variables range over $\{0,1\}$), the apparent advantage of semi-algebraic proofs over algebraic proofs in the following sense: it is necessary for any boolean algebraic proof system that simulates full boolean semi-algebraic proofs to admit short refutations of the principle; and if the algebraic proof system is strong enough to be able to efficiently perform basic bit arithmetic, this condition is also *sufficient* to achieve such a simulation. To formalize these results we introduce a very strong proof system CPS that derives polynomials in the cone of initial axioms instead of in the ideal.

We show that CPS is expected to be stronger than even the very strong algebraic Ideal Proof System (IPS) formulated by Grochow and Pitassi in [25], since our subset-sum principle is hard for IPS assuming the hardness of computing factorials [46]. We establish a related lower bound on IPS refutation-size based on the τ-conjecture [46]. These lower bounds extend the results of Forbes et al. [18]: whereas [18] showed how to obtain restricted IPS lower bounds for certain subset-sum instances, based on known lower bounds against restricted circuit classes, we show how to obtain *general* IPS lower bounds based on specific hardness assumptions from algebraic complexity.

5.1 Relation to Other Work

Bit arithmetic and semi-algebraic proofs. In this work we show how to reason about the bits of polynomial expressions within algebraic proofs. Bit arithmetic in proof complexity was used before in Frege systems (see [19] following [10]). Independently of our work [27], Impagliazzo et al. [29] considered the possibility to *effectively* simulate *weak* semi-algebraic proofs using medium-strength algebraic proofs. They have considered expressing and reasoning with the bits of algebraic expressions, as we do in this work. However, their proof methods and results are fundamentally different from ours: first, they work in the weak proof systems regime, while we work in the strong systems regime. I.e., they aim to effectively simulate *weak* proof systems like constant degree sum-of-squares (in which polynomials are written as sum of monomials), while we aim to simulate *very strong* proof systems such as CPS (essentially, Positivstellensatz written as algebraic circuits). Second, they use a different way to express bits in their work. This is done in order to be able to reason about bits with bounded-depth algebraic circuits, while we do not need this mechanism. Third, they show only *effective* simulation and not simulation (namely, before the algebraic proofs can simulate a system of polynomial equations or inequalities, the equations and inequalities need to be pre-processed, that is, translated "off-line" to their bit-vector representation). Fourth, they do not consider the VAL function nor the binary value principle, while our work shows that essentially this is a necessary ingredient in a full simulation of strong semi-algebraic proof systems. In fact, we have the following:

Assuming the Shub-Smale hypothesis, our results *rule out the possibility that even a very strong algebraic proof system such as IPS simulates (in contrast to the weaker notion of an effective simulation) even a weak semi-algebraic proof system like constant degree* SoS *measured by monomial size.* In other words, assuming Shub-Smale hypothesis, we rule-out the possibility that the arguments in [29] (or any other argument) can yield a simulation of constant degree SoS by algebraic proofs operating with constant depth algebraic circuits (depth-d PC in [29][6]). It remains however open whether depth-d PC simulates constant degree SoS *for the language of unsatisfiable CNF formulas* or for unsatisfiable sets of linear equations with small coefficients.

Subset-sum lower bounds in proofs complexity. Different instances of the subset sum problem have been considered as hard instances for algebraic proof systems. For example, Impagliazzo et al. [30] provided an exponential size lower bound on the symmetric subset sum instance $x_1+\cdots+x_n = n+1$, for boolean x_i's in the polynomial calculus refutation system. Grigoriev [22] proved that the version $\sum_{i=1}^{n} x_i = r$ for a non-integer $r \approx \frac{n}{2}$ requires linear degrees to refute in Positivstellensatz, and [23] later transformed this idea into an exponential-size lower bound for both Positivstellensatz and static high-degree Lovasz-Schrijver proof systems. Moreover, as already mentioned, our lower bounds can be seen as an extension to the case of general IPS refutations of the approach introduced by Forbes et al. [18].

