On the Degree Automatability of Sum-of-Squares Proofs

Alex Bortolotti [*]   Monaldo Mastrolilli [†]   Luis Felipe Vargas[‡]

Abstract

The Sum-of-Squares (SoS) hierarchy, also known as Lasserre hierarchy, has emerged as a promising tool in optimization. However, it remains unclear whether fixed-degree SoS proofs can be automated [O'Donnell (2017)]. Indeed, there are examples of polynomial systems with bounded coefficients that admit low-degree SoS proofs, but these proofs necessarily involve numbers with an exponential number of bits, implying that low-degree SoS proofs cannot always be found efficiently.

A sufficient condition derived from the Nullstellensatz proof system [Raghavendra and Weitz (2017)] identifies cases where bit complexity issues can be circumvented. One of the main problems left open by Raghavendra and Weitz is proving any result for refutations, as their condition applies only to polynomial systems with a large set of solutions.

In this work, we broaden the class of polynomial systems for which degree-d SoS proofs can be automated. To achieve this, we develop a new criterion and we demonstrate how our criterion applies to polynomial systems beyond the scope of Raghavendra and Weitz's result. In particular, we establish a separation for instances arising from Constraint Satisfaction Problems (CSPs). Moreover, our result extends to refutations, establishing that polynomial-time refutation is possible for broad classes of polynomial time solvable constraint problems, highlighting a first advancement in this area.

Keywords — Sum of squares, Polynomial calculus, Polynomial ideal membership, Polymorphisms, Gröbner basis theory, Constraint satisfaction problems, Proof complexity.

---

[*]University of Applied Sciences and Arts of Southern Switzerland, IDSIA, Lugano, Switzerland. E-mail: alex.bortolotti@supsi.ch.

[†]University of Applied Sciences and Arts of Southern Switzerland, IDSIA, Lugano, Switzerland. E-mail: monaldo.mastrolilli@supsi.ch.

[‡]University of Applied Sciences and Arts of Southern Switzerland, IDSIA, Lugano, Switzerland. E-mail: luis.vargas@supsi.ch.

Contents

1 Introduction

Semidefinite programming (SDP) relaxations have been a powerful technique for approximation algorithm design ever since the celebrated result of Goemans and Williamson [26]. With the aim to construct stronger and stronger SDP relaxations, the Sum-of-Squares (SoS) hierarchy has emerged as a systematic and versatile method for approximating many combinatorial optimization problems, see e.g. [42, 51, 24, 43]. However, fundamental questions remain unanswered. For instance, it is still unknown under what conditions SoS can be *automated*, meaning whether one can find a degree-d SoS proof in time $n^{O(d)}$, provided it exists. O'Donnell [48] observed that the prevailing belief regarding the automatability of SoS using ellipsoid algorithms is not entirely accurate. Issues may arise when the only degree-d proofs contain exceedingly large coefficients, thereby hindering the ellipsoid method from operating within polynomial time. In this paper, we establish novel conditions that ensure SoS automatability.

Polynomial optimization. Polynomial optimization asks for minimizing a polynomial over a given set of polynomial constraints. That is, given polynomials $r, p_1, \ldots, p_m \in \mathbb{R}[x_1, \ldots, x_n]$, the task is to find (or approximate) the infimum of the following probth:

$$\inf_{x \in S} r(x), \quad \text{where} \quad S = \{x \in \mathbb{R}^n \mid p_1(x) = \cdots = p_m(x) = 0\}. \tag{1}$$

Typically, S is defined by a set of equality constraints, in this case $\mathcal{P} = \{p_1, \ldots, p_m\}$, as well as a set of inequality constraints, $\mathcal{Q}$. For all applications considered here, however, it suffices to restrict to the case where $\mathcal{Q} = \emptyset$ and S is finite, enabling the modeling of various relevant combinatorial problems. Nonetheless, we emphasize that our results readily extend to the semialgebraic setting, where $\mathcal{Q} \neq \emptyset$. For further details, we refer to Section 3.5.

A common approach for solving (or approximating) a polynomial optimization problem is by means of sums of squares of polynomials, as we now explain.

Definition 1.1 (SoS Proof System). *Let $\mathcal{P} = \{p_1 = 0, \ldots, p_m = 0\}$ be a set of polynomial equations, and consider a polynomial $r \in \mathbb{R}[x_1, \ldots, x_n]$. An SoS proof of "$r \geq 0$" (over S) from $\mathcal{P}$ is an identity of the form $r = \sum_{i=1}^{t_0} s_i^2 + \sum_{i=1}^{m} h_i p_i$, where $s_i, h_i \in \mathbb{R}[x_1, \ldots, x_n]$. Moreover, we say that the above SoS proof has degree at most d if $\deg(s_i^2) \leq d$, for all $i \in [t_0]$, and $\deg(h_i p_i) \leq d$ for all $i \in [s]$. An SoS refutation of $\mathcal{P}$ is an SoS proof of "$-1 \geq 0$" from $\mathcal{P}$.*

The SoS hierarchy is based on the following observation: if there exists an SoS proof of "$r - \theta \geq 0$" from $\mathcal{P}$, then we have that $\min_{x \in S} r(x) \geq \theta$. Moreover, the supremum of the values θ such that there is an SoS proof of "$r - \theta \geq 0$" from $\mathcal{P}$ of degree d, is called d-th SoS relaxation, also known as the d-th Lasserre relaxation of problem (1) [42, 51]. It turns out that the d-th SoS relaxation can be formulated as an SDP of size $n^{O(d)}$.

SoS relaxations have gained increasing popularity and success; yet, they remain a relatively recent development. Fundamental questions about their properties and capabilities still lack definitive answers. O'Donnell [48] posed the open problem of identifying meaningful conditions that ensure that "small" SoS proofs can be found. We will consider systems $\mathcal{P} = \{p_1 = 0, \ldots, p_m = 0\}$ of polynomials and an "input" polynomial r of degree at most d, with the (mild) assumption that the bit complexity needed to represent $\mathcal{P}$ and r is polynomial in n. Moreover, we assume that $\mathcal{P}$ is explicitly Archimedean, i.e. there is $N < 2^{poly(n^d)}$ such that there exists a "small" SoS proof of "$N - x_i^2 \geq 0$" from $\mathcal{P}$ for any variable x_i. We restate O'Donnell's question as follows: *Consider an explicitly Archimedean polynomial system $\mathcal{P}$; under what conditions on $\mathcal{P}$ does the following property hold?*

(P) Assume there exists an SoS proof of "$r \geq 0$" from $\mathcal{P}$ of degree $2d$. Then, for every $\varepsilon > 0$, there also exists an SoS proof of "$r + \varepsilon \geq 0$" from $\mathcal{P}$ with degree $O(d)$ and coefficients bounded by $2^{poly(n^d, \lg \frac{1}{\varepsilon})}$.

The assumption of explicitly Archimedeanity guarantees that if there exists an approximate SoS proof of "$r - \theta \geq 0$", then there exists an (exact) SoS proof of "$r - \theta + \varepsilon \geq 0$", up to any arbitrary precision ε. Moreover, explicit Archimedeanity implies that the SDP has no duality gap [37]. Therefore, it is often assumed in literature since numerical methods for solving SDPs are guaranteed to converge only when the duality gap is zero.

Nsatz criterion. Since O'Donnell [48] raised his question in 2017, very few papers have been published that address this issue. An initial elegant solution to this question is provided by Raghavendra and Weitz [57], which is based on the Nullstellensatz proof system [4], as we will now outline. For additional results from the literature related to this problem, see Section 1.

We denote the vector space of polynomials for variables $x_1, \ldots, x_n$ up to degree d as $\mathbb{R}[x_1, \ldots, x_n]_d$. Moreover, we denote by $\mathrm{I}(S) = \{p \in \mathbb{R}[x_1, \ldots, x_n] \mid p(x) = 0 \; \forall x \in S\}$ the vanishing ideal generated by S, and by $\mathrm{I}_d(S) = \mathrm{I}(S) \cap \mathbb{R}[x_1, \ldots, x_n]_d$ the d-truncated ideal.

Definition 1.2 (Nsatz Proof System)**.** *Consider a system of polynomial equations $\mathcal{P} = \{p_1 = 0, \ldots, p_m = 0\}$. A Nullstellensatz (Nsatz) proof of "$p = 0$" from $\mathcal{P}$ is a sequence of polynomials $(h_1, \ldots, h_m)$ such that the polynomial identity $p = \sum_{i=1}^m h_i p_i$ holds. We say that the proof has degree d if $\max_i \{\deg h_i p_i\} = d$. We say that $\mathcal{P}$ is Nsatz d-complete over S if for every $p \in \mathrm{I}_d(S)$, the identity "$p = 0$" can be derived using a degree-$O(d)$ Nsatz proof from $\mathcal{P}$.*

Next, we recall the criterion proposed by Raghavendra and Weitz for the algebraic setting [1]. Moreover, for the sake of clarity of the exposition, we present their result in the case S is finite. We define the algebraic variety S as the set of common zeros of $\mathcal{P} = \{p_1 = 0, \ldots, p_m = 0\}$. We first observe that this criterion necessitates a technical condition on the solution set S, referred to as δ-*spectrality*, which we will outline below. Let $\mathbf{v}_d$ represent the column vector whose entries correspond to the elements of the standard monomial basis of $\mathbb{R}[x_1, \ldots, x_n]_d$. For $\alpha \in \mathbb{R}^n$, $\mathbf{v}_d(\alpha)$ denotes the vector of real numbers obtained by evaluating the entries of $\mathbf{v}_d$ at α.

Definition 1.3. *Let S be a finite algebraic variety. We say that S is δ-spectrally rich up to degree d if every nonzero eigenvalue of the moment matrix $\frac{1}{|S|} \sum_{\alpha \in S} \mathbf{v}_d(\alpha) \mathbf{v}_d^{\mathsf{T}}(\alpha)$ is at least δ.*

This property holds for $\frac{1}{\delta} = 2^{poly(n^d)}$ in many natural instances, for example when $S \subseteq \{0, 1\}^n$, or more in general, when $S \subseteq D^n$ for any finite domain $D \subseteq \mathbb{Q}$ (see Section 3.2 and [57]).

Theorem 1.4 (Nsatz criterion [57])**.** *Let $\mathcal{P}$ be a system of polynomial equalities over n variables with solution set S. Assume that*

(1) S is δ-spectrally rich up to degree d.

(2) $\mathcal{P}$ is Nsatz d-complete over S.

Let r be a polynomial and assume there exists an SoS proof of "$r \geq 0$" from $\mathcal{P}$ of degree d. Then, there also exists an SoS proof of "$r \geq 0$" from $\mathcal{P}$ with degree $O(d)$ and with absolute values of the coefficients bounded by $2^{poly(n^d, \lg \frac{1}{\delta})}$.

---

[1]We remark that their criterion is formulated for the semialgebraic setting, i.e. when there are also inequalities.

This criterion is applicable to various optimization problems, including MAX-CLIQUE, MATCH-ING, and MAX-CSP [57]. However, the NSATZ criterion is subject to significant limitations. First, the criterion is sufficient but not necessary. Second, it is important to observe how the NSATZ criterion (see condition *(2)* in Theorem 1.4) is influenced by the complexity of a well-known problem known as the *Ideal Membership Problem* (IMP). This problem involves determining whether an input polynomial r belongs to the ideal generated by $\{p_1, \ldots, p_m\}$. We denote the IMP where the input polynomial r has degree at most $d = O(1)$ as IMP_d. The IMP was first studied by Hilbert [32] and is a fundamental algorithmic problem with significant applications in solving polynomial systems and polynomial identity testing (see, for example, [21]). In general, the IMP is notoriously intractable, and the results of Mayr and Meyer demonstrate that it is EXPSPACE-complete [46, 47]. It remains unclear under what conditions the IMP is tractable within the NSATZ proof system, specifically regarding when condition *(2)* in Theorem 1.4 is satisfied. More importantly, the limitations of the NSATZ proof system (see e.g. [24]) affect the applicability of Theorem 1.4. In simpler terms, it is intuitive to suggest that if we could replace the NSATZ proof system with a more powerful proof system, we would be able to broaden the applicability of the criterion to new problems.

Finally, a key limitation—and one of the main open problems left by Raghavendra and Weitz [57, 62]—is the inapplicability of the NSATZ criterion to SoS refutations.

For example the NSATZ criterion does not allow one to show that the following decision problem can be solved in polynomial time.

Problem 1.5 (Degree-d Sum-of-Squares Refutation for CSP). *Given a Constraint Satisfaction Problem (CSP) with constraints $\phi_1(x) = 0, \ldots, \phi_m(x) = 0$ over a finite domain, decide whether:*

- **YES**: *There exists a degree-d sum-of-squares (SoS) proof of the infeasibility of the system, i.e., a derivation of $-1 \geq 0$ from the axioms $\phi_1(x) = 0, \ldots, \phi_m(x) = 0$ and the domain constraints.*

- **NO**: *No such degree-d SoS proof exists.*

Let us call this problem "SoS-CSP". This is perhaps the most natural formulation of "the SoS algorithm for CSPs". It is quite striking that we still do not know whether there exists or not a polynomial-time decider for SoS-CSP (even for certain restricted classes of problems).

References to the related literature

O'Donnell [48] raised the issue of SoS bit complexity in 2017, as discussed in Section 1. O'Donnell also presented an example of a polynomial system with bounded coefficients that allows for a degree 2 SoS proof, which necessarily has doubly-exponential coefficients.

The aforementioned result (Theorem 1.4) by Raghavendra and Weitz [57] offered an initial elegant, albeit partial, solution. Raghavendra and Weitz expanded O'Donnell's work and presented an example of a polynomial system containing the Boolean constraints and a polynomial that admits degree 2 SoS proof, but for which any SoS proof of degree $O(\sqrt{n})$ must have coefficients of doubly-exponential magnitude in n.

Interestingly, Hakoniemi [31] demonstrated that both SoS and Polynomial Calculus (PC) refutations over Boolean variables encounter the same bit complexity issue. This finding also raises significant concerns regarding the frequently asserted degree automatability of PC.

Furthermore, strategies in [6, 15, 13, 45] to address the problem of SoS bit complexity involve replacing the original input polynomial constraints $\mathcal{P}$ with a new set of polynomials $\mathcal{P}^{(d)}$ that satisfies the NSATZ criterion, and generally depends on the SoS degree d. This set $\mathcal{P}^{(d)}$ is computed

externally (by an algorithm specifically designed for this purpose [2]), serving as the input for SoS in place of $\mathcal{P}$. For example, in the semilattice case, if $\mathcal{P}$ consists of m polynomials, the set $\mathcal{P}^{(d)}$, used in [15, 45], is generated by a specific algorithm and has a size of $m^{O(d)}$; that is, $\mathcal{P}^{(d)}$ depends on d and grows exponentially with the SoS degree d. This preprocessing step ensures that SoS retains "low" bit complexity, but only if $\mathcal{P}$ is substituted with $\mathcal{P}^{(d)}$. Essentially, the approach utilized in [6, 15, 13, 45] is to apply the NSATZ criterion without enhancing or extending it, with the goal of replacing the initial input polynomial system with a new one that is computed externally and satisfies the NSATZ criterion. Our results demonstrate that all preprocessing steps employed in [6, 15, 45] are unnecessary, as SoS achieves low bit complexity for any fixed d when $\mathcal{P}$ is provided directly as input.

Recently, Gribling et al. [27] showed that, under specific algebraic and geometric conditions, SoS relaxations can be computed in polynomial time. However, as they noted, their algebraic conditions are more restrictive than d-completeness. Their geometric conditions apply to systems of only inequality constraints with full dimensional feasibility set. Additionally, Palomba et al. [49] showed that, under some mild conditions, sum-of-squares bounds for copositive programs can be computed in polynomial time. These results also yielded insights into the bit complexity of SoS proofs.

The quest of characterizing conditions for ensuring tractability of the SoS proof system fits into the more general context of real algebraic geometry, and in particular of the so-called *effective Positivstellensatz*, i.e., the study of the complexity for representing polynomials using rational sums of squares. To this end, we mention Baldi et al. [1], who recently proved an exponential upper bound on the coefficients' bit complexity of sums of squares proofs in the general case of radical zero-dimensional ideals when the equality constraints form a graded basis. Furthermore, as observed, SoS feasibility can be reformulated as an SDP feasibility problem, which remains a well-known open question. In this context, we highlight the work of Pataki and Touzov [53], who showed that many SDP's have feasible sets whose elements have large encoding size. Moreover, they initiated the study of characterizing the conditions under which SDP feasibility sets with large encoding sizes occur.

Several papers in the literature investigate the automatability of the SoS proof system in relation to degree lower bounds. Specifically, various instances have been studied where a set of polynomial equations $\mathcal{P} = \{p_1 = 0, \ldots, p_m = 0\}$ and a polynomial q satisfy the condition that " $q \geq 0$ on S", but any SoS proof of this fact necessarily requires a high degree (see, e.g., [29, 38, 39, 40, 41, 55]). Within the context of the Lasserre hierarchy, these examples correspond to polynomial optimization problems for which many rounds of the hierarchy are needed to reach optimality.

In contrast, our focus is on a different aspect of automatability: we aim to understand under what conditions a fixed-degree level of the Lasserre hierarchy can be computed in time polynomial in the input size (up to a prescribed precision). This shifts the question from degree necessity to degree tractability within a computational framework.

1.1 Our contributions

Our main contribution is to study and expand the class of polynomial systems for which finding degree-d SoS proofs can be automated. To this end, we first introduce a new criterion based on the *Polynomial Calculus* (PC) proof system which guarantees that property (P) holds, referred to as the PC *criterion*. This criterion holds both for SoS refutations and for SoS proofs over feasible systems over finite domains. Remarkably, as we will demonstrate, this criterion applies to

---

[2]In general such an algorithm cannot be simulated by SoS. We defer the interested reader to Section 5.1 for details.

a broad class of instances arising from Constraint Satisfaction Problems (CSPs) where the NSATZ criterion does not. Specifically, we will establish tractability results for broad class of SoS-CSP, and, moreover, prove complete degree-d automatability beyond refutations for certain polynomial systems arising from CSP(Γ). The proof of the PC criterion combines several results, including a simulation of the PC proof system by the SoS proof system together with the development of a different criterion based on the SoS proof system, called the SoS$_\varepsilon$ criterion.

PC criterion

We begin by introducing *polynomial systems over finite domains*, i.e., systems where each variable is restricted to assume values from a fixed set of k rational numbers $\rho_1, \rho_2, \ldots, \rho_k$. Given a polynomial system of equations $\mathcal{P}$ and a finite domain $D = \{\rho_1, \ldots, \rho_k\}$, we say that $\mathcal{P}$ is a *polynomial system over finite domain D* if it includes the following *domain polynomials*:

$$D_k(x_i) = (x_i - \rho_1)(x_i - \rho_2) \cdots (x_i - \rho_k) = 0 \qquad i \in [n]. \tag{2}$$

These polynomials ensure that each variable $x_1, \ldots, x_n$ is constrained to take values from D. Note that this formulation generalizes polynomial systems with Boolean variables, i.e., where the constraint $x_i^2 - x_i = 0$ enforces x_i to be either 0 or 1.

Polynomial Calculus over $\mathbb{R}$ (PC/$\mathbb{R}$), or in short Polynomial Calculus (PC), is a proof system used in computational complexity and proof complexity to study the efficiency of algebraic reasoning. It operates over polynomials and is particularly useful in analyzing the complexity of solving systems of polynomial equations. The goal is to derive the polynomial 1 (i.e., show inconsistency) or to demonstrate that a certain polynomial is implied by the given system. Originally introduced as a *refutation system* in [19], PC can be viewed as a degree-truncated version of Buchberger's algorithm [36, 21]. Essentially, PC is a dynamic version of the NSATZ proof system, employing schematic inference rules to reason about polynomial equations. We emphasize that, for the remainder of the paper, we will consider PC in the broader sense of polynomial derivation (so not restricted to refutation) and with polynomials over the reals.

PC consists of the following derivation rules for polynomial equations $(f = 0), (g = 0) \in \mathcal{P}$, domain polynomial equations $(D_k(x_j) = 0)$, variable x_j, and numbers $a, b \in \mathbb{R}$

$$\frac{}{f = 0} \qquad \frac{}{D_k(x_j) = 0} \qquad \frac{f = 0 \qquad g = 0}{af + bg = 0} \qquad \frac{f = 0}{x_j f = 0} \tag{3}$$

A PC *derivation* of "$r = 0$" from $\mathcal{P}$ is a sequence $(r_1 = 0, \ldots, r_L = 0)$ of polynomial equations iteratively derived by using (3) with $r = r_L$. The *size* of a derivation is the sum of the sizes of the binary encoding of the polynomials in the derivation and the *degree* is the maximum degree of the polynomials in the derivation. A *PC refutation* is a derivation of "$1 = 0$".

Next, we present one of our main contributions, namely a framework for showing that Property (P) holds for certain polynomial systems over finite domains.

Theorem 1.6 (PC criterion)**.** *Let $\mathcal{P}$ be a polynomial system over a finite domain D of k rational values, let S be its variety. Let $\mathcal{G}_{2d}$ be a $2d$-truncated Gröbner basis of $I(S)$ according to the grlex order such that $\|\mathcal{G}_{2d}\|_\infty \leq 2^{poly(n^d)}$. Assume that, for every $g \in \mathcal{G}_{2d}$, there exist a PC derivation of g from $\mathcal{P}$ of size $poly(n^d)$ and degree $O(d)$.*

Let r be a polynomial and assume there exists an SoS proof of "$r \geq 0$" from $\mathcal{P}$ of degree $2d$. Then, for every $\varepsilon > 0$, there exists an SoS proof of "$r + \varepsilon \geq 0$" of degree $O(d)$ such that the absolute values of the coefficients of every polynomial appearing in the proof are bounded by $2^{poly(n^d, \lg \frac{1}{\varepsilon})}$.

Moreover, suppose $\mathcal{P}$ is NSATZ d-complete over S. It follows that, for every $g \in \mathcal{G}_{2d}$, the identity "$g = 0$" admits a degree-$O(d)$ NSATZ proof from $\mathcal{P}$. Further, PC is known to be strictly stronger than NSATZ [19], thus implying that our criterion is also more powerful than the NSATZ criterion. The separation between the NSATZ and the PC criteria is strict and it will be further discussed in Section 1.1.

The proof of Theorem 1.6 combines multiple techniques, which we will present in greater detail in Section 1.1. A key component of this proof is the development of a different criterion, the SoS_ε criterion. As we will see, this criterion provides a more general framework that extends beyond finite domains (see Example 3.19). Nevertheless, the strong connection between PC and Buchberger's algorithm makes the PC criterion an effective tool for many instances where the IMP_d can be efficiently solved [21].

Two main applications from CSPs.

In the following, we present our main two applications of Theorem 1.6. In both cases we focus on restricted classes of Constraint Satisfaction Problems (CSPs), denoted as $\text{CSP}(\Gamma)$, where constraints are limited to relations from a specified set Γ. These language restrictions have proven effective for analyzing computational complexity classifications and other algorithmic properties of CSPs, leading to recent breakthroughs in [11, 63, 64] (see, e.g., [3, 12, 18] and Section 5.2 for further details and necessary background).

First main application: refutation for bounded width CSPs All known tractable Constraint Satisfaction Problems $\text{CSP}(\Gamma)$ for a fixed constraint language Γ are solvable using two fundamental algorithmic principles. The first relies on the *few subpowers property* (see e.g. [3]). The second, *local consistency checking*, is the most widely known and natural approach for solving CSPs [2, 3, 11].

We consider the class of constraint languages Γ for which $\text{CSP}(\Gamma)$ has *bounded width*, meaning that it can be solved by a local consistency checking algorithm (see e.g. [2, 3]). Identifying and characterizing such languages is crucial for understanding the tractability of constraint satisfaction problems [2, 3]. Note that for languages that rely on the few subpowers property in general SoS requires high degree for refutation [3, 2, 28].

As a corollary of Theorem 1.6, we establish the polynomial-time feasibility of the SoS refutations for the whole class $\text{SoS-CSP}(\Gamma)$ problems (see Problem 1.5) for which $\text{CSP}(\Gamma)$ has *bounded width*.

Corollary 1.7. *For constraint languages Γ over finite domains for which $\text{CSP}(\Gamma)$ has* bounded width*, the $\text{SoS-CSP}(\Gamma)$ Problem 1.5 can be solved in polynomial time for any fixed degree d.*

Proof. TOPROVE 0 □

Note that both the decision and search versions of Problem 1.5 with bounded width are solvable in polynomial time for any fixed degree d, as a consequence of Theorem 1.6.

Further, we mention that in [60] it was obtained a similar result in the context of the Sherali-Adams proof system. However, this result applies only to a fixed limited form of $\mathcal{P}$, namely the Boolean canonical linear program. As remarked in Section 1, our focus is on deriving SoS proofs directly from $\mathcal{P}$ with variables over general finite domains without any preprocessing. To this end, in Corollary 1.7, we demonstrate that for *any* system of equations $\mathcal{P}$ over a finite domain that defines a bounded-width relation, finding SoS refutations directly derived from $\mathcal{P}$ can be automated.

Second main application: strong separations arising from CSPs As second application of Theorem 1.6, we examine constraint languages (and polynomial equations) that are closed under the semilattice and dual discriminator polymorphisms[3] (see, e.g., [3] and Sections 5.3 and 5.4 for the necessary background). Propositional formulas from HORN-SAT or 2-SAT can be easily translated into system of polynomial equations that are semillatice or dual discriminator closed, respectively. Moreover, these two classes extend HORN-SAT and 2-SAT formulas, respectively, to general finite domain cases and have held a significant role in the theory of Constraint Satisfaction Problems of the form CSP(Γ); see, e.g., [3, 12] and references therein.