The work of Part and Tzameret [37] established unconditional exponential lower bounds on the size of resolution over linear equations refutations of the binary value principle, over any sufficiently large field $\mathbb{F}$, denoted Res(lin$_{\mathbb{F}}$). The proof techniques in [37] are completely different from the current work, but these results demonstrate that using instances with large coefficients in proof complexity provides new insight into the complexity of proof systems.

[6] Here we use the fact that IPS simulates depth-d PC.

ACKNOWLEDGEMENT

We wish to thank Michael Forbes, Dima Itsykson, Toni Pitassi and Dima Sokolov for useful discussions at various stages of this work.

REFERENCES

[1] Yaroslav Alekseev, Dima Grigoriev, Edward A. Hirsch and Iddo Tzameret. Semi-Algebraic Proofs, IPS Lower Bounds and the τ-Conjecture: Can a Natural Number be Negative? ArXiV: http://arxiv.org/abs/1911.06738

[2] Albert Atserias and Tuomas Hakoniemi. Size-degree trade-offs for sums-of-squares and Positivstellensatz proofs. In *34th Computational Complexity Conference, CCC 2019*, pages 24:1–24:20, 2019.

[3] Boaz Barak, Fernando G. S. L. Brandão, Aram Wettroth Harrow, Jonathan A. Kelner, David Steurer, and Yuan Zhou. Hypercontractivity, sum-of-squares proofs, and their applications. In *STOC*, pages 307–326, 2012.

[4] Paul Beame, Russell Impagliazzo, Jan Krajíček, Toniann Pitassi, and Pavel Pudlák. Lower bounds on Hilbert's Nullstellensatz and propositional proofs. *Proc. London Math. Soc. (3)*, 73(1):1–26, 1996.

[5] Christoph Berkholz. The relation between polynomial calculus, sherali-adams, and sum-of-squares proofs. In *35th Symposium on Theoretical Aspects of Computer Science, STACS 2018, February 28 to March 3, 2018, Caen, France*, pages 11:1–11:14, 2018.

[6] Grigoriy Blekherman, Pablo A. Parrilo, and Rekha Thomas, editors. *Semidefinite Optimization and Convex Algebraic Geometry.* MPS-SIAM Series on Optimization. Society for Industrial and Applied Mathematics (SIAM), March 2013.

[7] Lenore Blum, Felipe Cucker, Michael Shub, and Steve Smale. *Complexity and Real Computation.* Springer-Verlag, Berlin, Heidelberg, 1998.

[8] Lenore Blum, Mike Shub, and Steve Smale. On a theory of computation and complexity over the real numbers: np- completeness, recursive functions and universal machines. *Bull. Amer. Math. Soc. (N.S.)*, 21(1):1–46, 07 1989.

[9] Peter Bürgisser. On defining integers and proving arithmetic circuit lower bounds. *Computational Complexity*, 18(1):81–103, 2009.

[10] Samuel R. Buss. Polynomial size proofs of the propositional pigeonhole principle. *The Journal of Symbolic Logic*, (52):916–927, 1987.

[11] Samuel R. Buss, Russell Impagliazzo, Jan Krajíček, Pavel Pudlák, Alexander A. Razborov, and Jiří Sgall. Proof complexity in algebraic systems and bounded depth Frege systems with modular counting. *Computational Complexity*, 6(3):256–298, 1996.

[12] Qi Cheng. On the ultimate complexity of factorials. *Theor. Comput. Sci.*, 326(1-3):419–429, October 2004.

[13] Matthew Clegg, Jeffery Edmonds, and Russell Impagliazzo. Using the Groebner basis algorithm to find proofs of unsatisfiability. In *Proceedings of the 28th Annual ACM Symposium on the Theory of Computing*, pages 174–183, New York, 1996.

[14] Stephen A. Cook and Robert A. Reckhow. On the lengths of proofs in the propositional calculus (preliminary version). In *1974*, pages 135–148, 1974.