Theorem 1.8. *For a system $\mathcal{P}$ of polynomial equations over n variables that is closed under the semilattice (or dual discriminator) polymorphism, then the PC criterion (Theorem 1.6) applies.*

Note that these classes of problems are known to be bounded width (see e.g. [3]). Therefore, by Corollary 1.7 the refutation Problem 1.5 can be solved in polynomial time. However, Theorem 1.8 establishes a significantly stronger result. Indeed, Theorem 1.8 indicates that *any* degree d SoS proof of $p \geq 0$, for any polynomial p, can be computed in $n^{O(d)}$ time with arbitrary precision (not only degree-d SoS proofs for $-1 \geq 0$, as required by refutation).

We emphasize that Buss and Pitassi [17] show that the NSATZ proof system necessitates a degree $\Theta(\log n)$ proof for the induction principle IND_n, a polynomial inference rule that can be formalized as a derivation in either HORN-SAT or 2-SAT formulae. As a result, if $\mathcal{P}$ is closed under the semilattice (or dual discriminator) polymorphism, it cannot be NSATZ d-complete for any $d = o(\log n)$. Thus, Theorem 1.6, Theorem 1.8 and [17] establish a clear separation between the PC criterion and the NSATZ criterion.

Polynomial Calculus (PC) is a rule-based, dynamic extension of Nullstellensatz (see e.g. [24]). Due to its dynamic nature, it can sometimes achieve a refutation of significantly lower degree through cancellations than would be possible with the static Nullstellensatz system. A notable example is the induction principle IND_n mentioned above, which has degree 2 refutations in PC. By contrast, its Nullstellensatz degree has been shown to be $\Theta(\log n)$ [17].

We demonstrate that PC, in addition to solving refutation for the very special case of IND_n with low degree, also addresses the much more general Ideal Membership Problem IMP_d in $n^{O(d)}$ time for two families of problems that significantly generalize HORN-SAT and 2-SAT in multiple respects and apply to all finite rational domains. This also demonstrates that PC is complete and free from bit complexity issues for these problems (Hakoniemi recently raised concerns regarding the bit complexity in PC [31], see also Section 8).

This result is not only intrinsically interesting but also closely aligned with the main goal of this article. Indeed, the PC criterion demonstrates that solvability via Polynomial Calculus and the bit complexity of Sum-of-Squares are deeply interconnected. Finally, we emphasize that it is not implied by the recent result of Bulatov and Rafiey [15]; more details are given in Section 5.

The proof of this broad generalization is technically complex and lengthy, necessitating a dedicated space with the necessary preliminaries. Therefore, we defer the full discussion—including a detailed review, proof, the underlying intuition and the literature review—to Section 5.

This paper aims to deepen our understanding of the bit complexity issue and to explore the conditions under which it arises. For instance, we demonstrate that all preprocessing steps aimed at replacing $\mathcal{P}$ with a new set $\mathcal{P}'$ to satisfy the NSATZ criterion, as used in [15, 45, 7, 6] to circumvent the bit complexity issues of SoS for semilattice and dual discriminator polymorphisms, are unnecessary.

---

[3]In the context of CSPs, a *polymorphism* is a special kind of function that helps us understand the structure of the constraints. Specifically, it is a function that combines multiple solutions of a CSP in a way that still satisfies the constraints. Polymorphisms are useful because they reveal patterns in the constraints, and studying them can help determine how easy or hard a CSP is to solve. We refer to Definition 5.2 for a formal definition.

Specifically, SoS, when applied directly to $\mathcal{P}$ as input, achieves low bit complexity for any fixed d (refer to Section 5.1 for a more detailed discussion). This result appears to support and extend the hypothesis that CNF formulas do not exhibit a bit complexity issue, an open question posed by Hakoniemi [31] (see also Section 8).

SoS (approximately) simulates PC

As a main contribution, we address the existing knowledge gap regarding the relationship between SoS and PC in the general finite domain setting. Our main result shows that SoS can simulate PC derivations in this setting with an arbitrarily small error. Essentially, if PC can derive the equation "$p = 0$", then, for any arbitrary $\varepsilon > 0$, SoS can prove the statements "$p + \varepsilon \geq 0$" and "$-p + \varepsilon \geq 0$" with only a polynomial increase in size. While this result serves as a main technique for proving the PC criterion, as outlined in Section 1.1, it is also valuable on its own, as we present below. Our result builds upon and generalizes the simulation result of Berkholz [5] for Boolean variables to the broader context of general finite domains.

Berkholz [5] related different approaches for proving the unsatisfiability of a system of real polynomial equations. Over Boolean variables, he showed that SoS simulates PC refutations: any PC refutation of degree d can be converted into an SoS refutation of degree $2d$, with only a polynomial increase in size.

In the non-Boolean setting, there are systems of equations that are easier to refute for PC than for SoS [30]. Grigoriev and Vorobjov [30] show that the simulation of PC by SoS does not hold in the non-Boolean case, namely when the Boolean axioms $x_j^2 - x_j = 0$ are omitted. For example, the so-called telescopic system of equations, $\{yx_1 = 1, x_1^2 = x_2, \ldots, x_{n-1}^2 = x_n, x_n = 0\}$, has a PC refutation of degree n, but it requires exponential refutation degree in SoS [30]. It is worth noting that a similar (although much weaker than the one present in Lemma 1.9) generalization of Berkholz's result was considered in [59], when the variables take the values ± 1, and in [52], for a variation of PC endowed with a "radical rule" and a "sum-of-squares rule".

Whether SoS could simulate PC in the general finite domain setting, where variables can take values from any finite set, has remained an open question, despite known limitations in specific non-Boolean cases.

In this work, we answer this open question and complement the results in [30, 5] by establishing the following theorem. In summary, we first derive the following lemma.

Lemma 1.9. *Let $\mathcal{P}$ be a system of polynomial equations over a finite domain D with $|D| = k$. Assume that "$r = 0$" has a PC derivation of degree d and size S from $\mathcal{P}$. Then "$-r^2 \geq 0$" has an SoS proof of degree $2(d + k - 1)$ with coefficients of size $poly(k, S)$.*

The overall structure of the proof partially mirrors that in [5], but with notable differences. In particular, new ideas and techniques are introduced in the simulation of the multiplication rule of PC.

Furthermore, note that the result holds for the particular case of refutations, i.e. when $r = 1$. Indeed, if there exists a PC refutation of $\mathcal{P}$, i.e., a derivation of $1 = 0$, then Lemma 1.9 implies that there exists an SoS refutation "$-1 \geq 0$" with only polynomial increasing.

Then, employing Lemma 1.9, we prove the following result.

Theorem 1.10. *SoS approximates PC with degree linear in the domain size k over general finite domains. That is, if there exists a PC derivation of "$r = 0$" with degree d and size S, then for every $\varepsilon > 0$, we have SoS proofs of "$r + \varepsilon \geq 0$" and "$-r + \varepsilon \geq 0$" with degree $O(d + k)$ and coefficients bounded by $2^{poly(k, S, \lg \frac{1}{\varepsilon})}$.*

Outline of the techniques for proving the PC criterion

Below we outline the main techniques that will be used in the proof of the PC criterion.

1. SoS_ε **criterion** (Section 3.1) We begin by introducing a general criterion, called the SoS_ε criterion, which ensures that Property (P) is satisfied and, consequently, that SoS can be automated. This criterion is a natural generalization of the NSATZ criterion [57] (see Theorem 1.4), and serves as a foundational tool for presenting our main contributions. The key distinction lies in the notion of approximate completeness: the SoS_ε criterion requires SoS_ε completeness, a relaxed condition than the one required by Theorem 1.4, as discussed in Section 3. Informally, a system $\mathcal{P}$ is SoS_ε complete if for every $q \in \mathrm{I}_d(S)$ and every $\varepsilon > 0$, there exist an SoS proof of the inequality "$q + \varepsilon \geq 0$" using bounded coefficients (see Section 3.3 for the precise definition).

2. SoS **approximability of polynomial systems**(Section 3.3) As previously mentioned, the main difference between the NSATZ and SoS_ε criteria lies in their respective notions of completeness. Therefore, a main challenge in applying the SoS_ε criterion is proving that a given system $\mathcal{P}$ is SoS_ε complete. To this end, in Section 3.3 we present the notion of SoS-approximation ($\lesssim$) between polynomial systems defining the same zero set, which turns out to be a powerful tool for showing SoS_ε completeness, and applying the SoS_ε criterion. The key advantage of SoS-approximation is that it allows the inheritance of SoS_ε completeness between systems: under mild conditions, if $\mathcal{P}$ is SoS_ε complete and $\mathcal{P} \lesssim \mathcal{Q}$, then $\mathcal{Q}$ is also SoS_ε complete.

3. **SoS approximately simulates PC** (Section 4.2) The final tool we use to establish the PC criterion is the simulation of the PC derivation system by SoS. As previously emphasized, this simulation, presented formally in Lemma 1.9 and Theorem 1.10, is one of the main contributions of this work and can be appreciated independently. However, it also plays a crucial role in proving another major result: the PC criterion (Theorem 1.6). The proof strategy proceeds as follows. Under the theorem's hypotheses, the truncated Gröbner basis $\mathcal{G}$ is easily shown to be NSATZ-complete and hence SoS_ε complete. Using our simulation results, we then establish the following chain of SoS-approximations:

$$\mathcal{G} \lesssim \mathcal{G}^2 \lesssim \mathcal{P}.$$

Finally, using the properties of SoS-approximability, we conclude that $\mathcal{P}$ is also SoS_ε complete.

In the following sections, we explore these three concepts in more detail. Then, in Section 4.3, we combine these techniques to prove the PC criterion (Theorem 1.6).

1.2 Structure of the article

In Section 3, we give a full exposition of the SoS_ε criterion. Subsequently, we develop several tools to facilitate its application. In Section 4, we focus on the case of polynomial systems over finite domains, where we establish a connection to the Polynomial Calculus (PC) proof system and derive a weaker version of the SoS_ε, called the PC criterion. The latter will be used for the separation in the radical ideal case. Sections 5 to 7 are devoted to construct a class of examples arising from Constraint Satisfaction Problems, demonstrating a separation between our new criterion and the NSATZ criterion. Section 5 begins with an overview of relevant background and related results,

followed by a description of our proof strategy and main results. In Sections 6 and 7, we present the detailed proofs. Finally, in Section 8, we finish with concluding remarks and discuss potential research directions.

2 Preliminaries

Consider a set of variables $\{x_1, \ldots, x_n\}$ and denote the vector space of polynomials in x up to a fixed degree $d = O(1)$ as $\mathbb{R}[x_1, \ldots, x_n]_d$. We denote as $\mathbf{v}_d$ being the column vector whose entries are the elements of the usual monomial basis of $\mathbb{R}[x_1, \ldots, x_n]_d$ and, if $\alpha \in \mathbb{R}^n$, $\mathbf{v}_d(\alpha)$ is the vector of reals whose entries are the entries of $\mathbf{v}_d$ evaluated at α. It follows that for any polynomial $u(x) \in \mathbb{R}[x_1, \ldots, x_n]_d$, it holds that $u(x) = u^T \mathbf{v}_d$ for some $u \in \mathbb{R}^n$. We will consider systems $\mathcal{P} = \{p_1 = 0, \ldots, p_m = 0\}$ of polynomial equations and an "input" polynomial r of degree at most d, with the (mild) assumption that the bit complexity needed to represent $\mathcal{P}$ and r is polynomial in n. The string l representing polynomials $r, p_1, \ldots, p_m$ will be the input for our certification problems and this last assumption allows us to reduce any complexity reasoning to n instead of the length of l. We will sometimes refer to $\mathcal{P}$ as a set of *constraints* or *axioms*.

Next, we need to define the measures of norm and bit complexity for different objects. For the first measure, consider a polynomial $p = \sum_\alpha c_\alpha x^\alpha$, we define $\|p\|_\infty = \max_\alpha |c_\alpha|$. Similarly, for a set of polynomial we have $\|\mathcal{P}\|_\infty = \max_{p \in \mathcal{P}} \|p\|_\infty$. Throughout this paper, we will assume that the set S of common zeros of $\mathcal{P}$ is *finite* and that $\|S\| := \max_{\alpha \in S} \|\alpha\| < 2^{poly(n^d)}$. These assumptions are very general and are met in many different contexts. For the second measure, consider a polynomial p (or a polynomial system $\mathcal{P}$). We define the *bit complexity* of p (or $\mathcal{P}$) as the minimum length of a bit-string representing p (or $\mathcal{P}$) when the rational numbers are represented with their reduced fractions written in binary (see e.g. [31]). As noted above, the bit complexity of $\mathcal{P}$ is assumed polynomial in n.

Explicit Archimedeanity

We recall the notion of explicit Archimedeanity of polynomial systems. This property plays a crucial role in the context of computations of SoS proofs [37, 43]. The Archimedean property in algebraic optimization requires all variables to be bounded within some compact set. The *explicitly Archimedean* condition strengthens this by demanding that boundedness of variables is efficiently certifiable via SoS proofs. We give the following formal definition.

Definition 2.1 (Explicitly Archimedean System). *A system of polynomials $\mathcal{P}$ is said to be* explicitly Archimedean *if one of the following equivalent conditions holds.*

- *For every degree k polynomial $p \in \mathbb{R}[x_1, \ldots, x_n]$, there exists $0 < N_p \leq 2^{poly(n^{\max\{k,d\}}, \lg \|p\|_\infty)}$ such that there exists a SoS proof of "$N_p - p \geq 0$" from $\mathcal{P}$ of degree $O(k)$ and with coefficients bounded by $2^{poly(n^{\max\{k,d\}}, \lg \|p\|_\infty)}$.*

- *For every $i \in [n]$ there exists $0 < N_{x_i} \leq 2^{poly(n^d)}$ such that there exist SoS proofs "$N_{x_i} - x_i \geq 0$" and "$N_{x_i} + x_i \geq 0$" from $\mathcal{P}$ of degree $O(d)$ and with coefficients bounded by $2^{poly(n^d)}$.*

- *For every $i \in [n]$ there exists $0 < N_{x_i^2} \leq 2^{poly(n^d)}$ such that there exist SoS proofs "$N_{x_i^2} - x_i^2 \geq 0$" from $\mathcal{P}$ of degree $O(d)$ and with coefficients bounded by $2^{poly(n^d)}$.*

The assumption of explicit Archimedeanity is met in numerous natural cases. This is the case for Boolean systems, i.e. systems that contain the Boolean constraints $x_i^2 - x_i = 0$ for every variable,

where it suffices to set $N_{x_i} = 1$ for $i \in [n]$. Furthermore, every system with variables constrained over a finite domain D is explicitly Archimedean, as shown in Lemma 4.3.

Ideals and varieties

Let us recall here the notions of (polynomial) ideals, (algebraic) varieties and some of their properties (see e.g. [21]). Consider a set of polynomials $\mathcal{P} = \{p_1, \ldots, p_m\} \subseteq \mathbb{R}[x_1, \ldots, x_n]$.

Definition 2.2. *The algebraic variety generated by $\mathcal{P}$ is defined as*

$$S := \boldsymbol{V}(\mathcal{P}) = \{x \in \mathbb{R}^n \,|\, p_i(x) = 0, \forall i \in [m]\}.$$

We also define the following two sets.

Definition 2.3.

$$\langle \mathcal{P} \rangle := \{q \in \mathbb{R}[x_1, \ldots, x_n] \mid q = \sum_i h_i p_i, \; h_i \in \mathbb{R}[x_1, \ldots, x_n]\},$$

$$\mathrm{I}(S) := \{q \in \mathbb{R}[x_1, \ldots, x_n] \mid q(\alpha) = 0, \; \forall \alpha \in S\},$$

as the ideal generated by $\mathcal{P}$, *the former, and the* (vanishing) ideal generated by set S, *the latter. We refer to a d-truncated ideal *when we consider* $I_d := I \cap \mathbb{R}[x_1, \ldots, x_n]_d$, *where* $I \subseteq \mathbb{R}[x_1, \ldots, x_n]$ *is a polynomial ideal.*

Definition 2.4. *We say that an ideal I is* radical *if $p^m \in \mathrm{I}$ for some $m \in \mathbb{N}$ implies $p \in \mathrm{I}$.*

Gröbner bases

We give here a very brief introduction on the notion of Gröbner basis. For a complete exposition, we refer the reader to [21].

We first establish an order on the polynomial ring $\mathbb{R}[x_1, \ldots, x_n]$. Given a monomial $x^\alpha = x_1^{\alpha_1} \ldots x_n^{\alpha_n}$, this can be unambiguously associated to the n-tuple $\alpha = (\alpha_1, \ldots, \alpha_n) \in \mathbb{N}^n$.

Definition 2.5. *Let $\alpha = (\alpha_1, \ldots, \alpha_n), \beta = (\beta_1, \ldots, \beta_n) \in \mathbb{N}^n$ and $|\alpha| = \sum_{i=1}^n \alpha_i$, $|\beta| = \sum_{i=1}^n \beta_i$.*

(i) *Lexicographic order (lex): We say $\alpha >_{\mathsf{lex}} \beta$ if, in the vector difference $\alpha - \beta \in \mathbb{Z}^n$, the left most nonzero entry is positive.*

(ii) *Graded lexicographic order (grlex): We say $\alpha >_{\mathsf{grlex}} \beta$ if $|\alpha| > |\beta|$, or $|\alpha| = |\beta|$ and $\alpha >_{\mathsf{lex}} \beta$.*

Throughout this paper we will always assume that $\mathbb{R}[x_1, \ldots, x_n]$ is ordered according to the *graded lexicographic order* grlex.

One potential approach to solving the IMP is through polynomial division. The idea is that, given a polynomial r and a set of polynomials $\mathcal{P}$, if the remainder of the division of r by $\mathcal{P}$ is zero, then r belongs to the ideal $\langle \mathcal{P} \rangle$. However, it is well-known that polynomial division is generally not well-defined. Specifically, the remainder resulting from the division of r by $\mathcal{P}$ can vary depending on the order in which the polynomials in $\mathcal{P}$ are used for division.

To fix this issue, a special set of generators was introduced in [9].

Definition 2.6 (Gröbner Basis). *Let $\mathcal{G} = \{g_1, \ldots, g_s\}$ be a set of polynomials. Consider an ideal $\mathrm{I} \subseteq \mathbb{R}[x_1, \ldots, x_n]$ such that $\mathrm{I} = \langle g_1, \ldots, g_s \rangle$, and consider $r \in \mathbb{R}[x_1, \ldots, x_n]$. We say that $\mathcal{G}$ is a Gröbner basis of I if the following property holds*

$$r \in \mathrm{I} \iff r|_{\mathcal{G}} = 0,$$

where $r|_{\mathcal{G}}$ is the remainder of the polynomial division of r by $\mathcal{G}$.

Moreover, we will be mainly interested in solving problems of the form IMP_d, i.e. when r has degree at most d. Because $\mathbb{R}[x_1, \ldots, x_n]$ is ordered according to the grlex order, the only polynomials in $\mathcal{G}$ that can divide r are those with degree d or lower. Consequently, we give the following definition.

Definition 2.7. *Let $\mathcal{G}$ be a Gröbner basis of an ideal in $\mathrm{I} \in \mathbb{R}[x_1, \ldots, x_n]$, the d-truncated Gröbner basis $\mathcal{G}_d$ of I is defined as*

$$\mathcal{G}_d := \mathcal{G} \cap \mathbb{R}[x_1, \ldots, x_n]_d. \tag{4}$$

By the definition of Gröbner basis, we immediately conclude that IMP_d can be solved when the d-truncated Gröbner basis is available. Specifically, we have

$$r \in \mathrm{I}_d \iff r|_{\mathcal{G}_d} = 0.$$

Furthermore, if $\mathcal{G}_d$ can be calculated in polynomial time, then the IMP_d can be solved in polynomial time (see also Appendix B).

Polynomial Calculus over general finite domains

In this paper we consider PC over a general finite domain D, which is an immediate generalization of the classical PC over the Boolean domain [19], i.e. when the Boolean constraints $x_j^2 - x_j = 0$ belong to the set of constraints. Let $D = \{\rho_1, \rho_2, \ldots, \rho_k\}$ be a finite domain. For every variable x_j where $j \in [n]$, to enforce x_j to assume values in D, we include the following univariate *domain polynomials* in $\mathcal{P}$

$$D_k(x_j) = (x_j - \rho_1)(x_j - \rho_2) \cdots (x_j - \rho_k) \qquad j \in [n].$$

PC over a general finite domain is a proof system that consists of the following derivation rules for polynomial equations $(f = 0), (g = 0) \in \mathcal{P}$, domain polynomial equations $(D_k(x_j) = 0)$, variable x_j, and numbers $a, b \in \mathbb{R}$

$$\frac{}{f = 0} \qquad \frac{}{D_k(x_j) = 0} \qquad \frac{f = 0 \qquad g = 0}{af + bg = 0} \qquad \frac{f = 0}{x_j f = 0} \tag{5}$$

Definition 2.8. *A PC derivation (or PC proof) of "$r = 0$" from $\mathcal{P}$ is a sequence $(r_1 = 0, \ldots, r_L = 0)$ of polynomial equations iteratively derived by using (5) with $r = r_L$. The size of a derivation is the sum of the sizes of the binary encoding of the polynomials in the derivation and the degree is the maximum degree of the polynomials in the derivation. A PC refutation is a derivation of "$1 = 0$".*

3 SoS_ε criterion

As outlined in Section 1, we will examine *sufficient* conditions on a polynomial system $\mathcal{P}$ for ensuring that the following property holds.

(P) Assume there exists an SoS proof of "$r \geq 0$" from $\mathcal{P}$ of degree $2d$. Then, for every $\varepsilon > 0$, there also exists an SoS proof of "$r + \varepsilon \geq 0$" from $\mathcal{P}$ with degree $O(d)$ and coefficients bounded by $2^{poly(n^d, \lg \frac{1}{\varepsilon})}$.

14

In this section we formulate the SoS_ε *criterion*, a set of sufficient conditions that guarantee that property (P) holds (see Section 3.1, in particular Theorem 3.4). The SoS_ε criterion has two requirements: δ-spectrality and SoS_ε-completeness. We then proceed to give general settings and techniques to verify that the requirements are satisfied (see Section 3.2 and Section 3.3). Finally, we discuss the separation between the NSATZ criterion and the SoS criterion (see Sections 3.4, 4.3 and 5).

3.1 SoS_ε criterion

Recall we are assuming that S is finite and that $\|S\| < 2^{poly(n)}$. The *moment* matrix is defined as follows.

$$M = M_{S,d} := \mathbb{E}_{\alpha \in S}[\mathbf{v}_d(\alpha)\mathbf{v}_d^\mathsf{T}(\alpha)] = \frac{1}{|S|}\sum_{\alpha \in S} \mathbf{v}_d(\alpha)\mathbf{v}_d^\mathsf{T}(\alpha), \tag{6}$$

where the expectation is over the uniform distribution over S. Note that M is positive semidefinite, i.e. it is a real symmetric matrix with nonnegative eigenvalues. Let $\lambda_1, \lambda_2, \ldots, \lambda_{\binom{n+d}{d}}$ be the eigenvalues of $M_{s,d}$ with corresponding eigenvectors $u_1, \ldots, u_{\binom{n+d}{d}}$ forming an orthonormal basis for $\mathbb{R}^{\binom{n+d}{d}}$. Let U be the matrix where the columns are the eigenvectors $u_1, \ldots, u_{\binom{n+d}{d}}$. Now, we define

$$\Pi^+ := \sum_{i \text{ s.t. } \lambda_i > 0} u_i u_i^\mathsf{T}, \qquad \Pi^0 := \sum_{i \text{ s.t. } \lambda_i = 0} u_i u_i^\mathsf{T}. \tag{7}$$

Then, we have

$$I = U^\mathsf{T} U \qquad \text{and} \qquad I = \Pi^+ + \Pi^0.$$

We have the following lemma.

Lemma 3.1. *Let u be an eigenvector for the zero eigenvalue $\lambda = 0$ of M. Then, we have $u^\mathsf{T}\mathbf{v}_d \in \mathrm{I}_d(S)$.*

Proof. TOPROVE 1 $\qquad\qquad\square$

Definition 3.2 (δ-spectrality and SoS_ε-completeness). *Let $\mathcal{P} = \{p_1 = 0, \ldots p_m = 0\}$ be polynomial system with variety $S = \boldsymbol{V}(\mathcal{P})$, and let $\delta \in \mathbb{R}_{>0}$.*

1. *We say that S is δ-spectrally rich up to degree d if every nonzero eigenvalue of M is at least δ.*

2. *We say that $\mathcal{P}$ is SoS_ε-d-complete over S (or simply SoS_ε-complete) if, for every polynomial in the 2d-truncated vanishing ideal $q \in \mathrm{I}_{2d}(S)$ and every $\varepsilon > 0$, there exists an SoS proof of "$q + \varepsilon \geq 0$" of degree $O(d)$ from $\mathcal{P}$ with absolute value of the coefficients bounded by $2^{poly(n^d, \lg\|q\|_\infty, \lg\frac{1}{\varepsilon})}$.*

Let r be a polynomial. Suppose that "$r \geq 0$" admits an SoS proof Π from $\mathcal{P}$. Although Π may, in principle, have coefficients with magnitude of the order 2^{2^n} (see [48, 57]), we show in the next result that r can be decomposed as a sum-of-squares component plus an "ideal part" component, both having bounded coefficients. In general, the ideal part does not necessarily take the form $\sum h_i p_i$ as in Definition 1.1. Nevertheless, this result allows us to reduce the discussion to focus on the ideal part of the decomposition. The following lemma, essentially from Raghavendra and Weitz [57], is presented here separately as we use it to extend their result.