[15] Stephen A. Cook and Robert A. Reckhow. The relative efficiency of propositional proof systems. *J. Symb. Log.*, 44(1):36–50, 1979.

[16] W. de Melo and B. F. Svaiter. The cost of computing integers. *Proc. Amer. Math. Soc.*, 124(5):1377–1378, 1996.

[17] Noah Fleming, Pravesh Kothari, and Toniann Pitassi. Semialgebraic proofs and efficient algorithm design. *Electronic Colloquium on Computational Complexity (ECCC)*, 26:106, 2019.

[18] Michael A. Forbes, Amir Shpilka, Iddo Tzameret, and Avi Wigderson. Proof complexity lower bounds from algebraic circuit complexity. In *31st Conference on Computational Complexity, CCC 2016, May 29 to June 1, 2016, Tokyo, Japan*, pages 32:1–32:17, 2016.

[19] Andreas Goerdt. Cutting plane versus Frege proof systems. *Computer Science Logic, 4th Workshop, CSL '90, Heidelberg, Germany, October 1-5, 1990, Proceedings*, volume 533 of *Lecture Notes in Computer Science*, pages 174–194. Springer, 1990.

[20] Mika Göös, Pritish Kamath, Robert Robere, and Dmitry Sokolov. Adventures in monotone complexity and TFNP. In *10th Innovations in Theoretical Computer Science Conference, ITCS 2019, January 10-12, 2019, San Diego, California, USA*, pages 38:1–38:19, 2019.

[21] Mika Göös and Toniann Pitassi. Communication lower bounds via critical block sensitivity. *SIAM J. Comput.*, 47(5):1778–1806, 2018.

[22] D. Grigoriev. Complexity of Positivstellensatz proofs for the knapsack. *Comput. Complexity*, 10(2):139–154, 2001.

[23] Dima Grigoriev, Edward A. Hirsch, and Dmitrii V. Pasechnik. Complexity of semialgebraic proofs. *Mosc. Math. J.*, 2(4):647–679, 805, 2002.

[24] Dima Grigoriev and Nicolai Vorobjov. Complexity of Null- and Positivstellensatz proofs. *Ann. Pure Appl. Logic*, 113(1-3):153–160, 2002.

[25] Joshua A. Grochow and Toniann Pitassi. Circuit complexity, proof complexity, and polynomial identity testing: The ideal proof system. *J. ACM*, 65(6):37:1–37:59, 2018.

[26] David Hilbert. *Hilbert's invariant theory papers.* Lie Groups: History, Frontiers and Applications, VIII. Math Sci Press, Brookline, Mass., 1978. Translated from the German by Michael Ackerman, With comments by Robert Hermann.

[27] Edward Hirsch and Iddo Tzameret. Nullstellensatz is equivalent to sum-of-squares, over algebraic circuits. In Proof Complexity (Dagstuhl Seminar 18051), pages 124–157. Schloss Dagstuhl Leibniz-Zentrum fuer Informatik, 2018., Feb. 2018. https://materials.dagstuhl.de/files/18/18051/18051.IddoTzameret.Slides.pptx.

[28] Trinh Huynh and Jakob Nordström. On the virtue of succinct proofs: amplifying communication complexity hardness to time-space trade-offs in proof complexity. In *Proceedings of the 44th Symposium on Theory of Computing (STOC)*, pages 233–248. ACM, 2012.

[29] Russell Impagliazzo, Sasank Mouli, and Toniann Pitassi. The surprising power of constant depth algebraic proofs. *Electronic Colloquium on Computational Complexity (ECCC)*, 26:24, 2019.

[30] Russell Impagliazzo, Pavel Pudlák, and Jiří Sgall. Lower bounds for the polynomial calculus and the gröbner basis algorithm. *Computational Complexity*, 8(2):127–144, 1999.