Lemma 3.3. *Consider the a polynomial system $\mathcal{P} = \{p_1 = 0, \ldots, p_m = 0\}$ with finite variety $S = \mathbf{V}(\mathcal{P})$ such that $\|S\| \leq 2^{poly(n^d)}$. Assume that S is δ-spectrally rich up to degree d.*
Let r be a polynomial nonnegative on S with coefficients bounded by $2^{poly(n^d)}$. If there exists an SoS *proof of "$r \geq 0$" from $\mathcal{P}$ with degree at most $2d$*

$$r = \sum_{i=1}^{t_0} q_i^2 + \sum_{i=1}^{m} h_i p_i, \tag{8}$$

for some $s_i, h_i \in \mathbb{R}[x_1, \ldots, x_n]$. Then, we have

$$r = \sum_{i=1}^{t_1} s_i^2 + p, \tag{9}$$

for some polynomials s_i of degree at most d, some $p \in \mathrm{I}_{2d}(S)$ and with all the coefficients on the right-hand side of (9) *bounded by $2^{poly(n^d, \lg \frac{1}{\delta})}$.*

Proof. TOPROVE 3 □

Note that if $\mathcal{P}$ is Nsatz d-complete, then the identity $p = \sum h_i p_i$ for the "ideal part" can be computed efficiently by the Nsatz proof system, i.e. with degree at most $O(d)$ and coefficients bounded by $2^{poly(n^d, \lg \frac{1}{\delta})}$. This is the idea behind the Nsatz criterion. However, this is sufficient but not a necessary condition (see [57] for a further discussion on the limitations of the Nsatz criterion).

We next present a new criterion called SoS_ε *criterion*. In essence, the SoS_ε criterion requires that any degree $2d$ polynomial from the ideal part can be SoS proven efficiently to be nonnegative (up to an additive error ε). This replaces the requirement that there exists Nsatz proofs of the the ideal part in the Nsatz criterion. Since SoS is stronger than Nsatz as a proof system, it follows that the SoS_ε criterion extends and generalizes the Nsatz criterion. In the following, we will provide natural examples of separation between the two criteria (see Section 3.4).

Theorem 3.4 (SoS_ε criterion). *Consider a polynomial system $\mathcal{P} = \{p_1 = 0, \ldots, p_m = 0\}$ with finite variety $S = \mathbf{V}(\mathcal{P})$ such that $\|S\| \leq 2^{poly(n^d)}$. Assume that (see Definition 3.2)*

1) S is δ-spectrally rich up to degree d, and

2) $\mathcal{P}$ is SoS_ε-complete over S.

Let r be a polynomial. If "$r \geq 0$" has a degree $2d$ SoS *proof*

$$r = \sum_{i=1}^{t_0} \sigma_i^2 + \sum_{i=1}^{m} h_i p_i,$$

then, for every $\varepsilon > 0$, there exists an SoS *proof of "$r + \varepsilon \geq 0$" of degree $O(d)$*

$$r + \varepsilon = \sum_{i=1}^{t} \tilde{\sigma}_i^2 + \sum_{i=1}^{m} \tilde{h}_i p_i, \tag{10}$$

such that the coefficients of every polynomial appearing in the proof are bounded by $2^{poly(n^d, \lg \frac{1}{\delta}, \lg \frac{1}{\varepsilon})}$.

Proof. TOPROVE 4 □

Corollary 3.5. *Suppose $\mathcal{P}$ satisfies the assumptions of the SoS_ε criterion with $\frac{1}{\delta} = 2^{poly(n^d)}$. Assume, moreover, that $\mathcal{P}$ is explicitly Archimedean. If "$r \geq 0$" has an* SoS *proof of degree $2d$ from $\mathcal{P}$, then "$r + \varepsilon \geq 0$" has an* SoS *proof of degree $O(d)$ from $\mathcal{P}$ that can be computed in time $poly(n^d, \lg \frac{1}{\varepsilon})$, up to any additive error ε.*

3.2 δ-spectrality

The δ-spectrality (Definition 3.2) hypothesis in Theorem 3.4 is, to some extent, a mild hypothesis. It is satisfied by many interesting instances where the variety S is discrete. For example, it is satisfied by combinatorial problems having varieties contained in the Boolean hypercube $\{0,1\}^n$. To see this, we first state a lemma for integer-valued matrices.

Lemma 3.6 ([57])**.** *Let $M \in \mathbb{S}^{N \times N}$ be an integer matrix with $|M_{ij}| \leq B$ for all $i, j \in [N]$. The smallest non-zero eigenvalue of M is at least $(BN)^{-N}$.*

By observing that $|S| \cdot M_{S,d}$ is a $O(n^d) \times O(n^d)$ integer-valued matrix, we immediately get δ-spectrality over integer-valued varieties.

Corollary 3.7 ([57])**.** *Let $\mathcal{P}$ be a polynomial system such that $S \subseteq \mathbb{Z}^n$ such that $\|S\| < 2^{poly(n^d)}$. Then S is δ-spectrally rich with $\frac{1}{\delta} = 2^{poly(n^d)}$.*

Another wide class of polynomial problems for which δ-spectrality is easily satisfied are the polynomial systems $\mathcal{P}$ with *variables constrained over a finite domain D*. Assuming $D = \{\rho_1, \ldots, \rho_k\} \subseteq \mathbb{Q}$ with constant $k = O(1)$, these systems are described as containing the domain polynomials $(x_i - \rho_1)(x_i - \rho_2) \cdots \cdots (x_i - \rho_k)$ for each variables x_i.

Corollary 3.8. *Let $\mathcal{P}$ be a polynomial system with variables constrained over a finite domain $D \subseteq \mathbb{Q}$, i.e. $S \subseteq D^n$. Then S is δ-spectrally rich (up to degree d) for some $\frac{1}{\delta} = 2^{poly(n^d)}$*

Proof. TOPROVE 5 □

3.3 SoS_ε-completeness

In this section we develop tools for showing that a polynomial system is SOS_ε-complete. We will consider multiple polynomial systems $\mathcal{Q}$ preserving geometric and bit complexity properties of $\mathcal{P}$, namely

A1. *Same zero set:* $S = \mathbf{V}(\mathcal{P}) = \mathbf{V}(\mathcal{Q})$.

A2. *Same degree order:* $\deg(q) = O(d), \forall q \in \mathcal{Q}$, where d is the maximum degree of the polynomials in $\mathcal{P}$.

A3. *Polynomial bit complexity:* the bit complexity for representing $\mathcal{Q}$ is polynomial in n. Note that this implies that the cardinality of $\mathcal{Q}$ is polynomially bounded, i.e. $|\mathcal{Q}| = poly(n)$, and that all coefficients of the polynomials in $\mathcal{Q}$ are bounded by $2^{poly(n^d)}$.

SoS-approximability. We define the relation of SoS-*approximability* between polynomial systems with the same zero set. This relation arises by considering SoS proofs of approximate objective polynomials $p + \varepsilon$. Therefore, it cannot be simulated by the NSATZ proof system. We will see that SoS-approximability is a powerful tool for showing that a polynomial system $\mathcal{P}$ is SOS_ε-complete.

Definition 3.9 (SoS approximation)**.** *Let $\mathcal{P} = \{p_1 = 0, p_2 = 0, \ldots, p_m = 0\}$ and $\mathcal{P}' = \{p'_1 = 0, p'_2 = 0, \ldots, p'_l = 0\}$ be two polynomial systems such that $\mathbf{V}(\mathcal{P}) = \mathbf{V}(\mathcal{P}')$. We say that $\mathcal{P}'$ SoS-approximates $\mathcal{P}$, and it is denoted by $\mathcal{P} \lesssim_{SoS} \mathcal{P}'$, if for every $p \in \mathcal{P}$ and every $\varepsilon > 0$ there exist SoS proofs*

$$
\begin{aligned}
&\text{``}p + \varepsilon \geq 0\text{'' from } \mathcal{P}' \text{ and} \\
&\text{``} - p + \varepsilon \geq 0\text{'' from } \mathcal{P}'
\end{aligned}
\tag{11}
$$

with degree $O(d)$ and with coefficients bounded by $2^{poly(n^d, \lg \frac{1}{\varepsilon})}$.

Next, we introduce a property that will prove valuable throughout the rest of this section. Roughly speaking, we will show that, under the assumption of explicit Archimedeanity, if SoS can (approximately) prove "$p = 0$" in a precise sense, then it can (approximately) prove the product "$gp \geq 0$" for any polynomial g.

Lemma 3.10. *Let $\mathcal{P}$ be an explicitly Archimedean polynomial system. Let $p \in \mathbb{R}[x_1, \ldots, x_n]$ be a polynomial of degree (at most) $2d$ with coefficient norm bounded by $2^{poly(n^d)}$. Assume that, for every $\varepsilon > 0$, we have SoS proofs of degree $2d$ from $\mathcal{P}$ of*

$$
\begin{aligned}
&\text{``}p + \varepsilon \geq 0\text{'', and of} \\
&\text{``} - p + \varepsilon \geq 0\text{'',}
\end{aligned}
\tag{12}
$$

with coefficients bounded by $2^{poly(n^d, \lg \frac{1}{\varepsilon})}$. Then, for every $\varepsilon > 0$ and every $g \in \mathbb{R}[x_1 \ldots, x_n]$ with $\deg(g) = O(d)$ and $\|g\|_\infty < 2^{poly(n^d)}$, there exists an SoS proof from $\mathcal{P}$ of

$$\text{``}pg + \varepsilon \geq 0\text{'',}$$

of degree $O(d)$ with coefficients bounded by $2^{poly(n^d, \lg \frac{1}{\varepsilon})}$.

Proof. TOPROVE 6 □

With the notions of SoS-approximability and Lemma 3.10 at hand, we begin by demonstrating an interesting property of the relation $\lesssim_{\text{SoS}}$.

Lemma 3.11 (Transitivity). *Let $\mathcal{P}_1, \mathcal{P}_2$ and $\mathcal{P}_3$ be three systems of polynomials with zero set S. Assume that $\mathcal{P}_3$ is explicitly Archimedean. If $\mathcal{P}_1 \lesssim_{SoS} \mathcal{P}_2$ and $\mathcal{P}_2 \lesssim_{SoS} \mathcal{P}_3$, then $\mathcal{P}_1 \lesssim_{SoS} \mathcal{P}_3$.*

Proof. TOPROVE 7 □

Next we present a few relevant examples for which it is possible to show SoS-approximability. The first example focuses on the powers of polynomial systems. Namely, let $\mathcal{P} = \{p_1 = 0, \ldots, p_m = 0\}$ be a polynomial system of equations. We define the α-*power of* $\mathcal{P}$ as the polynomial system $\mathcal{P}^\alpha = \{p_1^{\alpha_1} = 0, p_2^{\alpha_2} = 0, \ldots, p_m^{\alpha_m} = 0\}$, where α is a multi-index $\alpha = (\alpha_1, \alpha_2, \ldots, \alpha_m) \in \mathbb{N}^m$. The next result shows that α-powers of a polynomial system approximate the set itself.

Proposition 3.12. *Let $\alpha \in \mathbb{N}^n$, with $|\alpha| = O(d)$. Then, $\mathcal{P} \lesssim_{SoS} \mathcal{P}^\alpha$.*

Proof. TOPROVE 8 □

In the second example, we show that when the polynomials in a system of polynomials are multiplied by positively shifted sums-of-squares, approximation is possible. More precisely, let $\mathcal{P} = \{p_1 = 0, \ldots, p_m = 0\}$ be a polynomial system. Let $g \in \mathbb{R}[x_1, \ldots, x_n]$ be a polynomial of the form

$$g = \sum_{i=1}^{t} q_i^2 + c, \tag{13}$$

for some constant $c > 0$. We consider the polynomial system $\mathcal{P}' = \{gp_1 = 0, \ldots, p_m = 0\}$. Clearly, $\mathcal{P}$ and $\mathcal{P}'$ have the same variety. We also make the assumptions A2. and A3. for set $\mathcal{P}'$. We have the following result.

Proposition 3.13. *Let $\mathcal{P}$ and $\mathcal{P}'$ as defined above. Then, we have $\mathcal{P} \lesssim_{SoS} \mathcal{P}'$.*

Proof. TOPROVE 9 □

Showing SOS_ε-completeness. The main consequence of the concept of SoS-approximability is that it allows for the inheritance of SOS_ε-completeness among different polynomial systems.

Theorem 3.14. *Let $\mathcal{P}_1$ and $\mathcal{P}_2$ be polynomials systems with zero set S. Assume that $\mathcal{P}_1$ is SOS_ε-complete and that $\mathcal{P}_2$ is explicitly Archimedean. If $\mathcal{P}_1 \lesssim_{SoS} \mathcal{P}_2$, then $\mathcal{P}_2$ is SOS_ε-complete.*

Proof. TOPROVE 10 $\square$

Corollary 3.15. *Let $\mathcal{P}_1, \ldots, \mathcal{P}_k$ be polynomial systems for some integer $k = O(1)$. Assume that $\mathcal{P}_1 \lesssim_{SoS} \cdots \lesssim_{SoS} \mathcal{P}_k$. If $\mathcal{P}_1$ is SOS_ε-complete and $\mathcal{P}_k$ is explicitly Archimedean, then $\mathcal{P}_k$ is SOS_ε-complete.*

It follows that the problem of showing that an explicitly Archimedean system $\mathcal{P}$ is SOS_ε-complete can be reduced to identifying SOS_ε-complete polynomial systems $\mathcal{Q}$ such that $\mathcal{Q} \lesssim_{SoS} \mathcal{P}$. Interestingly, this can be achieved in various instances.

A broad class of such reductions arises from Gröbner basis theory. We recall that Gröbner bases completely characterize polynomial ideals. Specifically, for polynomial rings ordered by the *grlex* order, every polynomial in the $2d$-truncated ideal $q \in I_{2d}$ has remainder 0 when reduced by the set $\mathcal{G}_{2d}$ of elements of degree at most $2d$ of a Gröbner basis $\mathcal{G}$ (see e.g. [21]). Therefore, we have the following result.

Lemma 3.16. *Let $\mathcal{P}$ be a polynomial system with $S = \boldsymbol{V}(\mathcal{P})$. Let $\mathcal{G}_{2d}$ be a $2d$-truncated Gröbner basis of $I(S)$ according to the grlex order. Assume that $\|\mathcal{G}_{2d}\|_\infty \leq 2^{poly(n^d)}$. Then $\mathcal{G}_{2d}$ is SOS_ε-complete.*

Proof. TOPROVE 11 $\square$

Finally, we obtain a method for checking whether a system $\mathcal{P}$ is SOS_ε-complete.

Corollary 3.17. *Let $\mathcal{P}$ be an explicitly Archimedean polynomial system and let $\mathcal{G}_{2d}$ be a $2d$-truncated Gröbner basis of $\text{I}(\boldsymbol{V}(\mathcal{P}))$ according to the grlex order. Assume that $\|\mathcal{G}_{2d}\|_\infty \leq 2^{poly(n^d)}$. If there exists a multi-index α with $|\alpha| = O(d)$ such that $\mathcal{G}^\alpha \lesssim_{\text{SOS}} \mathcal{P}$, then $\mathcal{P}$ is SOS_ε-complete.*

Proof. TOPROVE 12 $\square$

Furthermore, the relation $\lesssim_{SoS}$ induces an order structure over the set of explicitly Archimedean polynomials systems with the same variety.

Proposition 3.18. *Consider the set*

$$\mathfrak{P}_S = \{\mathcal{P} \,|\, \mathcal{P} \text{ is explicitly Archimedean with } \boldsymbol{V}(\mathcal{P}) = S\}.$$

Then $\lesssim_{SoS}$ is a preorder *(i.e. a reflexive and transitive relation) of the set $\mathfrak{P}_S$.*

3.4 Separation between Nsatz and SoS criteria

We now highlight some differences distinguishing the two notions of completeness. We distinguish between two fundamental cases: non-radical and radical ideals (see Definition 2.4).

Non-radical ideals. When the ideal generated by the input polynomials is not radical, the NSATZ criterion is inherently weak and does not apply, as the d-completeness property cannot be satisfied. In contrast, below we show that our criterion is more robust and it may be effective even for non-radical ideals. In the following, we show a concrete example of an application of the SOS_ε criterion for polynomial systems with non-radical ideals.

Example 3.19. *Let $\mathcal{P} = \{x_1^2 + x_2^2 + \ldots + x_n^2 = 0\}$. We show first that $\mathcal{P}$ is explicitly Archimedean. It suffices to show that $\mathcal{P}$ has bounded variables by Definition 2.1. Indeed, let $i \in [n]$ and consider a variable x_i. We have that*

$$\frac{1}{4} - x_i = \left(\frac{1}{2} - x_i\right)^2 + x_2^2 + \cdots + x_n^2 - (x_1^2 + x_2^2 + \cdots + x_n^2),$$

$$\frac{1}{4} + x_i = \left(\frac{1}{2} + x_i\right)^2 + x_2^2 + \cdots + x_n^2 - (x_1^2 + x_2^2 + \cdots + x_n^2).$$

Hence, $\mathcal{P}$ is explicitly Archimedean.

Observe now that $\boldsymbol{V}(\mathcal{P}) = \{(0, 0 \ldots, 0)\}$. Therefore, the reduced Gröbner Basis for $I(\boldsymbol{V}(\mathcal{P}))$ is given by $\mathcal{G} = \{x_1, x_2, \ldots, x_n\}$. Next we observe that $I(\boldsymbol{V}(\mathcal{P}))$ is not radical. Indeed, that there are no NSATZ proofs of polynomials x_i (for every $i \in [n]$) from $\mathcal{P}$ since the polynomial $h(x_1^2 + \cdots + x_n^2)$ is the zero polynomial or it has degree at least 2, for every $h \in \mathbb{R}[x_1, \ldots, x_n]$. Hence the NSATZ criterion cannot be applied to $\mathcal{P}$. However, it is still possible to find SoS proofs of "$-x_i^2 \geq 0$" and "$x_i^2 \geq 0$" from $\mathcal{P}$. Thus, by definition, $\mathcal{G}^2 = \{x_1^2, \ldots, x_n^2\} \lesssim_{SoS} \mathcal{P}$. Also, by Proposition 3.12, we have $\mathcal{G} \lesssim_{SoS} \mathcal{G}^2$ and by Lemma 3.16 we have that $\mathcal{G}$ is SOS_ε-complete. Thus by Corollary 3.15 we have that $\mathcal{P}$ is SOS_ε-complete.

Lastly, we note that the moment matrix is

$$(M_{\boldsymbol{V}(\mathcal{P}),d})_{ij} = \begin{cases} 1 & \text{for } (i,j) = (1,1), \\ 0 & \text{otherwise,} \end{cases}$$

thus $\boldsymbol{V}(\mathcal{P})$ is 1-spectrally rich. Therefore, the SOS_ε criterion of Theorem 3.4 applies, i.e. for every polynomial r with a degree $2d$ SoS proof from $\mathcal{P}$, there exists also a proof of "$r + \varepsilon \geq 0$" of degree $O(d)$ and coefficients bounded by $2^{poly(n^d, \lg \frac{1}{\varepsilon})}$ for any additive error $\varepsilon > 0$.

Radical ideals. The previous separation example show advantages of the SOS_ε criterion over the NSATZ criterion. But what happens in the case of radical ideals?

For problems with a finite domain, the ideal is radical, and it is well known that the NSATZ proof system is complete for sufficiently large degrees. However, in general, a linear lower bound on the degree $O(n)$ is unavoidable, in the sense that there are instances of systems $\mathcal{P}$ and polynomials r such that for proving that "$r = 0$" from $\mathcal{P}$ by the NSATZ proof system there is a lower bound $\Omega(n)$ on the degree [16]. Thus, if the degree is bounded by a constant d, both the NSATZ and the SoS proof systems are again incomplete, even though we are in a radical setting. We address whether a separation can be established between the SOS_ε criterion and the NSATZ criterion in this context. Specifically, we ask whether there exists a polynomial system $\mathcal{P}$ and a polynomial r such that any NSATZ proof of "$r = 0$" from $\mathcal{P}$ necessarily has a degree that depends non-constantly on n, while, for every additive error $\varepsilon > 0$, SoS proofs of "$r + \varepsilon \geq 0$" and "$-r + \varepsilon \geq 0$" from $\mathcal{P}$ can be achieved with degree $O(d)$ and coefficients bounded by $2^{poly(n^d)}$. We affirmatively answer this question.

To do so, in Section 5 we will examine two natural families of problems, which have played a crucial role in the theory of $\mathrm{CSP}(\Gamma)$. In these cases, the ideal is radical because the variables take values from a finite domain. For these families, we show a strict separation between the two criteria.

Role of ε in the criteria. We highlight that our criterion asks for an *approximated* proof of the nonnegativity of the elements in the truncated ideal $I_{2d}(S)$. This seemingly subtle difference has a significant impact on the application of the criterion. Indeed, Example 3.19 also shows that even if for some $q \in I_{2d}(S)$ there is no SoS proof for "$q \geq 0$", there may be one for "$q + \varepsilon \geq 0$" satisfying the criterion conditions. This shows that not only replacing the proof system with stronger one

(NSATZ with SoS) plays a role, but also extending it to an approximate form. We further note that allowing this approximation in the condition has an impact in the result of the criterion. Whereas the NSATZ criterion shows the existence of an SoS proof of "$r \geq 0$" with bounded coefficients, the SoS_ε criterion guarantees the existence of a proof of "$r + \varepsilon \geq 0$" with bounded coefficients. However, following the discussion in the introduction, this second property is enough to guarantee the polynomial-time computability (up to arbitrary precision) and essentially does not compromise the quality of the computed solutions.

3.5 The semialgebraic case

We have seen the SoS_ε criterion in the algebraic case with finite S. As noted, this setting is very general and covers a wide range of combinatorial problems. Moreover, all the separations we present in this paper are in this case. Nonetheless, it is not hard to generalize the SoS_ε criterion to the case where S is *infinite* and there are inequality constraints. Indeed, Raghavendra and Weitz [57] originally formulated the NSATZ criterion in this more general setting. For completeness of exposition, we proceed to formulate the SoS_ε criterion in its full generality. The proof in the semialgebraic case, is similar, mutatis-mutandis, to the proof of the SoS_ε criterion (see Section 3.1).

We begin by defining the SoS proof system in this setting.

Definition 3.20. *Let $\mathcal{P} = \{p_1 = 0, \ldots, p_m = 0\}$ be a set of polynomial equality constraints and $\mathcal{Q} = \{q_1 \geq 0, \ldots, q_\ell \geq 0\}$ be a set of polynomial inequality constraints. Consider a polynomial $r \in \mathbb{R}[x_1, \ldots, x_n]$. An SoS proof of "$r \geq 0$" (over S) from $(\mathcal{P}, \mathcal{Q})$ is an identity of the form*

$$r = \sum_{i=1}^{t_0} s_i^2 + \sum_{i=1}^{\ell} \left(\sum_{j=1}^{t_i} \lambda_j^2 \right) q_i + \sum_{i=1}^{m} h_i p_i,$$

where $s_i, \lambda_j, h_i \in \mathbb{R}[x_1, \ldots, x_n]$. Moreover, we say that the above SoS proof has degree at most d if $\max\{\deg(s_i^2), \deg(\lambda_j^2 q_i), \deg(h_i p_i)\} \leq d$.

Next, we "adjust" various definitions to the semialgebraic case.

Definition 3.21. *Let $\mathcal{P} = \{p_1 = 0, \ldots, p_m = 0\}$ be a set of polynomial equality constraints and $\mathcal{Q} = \{q_1 \geq 0, \ldots, q_\ell \geq 0\}$ be a set of polynomial inequality constraints. We define as*

$$S = \{x \in \mathbb{R}^n \mid p_1(x) = \ldots = p_m(x) = 0, \ q_1(x), \ldots, q_\ell(x) \geq 0\}$$

as the feasibility set (or zero set) of $(\mathcal{P}, \mathcal{Q})$. Moreover, we recall that the moment matrix is defined as $M = M_{S,d} = \mathbb{E}_{\alpha \in S}[\mathbf{v}_d(\alpha)\mathbf{v}_d^\top(\alpha)]$, where the expectation is taken over the uniform distribution over S.

Lastly, we introduce a new notion to relates to the set of inequality constraints $\mathcal{Q}$

Definition 3.22. *We say that S is μ-robust for $\mathcal{Q}$ if for all $q \in \mathcal{Q}$ and all $\alpha \in S$, it holds that $q(\alpha) > \mu$.*

We are ready to state the SoS_ε in the semialgebraic case.

Theorem 3.23 (SoS_ε criterion)**.** *Let $\mathcal{P} = \{p_1 = 0, \ldots, p_m = 0\}$ be a set of polynomial equality constraints and $\mathcal{Q} = \{q_1 \geq 0, \ldots, q_\ell \geq 0\}$ be a set of polynomial inequality constraints, with feasibility set S such that $\|S\| \leq 2^{poly(n^d)}$. Assume that*

1) S is δ-spectrally rich up to degree d,

2) $\mathcal{P}$ is SoS_ε-complete over S,

3) S is μ-robust for $\mathcal{Q}$.

Let r be a polynomial. If "$r \geq 0$" has a degree $2d$ SoS proof from $(\mathcal{P}, \mathcal{Q})$ then, for every $\varepsilon > 0$, there exists an SoS proof of "$r + \varepsilon \geq 0$" of degree $O(d)$ from $(\mathcal{P}, \mathcal{Q})$ such that the coefficients of every polynomial appearing in the proof are bounded by $2^{poly(n^d, \lg \frac{1}{\delta}, \lg \frac{1}{\mu} \lg \frac{1}{\varepsilon})}$.

4 SoS and PC for polynomials over finite domains

This section provides a complete exposition of the main technical results concerning the automatability of degree-d SoS proofs. It presents two primary results: the PC criterion, a sufficient condition for automatability based on the PC proof system, and the approximate simulation of PC by SoS over finite domains. The simulation result is used in the proof of the PC criterion and may be of independent interest.

We begin by formally defining polynomial systems over finite domains. These are systems of polynomials where variables are restricted to take values over finite sets. Following this, we present Theorem 4.8, which states that SoS approximately simulates PC in this finite domain setting. This result builds upon on Lemma 4.5; its relation to prior work and its role in proving the main criterion are discussed. Finally, the section concludes with the presentation of the PC criterion (Theorem 4.9).