[31] Dmitry Itsykson and Arist Kojevnikov. Lower bounds of static lovasz-schrijver calculus proofs for tseitin tautologies. *Zapiski Nauchnyh Seminarov POMI*, 340:10–32, 2006. (in Russian). English translation appeared in Journal of Mathematical Sciences 145(3):4942-4952, 2007.

[32] J. L. Krivine. Anneaux preordonnes. *Journal d'Analyse Mathématique*, 12(1):307–326, 1964.

[33] Fu Li, Iddo Tzameret, and Zhengyu Wang. Characterizing propositional proofs as noncommutative formulas. In *SIAM Journal on Computing*, volume 47, pages 1424–1462, 2018.

[34] L. Lovász. Stable sets and polynomials. *Discrete Mathematics*, 124:137–153, 1994.

[35] L. Lovász and A. Schrijver. Cones of matrices and set-functions and 0–1 optimization. *SIAM Journal on Optimization*, 1:166–190, 1991.

[36] Ryan O'Donnell and Yuan Zhou. Approximability and proof complexity. In *Proceedings of the Twenty-Fourth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2013, January 6-8, 2013*, pages 1537–1556, 2013.

[37] Fedor Part and Iddo Tzameret. Resolution with counting: Dag-like lower bounds and different moduli. *To appear in 11th Innovations in Theoretical Computer Science Conference (ITCS) 2020, January, 2020, Seattle, WA, USA*, 2020.

[38] Toniann Pitassi Paul Beame and Nathan Segerlind. Lower bounds for lovász–schrijver systems and beyond follow from multiparty communication complexity. *SIAM Journal on Computing*, 37(3):845–869, 2007.

[39] Toniann Pitassi. Unsolvable systems of equations and proof complexity. In *Proceedings of the International Congress of Mathematicians, Vol. III (Berlin, 1998)*, number Vol. III, pages 451–460, 1998.

[40] Tonnian Pitassi and Iddo Tzameret. Algebraic proof complexity: Progress, frontiers and challenges. *ACM SIGLOG News*, 3(3), 2016.

[41] Pavel Pudlák. On the complexity of the propositional calculus. In *Sets and proofs (Leeds, 1997)*, volume 258 of *London Math. Soc. Lecture Note Ser.*, pages 197–218. Cambridge Univ. Press, Cambridge, 1999.

[42] Mihai Putinar. Positive polynomials on compact semi-algebraic sets. *Indiana University Mathematics Journal*, 42(3):969–984, 1993.

[43] Alexander A. Razborov. Lower bounds for the polynomial calculus. *Comput. Complexity*, 7(4):291–324, 1998.

[44] Robert A. Reckhow. *On the lengths of proofs in the propositional calculus.* PhD thesis, University of Toronto, 1976.

[45] Amir Shpilka and Amir Yehudayoff. Arithmetic circuits: A survey of recent results and open questions. *Foundations and Trends in Theoretical Computer Science*, 5(3-4):207–388, 2010.

[46] Michael Shub and Steve Smale. On the intractability of Hilbert's Nullstellensatz and an algebraic version of "NP≠P?". *Duke Math. J.*, 81:47–54, 1995.

[47] Steve Smale. Mathematical problems for the next century. *The Mathematical Intelligencer*, 20(2):7–15, 1998.

[48] Gilbert Stengle. A Nullstellensatz and a Positivstellensatz in semialgebraic geometry. *Mathematische Annalen*, 207(2):87–97, 1974.

[49] Volker Strassen. Vermeidung von divisionen. *J. Reine Angew. Math.*, 264:182–202, 1973. (in German).

[50] Leslie G. Valiant. Completeness classes in algebra. In *Proceedings of the 11th Annual ACM Symposium on the Theory of Computing*, pages 249–261. ACM, 1979.

[51] Leslie G. Valiant. The complexity of computing the permanent. *Theor. Comput. Sci.*, 8:189–201, 1979.

[52] Leslie G. Valiant. Reducibility by algebraic projections. *Logic and Algorithmic: International Symposium in honour of Ernst Specker*, 30:365–380, 1982.