4.1 Finite domains systems

Consider a system of real polynomial equations

$$\mathcal{F} = \{f_1 = 0, \ldots, f_m = 0\} \tag{14}$$

over variables $x_1, \ldots, x_n$. We consider the general case of polynomial equations over a finite domain D of even size $2k$, with $k \in \mathbb{N}$, namely every variable can take a value from among $2k$ given rational values $\rho_1, \rho_2, \ldots, \rho_{2k}$. If the domain has an odd number of (distinct) elements, repeat an element so the resulting domain has an even number of (not distinct) elements. We define the univariate rational *domain polynomials* $D_{2k}(x_j)$ of degree $2k$ for each variable x_j as follows:

$$D_{2k}(x_j) = (x_j - \rho_1)(x_j - \rho_2)\cdots(x_j - \rho_{2k}) \qquad j \in [n], \text{ or equivalently,} \tag{15}$$

$$D_{2k}(x_j) = x_j^{2k} + \alpha_{2k-1}x_j^{2k-1} + \cdots + \alpha_1 x_j + \alpha_0 \qquad j \in [n], \tag{16}$$

where the correspondence between $\{\alpha_0, \ldots, \alpha_{2k-1}\}$ and $\{\rho_1, \ldots, \rho_{2k}\}$ is given by the well-known Vieta's formulas. To enforce finite domain variables, the axioms

$$D_{2k}(x_j) = 0 \qquad j \in [n], \tag{17}$$

are included in the proof systems. Hence every variable x_j can take $2k$ possible values which are the roots of Eq. (17). We denote by $D = \{\rho_1, \rho_2, \ldots, \rho_{2k}\}$ the set of domain values, i.e. $D_{2k}(v) = 0$ if and only if $v \in D$, and we set

$$\mathcal{D} = \{D_{2k}(x_j) = 0 \mid j \in [n]\}. \tag{18}$$

For example, to enforce Boolean variables, the axioms $x_j^2 - x_j = 0$ are always included in the proof systems and in this case $D = \{0, 1\}$.

In summary, we will consider polynomial systems of equations of the following form.

22

Definition 4.1. *Let D be a finite domain. A* polynomial system over a finite domain D *is defined as a set $\mathcal{P}$ of the form*

$$\mathcal{P} = \mathcal{F} \cup \mathcal{D} = \{f_1 = 0, \ldots, f_m = 0\} \cup \{D_{2k}(x_j) = 0 \mid j \in [n]\}. \tag{19}$$

Recall that we are considering rational values for $\rho_1, \rho_2, \ldots, \rho_{2k}$ in (15). It follows that $\alpha_{2k-1}, \ldots, \alpha_0$ are also rational. Let β be the the minimum number of bits needed to encode each of the numbers $\alpha_{2k-1}, \ldots, \alpha_0$, and the values $\rho_1, \rho_2, \ldots, \rho_{2k}$ when the rational coefficients are represented with their reduced fractions written in binary. The *polynomial domain size* is the bit length of the binary encoding of $D_{2k}(x)$, namely $O(k\beta)$.

Now, we recall the following result claiming that every globally nonnegative univariate polynomial has an SoS decomposition with well structured coefficients.

Lemma 4.2. *[44, Section 4, Thm 23] Let $p \in \mathbb{R}[x]$ be a univariate polynomial of degree $2d$ with rational coefficients, such that each of them can be encoded with τ bits. Assume that $p(x) \geq 0$ for all $x \in \mathbb{R}$. Then, we have*

$$p = \sum_{i=1}^{2d+3} a_i q_i^2,$$

for some nonnegative rational constants a_i and some polynomials q_i of degree d with rational coefficients. All coefficients in this representation can be encoded with $O(d^3 + d^2\tau)$ bits.

The following result demonstrates that, for any polynomial system $\mathcal{P}$ over a finite domain D, polynomial-size SoS proofs can be constructed to establish that the variables are bounded.

Lemma 4.3. *There exists a positive rational number $t > \max_{i\in[2k]}\{2, |\rho_i|\}$ that can be encoded with $O(k\beta)$ bits such that there exist SoS proofs of degree $2k$*

$$t - x = \sum_{i=1}^{2k+3} a_i q_i^2 - D_{2k}(x), \tag{20}$$

$$t + x = \sum_{i=1}^{2k+3} \tilde{a}_i \tilde{q}_i^2 - D_{2k}(x), \tag{21}$$

for some rational constants a_i and some polynomials q_i with rational coefficients. All coefficients in this representations can be encoded with $poly(k, \beta)$ bits.

Proof. TOPROVE 13 □

The results of Lemma 4.3 and Definition 2.1 allow us to conclude that any polynomial system over a finite domain is explicitly Archimedean.

Proposition 4.4. *Let $\mathcal{P}$ be a polynomial system over a finite domain D. Then, $\mathcal{P}$ is explicitly Archimedean.*

4.2 Approximate simulation of PC by SoS

Berkholz [5] related different approaches for proving the unsatisfiability of a system of real polynomial equations. Over Boolean variables, he showed that SoS simulates PC refutation: any PC refutation of degree d can be converted into an SoS refutation of degree $2d$, with only a polynomial increase in size. In the non-Boolean setting, there are systems of equations that are easier to refute

for PC than for SoS [30]. Grigoriev and Vorobjov [30] show that the simulation of PC by SoS does not hold in the non-Boolean case, namely when the Boolean axioms $x_j^2 - x_j = 0$ are omitted. For example, the so-called telescopic system of equations, $\{yx_1 = 1, x_1^2 = x_2, \ldots, x_{n-1}^2 = x_n, x_n = 0\}$, has a PC refutation of degree n, but it requires exponential refutation degree in SoS [30]. However, it is not known if SoS can simulate PC in the (non-Boolean) general domain setting, namely when variables can take values from a general finite set of values. In this section, we address the existing knowledge gap by extending Berkholz's result to general domains. This complements the results in [30, 5]. Recall that we are considering a polynomial systems of the form

$$\mathcal{F} = \{f_1 = 0, \ldots, f_m = 0\} \cup \mathcal{D} = \{D_{2k}(x_j) = 0 \mid j \in [n]\} \tag{22}$$

over variables $x_1, \ldots, x_n$. We prove the following Lemma 4.5, which is a generalization of ([5, Lemma 1]). The overall structure of the proof partially follows from the one in [5] but with some significant differences that will be emphasized below in the proof (see Case 4).

Lemma 4.5. *Let $(r_1, r_2, \ldots r_L)$ be a PC derivation from $\mathcal{F} \cup \mathcal{D}$ of degree d and size S. Then, for every $H \leq L$ there exists an SoS proof of $-(r_H)^2$ of degree $2(d + k - 1)$ of the following form*

$$\sum_{i=1}^{m}(-a_i f_i)f_i + \sum_{j=1}^{n} q_j D_{2k}(x_j) + \sum_{j=1}^{m_1} c_j(p_j)^2 = -(r_H)^2, \tag{23}$$

such that:

- *m_1 is constant bounded by $O(k, H)$,*

- *a_i, c_j are nonnegative constants, and q_j (for $j \in [n]$), p_j (for $j \in [m_1]$) are polynomials,*

- *all coefficients in the proof can be encoded with $poly(k, \beta, S)$ bits.*

Proof. TOPROVE 14 □

Remark 4.6 (Finite domains of odd size)**.** *The result of Lemma 4.5 extends to domains with an odd number of elements. Specifically, if the domain D has size $|D| = |\rho_1, \rho_2, \ldots, \rho_{2k-1}| = 2k - 1$, the same conclusion follows. To show this, we employ the same proof strategy as used in Cases 1, 2, and 3. However, in Case 4, instead of the domain polynomials defined as*

$$D_{2k-1}(x_i) = (x_i - \rho_1) \cdots (x_i - \rho_{2k-1}),$$

one can consider a modified set of polynomials, denoted by $\tilde{D}$, where each polynomial is defined as

$$\tilde{D}_{2k-1}(x_i) = (x_i - \rho_1)^2(x_i - \rho_2) \cdots (x_i - \rho_{2k-1}).$$

Here, one root is repeated to ensure an even degree for the polynomials in $\tilde{D}$. With this adjustment, the same arguments apply, yielding an SoS proof from $\mathcal{F} \cup \mathcal{D} \cup \tilde{D}$, and hence from $\mathcal{F} \cup \mathcal{D}$, of degree $2(d + k - 1)$.

Remark 4.7. *In the previous result, we highlighted the dependence of the coefficients present in the SoS proof on the parameter β. Recall that β corresponds to the number of bits needed to encode the coefficients of the polynomials $D_{2k}(x)$ and its roots $\rho_1, \ldots, \rho_{2k}$. This parameter can be eliminated and implicitly linked to the size of the PC proof S. Specifically, it suffices to assume that the PC proof begins by deriving all the polynomials $D_{2k}(x_j)$ for $j \in [n]$.*

While Lemma 4.5 immediately establishes a simulation of PC by SoS as *refutation* systems, it remains unclear whether SoS can also simulate PC as a *derivation* system. Specifically, the existence of an SoS proof of "$-r^2 \geq 0$" does not immediately guarantee the existence of an SoS proof of "$\pm r \geq 0$". Further, it is not hard to find polynomial systems for which the latter does not hold. Consider the simple polynomial system $\{x^2\}$ from which it is trivial to derive "$-x^2 \geq 0$". However, there are no SoS proofs of "$\pm x \geq 0$" from such premise.

Interestingly, by the use of SoS approximability techniques developed in Section 3.3, we are able to work around and resolve this issue. Provided that, given a statement derived by PC "$r = 0$", an (arbitrarily) small approximation ε of the statement is allowed, the simulation holds. That is, there exist SoS proofs of "$r + \varepsilon \geq 0$" and of "$-r + \varepsilon \geq 0$".

Theorem 4.8. SoS approximates PC *with degree linear in the domain size k over general finite domains. That is, if there exists a* PC *derivation of "$r = 0$" with degree d and size S, then for every $\varepsilon > 0$, we have* SoS *proofs of "$r + \varepsilon \geq 0$" and "$-r + \varepsilon \geq 0$" with degree $O(d + k)$ and coefficients bounded by $2^{poly(k,S,\lg \frac{1}{\varepsilon})}$.*

Proof. TOPROVE 15 □

4.3 PC criterion

In this section we show that, within the context of finite domains, Lemma 4.5 can be combined with the SoS$_\varepsilon$ criterion to formulate a new criterion, called PC *criterion*, based on the PC proof system. While, in general, PC is weaker than SoS as a proof system, it naturally connects to the theory of Gröbner basis, in particular to Buchberger's algorithm for their computation (see [9]). As we will see in Section 5, this connection enables the application of the PC criterion to certain families of problems arising from CSPs, for which the Nsatz criterion is not satisfied.

Theorem 4.9 (PC criterion). *Let $\mathcal{P} = \{p_1 = 0, \ldots, p_m = 0\}$ polynomial system over a finite domain D of $2k$ rational values, let $S = \boldsymbol{V}(\mathcal{P})$ be its variety and let $r \in \mathbb{R}[x_1, \ldots, x_n]$ be a polynomial nonnegative over S. Assume there exists an* SoS *proof of "$r \geq 0$" from $\mathcal{P}$ of degree $2d$*

$$r = \sum_{i=1}^{t_0} \sigma_i^2 + \sum_{i=1}^{m} h_i p_i.$$

Let $\mathcal{G}_{2d}$ be a $2d$-truncated Gröbner basis of $I(S)$ according to the grlex *order such that $\|\mathcal{G}_{2d}\|_\infty \leq 2^{poly(n^d)}$. Assume that, for every $g \in \mathcal{G}_{2d}$, there exist a* PC *derivation of g from $\mathcal{P}$ of size $poly(n^d)$ and degree $O(d)$. Then, for every $\varepsilon > 0$, the polynomial "$r + \varepsilon \geq 0$" has a degree-$O(d)$* SoS *proof*

$$r + \varepsilon = \sum_{i=1}^{t} \tilde{\sigma}_i^2 + \sum_{i=1}^{m} \tilde{h}_i p_i,$$

where the coefficients of every polynomial appearing in the proof are bounded by $2^{poly(n^d, \lg \frac{1}{\varepsilon})}$.

Proof. TOPROVE 16 □

5 Strong Separation for certain Constraint Satisfaction Problems

In what follows, we establish Theorem 1.8 by demonstrating and utilizing the ability of SoS to approximate a dynamic proof system, such as PC (see Theorem 4.8). In light of Theorem 4.9, it

is sufficient to show that PC can solve in polynomial time $\text{IMP}_d(\Gamma)$ when Γ is a finite constraint language closed under a semilattice polymorphism (see Theorem 5.5), and in the case it is closed under a dual-discriminator polymorphism (see Theorem 5.6). The degree lower bound for NSATZ given in [17], along with the results of this section, and Theorem 4.9 gives the claimed separation among the SoS_ε and NSATZ criteria.

The structure of the following sections is outlined as follows. The literature review, along with essential background and notation, is presented in Section 5.1 and Section 5.2, respectively. The proofs of Theorem 5.5 and Theorem 5.6 are provided in Section 6 and Section 7, respectively.

5.1 Related results

In [45, 6], Mastrolilli and Bharathi initiated a systematic study of the IMP_d tractability for combinatorial ideals arising from Constraint Satisfaction Problems $\text{CSP}(\Gamma)$ in which the type of constraints is restricted to relations from a set Γ over the Boolean domain. Note that $\text{CSP}(\Gamma)$ is just the special case of not-$\text{IMP}_0(\Gamma)$ with $r = 1$. The main results of [45, 6] identified the borderline of tractability of $\text{IMP}_d(\Gamma)$ for languages Γ over the Boolean domain. By using Gröbner bases techniques, they expanded Schaefer's dichotomy theorem [58] which classifies all CSPs of the form $\text{CSP}(\Gamma)$ over the Boolean domain to be either in P or NP-complete. Recently, Bulatov and Rafiey [15, 14] continued this line of research by extending [45, 6] beyond Boolean domains in several ways.

With the aim of expanding the class of $\text{IMP}_d(\Gamma)$s tractable by PC, we observe that some of the algorithms that are considered in [15, 14, 45, 6] for solving the $\text{IMP}_d(\Gamma)$ are known to not being simulable by PC and by SoS. For example, when Γ is closed under the minority polymorphism, in [6] it is shown that the membership proof for $\text{IMP}_d(\Gamma)$ can be computed in $n^{O(d)}$ time for any $d \in \mathbb{N}$. Note that $3\text{LIN}(2)$ is a special case of this class of problems. However, linear (thereby, sharp) lower bounds on degrees for SoS refutations are known [28] for $3\text{LIN}(2)$. It follows that bounded degree SoS and PC over the reals cannot simulate the algorithm in [6]. The approach in [6] has been generalized by [15] by showing that constructing a d-truncated Gröbner Basis for an ideal I is reducible to solving χIMP_d for the ideal I (see [15] for details). With this reduction at hand, they designed a general algorithmic approach, inspired by the famous FGLM algorithm [23] and the conversion algorithm in [6], to construct d-truncated Gröbner Basis for many combinatorial ideals, in particular, combinatorial ideals arising from languages invariant under a semilattice, or the dual-discriminator, or languages expressible as linear equations over $GF(p)$. In light of the impossibility result for the particular case of $3\text{LIN}(2)$ discussed earlier, the general approach presented by [15], which also works for $3\text{LIN}(2)$, cannot in general be simulated by PC.

In Section 5, we complement the aforementioned impossibility result with some positive results. More precisely, we show that PC is powerful enough to solve $\text{IMP}_d(\Gamma)$ when Γ is closed under a semilattice polymorphism or the dual discriminator. As a result of the aforementioned considerations, our approach differs fundamentally from the general methodology employed in [15] (see also the discussion in Remark 6.1).

Furthermore, strategies in [6, 15, 13, 45] to address the problem of SoS bit complexity involve replacing the original input polynomial constraints $\mathcal{P}$ (see Definition 1.1) with a new set of polynomials $\mathcal{P}^{(d)}$ that satisfies the NSATZ criterion, and generally depends on the SoS degree d. This set $\mathcal{P}^{(d)}$ is computed externally (by an algorithm specifically designed for this purpose), serving as the input for SoS in place of $\mathcal{P}$. For example, in the semilattice case, if $\mathcal{P}$ consists of m polynomials, the set $\mathcal{P}^{(d)}$, used in [15, 45], is generated by a specific algorithm and has a size of $m^{O(d)}$; that is, $\mathcal{P}^{(d)}$ depends on d and grows exponentially with the SoS degree d. This preprocessing step ensures that SoS retains "low'' bit complexity, but only if $\mathcal{P}$ is substituted with $\mathcal{P}^{(d)}$. Essentially, the approach utilized in [6, 15, 13, 45] is to apply the NSATZ criterion without enhancing or extending

it, with the goal of replacing the initial input polynomial system with a new one that is computed externally and satisfies the NSATZ criterion. Our results demonstrate that all preprocessing steps employed in [6, 15, 45] are unnecessary, as SOS achieves low bit complexity for any fixed d when $\mathcal{P}$ is provided directly as input.

5.2 Background and notation for $\mathrm{CSP}(\Gamma)$

In this section we give the basic definitions and results that we will need later. We refer to [3, 12, 18, 45, 15] for more details.

Let D denote a finite set called the **domain**. By a k-ary **relation** R on a domain D we mean a subset of the k-th cartesian power D^k; k is said to be the **arity** of the relation. We often use relations and (affine) varieties interchangeably since both are subsets of D^k (we will not refer to varieties from universal algebra in this paper). A **constraint language** Γ over D is a set of relations over D. A constraint language is **finite** if it contains finitely many relations, and is **Boolean** if it is over the two-element domain $\{0, 1\}$.

A **constraint** over a constraint language Γ is an expression of the form $R(x_{i_1}, \ldots, x_{i_k})$ where R is a relation of arity k contained in Γ, and $x_{i_1}, \ldots, x_{i_k}$ are variables that belong to the variable set X. A constraint is satisfied by a mapping ϕ defined on the variables if $(\phi(x_{i_1}), \ldots, \phi(x_{i_k})) \in R$.

Definition 5.1. *The* (nonuniform) CONSTRAINT SATISFACTION PROBLEM (CSP) *associated with language* Γ *over* D *is the problem* $\mathrm{CSP}(\Gamma)$ *in which: an instance is a triple* $\mathcal{C} = (X, D, C)$ *where* $X = \{x_1, \ldots, x_n\}$ *is a set of* n *variables and* C *is a set of constraints over* Γ *with variables from* X. *The goal is to decide whether or not there exists a solution, i.e. a mapping* $\phi : X \to D$ *satisfying all of the constraints. We will use* $Sol(\mathcal{C}) \subseteq D^n$ *to denote the set of solutions of* $\mathcal{C}$.

Moreover, we follow the algebraic approach to Schaefer's dichotomy result [58] formulated by Jeavons [33] where each class of CSPs that are polynomial time solvable is associated with a polymorphism. Recall that a polymorphism of a constraint language Γ over a set D is a multi-ary operation on D that can be viewed as a multidimensional symmetry of relations from Γ (see e.g. [3]).

Definition 5.2. *An operation* $f : D^m \to D$ *is a* **polymorphism** *of a relation* $R \subseteq D^k$ *if for any choice of* m *tuples* $(t_{11}, \ldots, t_{1k}), \ldots, (t_{m1}, \ldots, t_{mk})$ *from* R *(allowing repetitions), it holds that the tuple obtained from these* m *tuples by applying* f *coordinate-wise,* $(f(t_{11}, \ldots, t_{m1}), \ldots, f(t_{1k}, \ldots, t_{mk}))$, *is in* R. *We also say that* f **preserves** R, *or that* R *is* **invariant** *or* **closed** *with respect to* f. *A polymorphism of a constraint language* Γ *is an operation that is a polymorphism of every* $R \in \Gamma$. *By* $\mathsf{Pol}(\Gamma)$ *we denote the set of all polymorphisms of* Γ.

5.2.1 The ideal membership problem of a constraint language $IMP(\Gamma)$

The polynomial IDEAL MEMBERSHIP PROBLEM (IMP) is the following computational task. Let $\mathbb{Q}[x_1, \ldots, x_n]$ be the ring of polynomials over the field $\mathbb{Q}$ and indeterminates $\{x_1, \ldots, x_n\}$ ordered according to the grlex order (see Section 2). Given $f_0, f_1, \ldots, f_r \in \mathbb{Q}[x_1, \ldots, x_n]$ we want to decide if $f_0 \in \mathrm{I} = \langle f_1, \ldots, f_r \rangle$, where I is the ideal generated by $F = \{f_1, \ldots, f_r\}$. If the ideal I corresponds to a CSP instance we can be specific on its structure. Here, we explain how to construct an ideal corresponding to a given $\mathrm{CSP}(\Gamma)$ instance $\mathcal{C}$ by following [45]. Constraints are in essence varieties (see e.g. [61, 36]).

Definition 5.3. *For any given* $\mathrm{CSP}(\Gamma)$ *instance* $\mathcal{C} = (X, D, C)$, *the* **combinatorial ideal**

$$\mathrm{I}_{\mathcal{C}} = \langle f_{R_1}(X_{R_1}), \ldots, f_{R_\ell}(X_{R_\ell}), f_D(x_1), \ldots, f_D(x_n) \rangle \tag{24}$$

is defined as the vanishing ideal of the set $Sol(\mathcal{C})$ *and it is constructed as follows.*

- *For every $x_i \in X$ the ideal $I_\mathcal{C}$ contains a domain polynomial $f_D(x_i)$ whose zeroes are precisely the elements of the domain D.*

- *For every constraint $R_j(X_{R_j}) \in C$, where X_{R_j} is a tuple of variables from X, the ideal $I_\mathcal{C}$ contains a polynomial $f_{R_j}(X_{R_j})$ such that for $X_{R_j} \in D^{|X_{R_j}|}$ it holds $f_{R_j}(X_{R_j}) = 0$ if and only if $R_j(X_{R_j})$ is true.*

See [45] for more details and properties.

Definition 5.4. *The* IDEAL MEMBERSHIP PROBLEM *associated with language Γ is the problem* $\mathrm{IMP}(\Gamma)$ *in which the input consists of a polynomial $f \in \mathbb{Q}[x_1, \ldots, x_n]$ and a* $\mathrm{CSP}(\Gamma)$ *instance $\mathcal{C} = (X, D, C)$ where $D \subset \mathbb{Q}$. The goal is to decide whether f lies in the combinatorial ideal $I_\mathcal{C}$. We use $\mathrm{IMP}_d(\Gamma)$ to denote $\mathrm{IMP}(\Gamma)$ when the input polynomial f has degree at most d.*

Ideal membership testing can be performed by means of Gröbner bases. Indeed, if we can compute the d-truncated Gröbner basis $\mathcal{G}_d$ of $I_\mathcal{C}$ in $n^{poly(n^d)}$ time, then we can solve $\mathrm{IMP}_d(\Gamma)$ in polynomial time (see Section 2).

As in the case of the CSP, polymorphisms of Γ are what determines the complexity of $\mathrm{IMP}_d(\Gamma)$ (see [45, 6, 15]).

5.3 Polynomial Calculus and semilattice polymorphism

We consider the complexity of $\mathrm{IMP}_d(\Gamma)$ for constraint languages Γ closed under a *semilattice* operation ψ (either meet or join). There are two kinds of semilattice operations (see e.g. [22]). A *join*-semilattice, also known as an *upper*-semilattice, refers to a partially ordered set that possesses a *join* (or least upper bound) for every nonempty finite subset. Conversely, a *meet*-semilattice, or *lower*-semilattice, is a partially ordered set characterized by having a *meet* (or greatest lower bound) for any nonempty finite subset. Algebraically, semilattices can be defined as pairs $\mathcal{D} = (D, \phi)$, where D is a domain and ϕ is the semilattice operation *join* or *meet*. Note that both operations are associative, commutative and idempotent binary operations.

In the following, we show that standard PC is d-complete and efficient for constraint languages that are closed under a semilattice polymorphism. Our result greatly simplifies known approaches [15, 45] and unifies them into one simple PC-based algorithm. Further details explaining the substantial differences with what is already known are given and discussed in Remark 6.1. Our main technical result is as follows.

Theorem 5.5. *Let Γ be a finite constraint language over a domain D. Consider an instance $\mathcal{C}$ of* $\mathrm{CSP}(\Gamma)$*. If Γ is closed under a semilattice polymorphism, then $O(d)$-degree PC can compute in $n^{O(d)}$ time the reduced d-truncated Gröbner basis $\mathcal{G}_d$ (in grlex order) of the combinatorial ideal $I_\mathcal{C}$, for any degree $d \in \mathbb{N}$ and where n is the number of variables.*

Proof. TOPROVE 17 $\qquad\square$

Theorem 5.5 in conjunction with Theorem 4.9 implies Theorem 1.8 for semilattice structures.

5.4 Polynomial Calculus and dual discriminator polymorphism

We consider the complexity of $\mathrm{IMP}_d(\Gamma)$ for constraint languages where the dual discriminator operation is a polymorphism of Γ. The dual discriminator is a well-known majority operation [34, 3] and is often used as a starting point in many CSP related classifications [3]. For a finite domain D, a ternary operation f is called a majority operation if $f(a, a, b) = f(a, b, a) = f(b, a, a) = a$ for all

$a, b \in D$. The *dual discriminator* ∇ on a domain D, is a majority operation such that $\nabla(a, b, c) = a$ for pairwise distinct $a, b, c \in D$.

In [15] it is shown that $\mathrm{IMP}_d(\Gamma)$ is solvable in polynomial time for any fixed d. The work in [6] complements the result in [15] by proving that the *full* (as opposed to *truncated*) Gröbner basis in graded lexicographic order can be computed in polynomial time and with bounded degree, thus proving polynomial time efficiency for solving the general $\mathrm{IMP}(\Gamma)$ (see also Appendix B).

In the following, we again show the power of the PC by demonstrating that the ad hoc algorithm presented in [6] is simulable by PC. This greatly simplifies previous algorithms [15, 6, 7] and provides another family of problems for which the SoS_ε criterion is provably stronger than the NSATZ criterion. Our main technical result is as follows.

Theorem 5.6. *Let Γ be a finite constraint language over a domain D. Consider an instance $\mathcal{C}$ of $\mathrm{CSP}(\Gamma)$. If Γ is closed under a dual discriminator polymorphism, then PC can compute in $n^{O(1)}$ time the reduced Gröbner basis $\mathcal{G}$ (in grlex order) of the combinatorial ideal $I_{\mathcal{C}}$, where n is the number of variables and $|D| = O(1)$.*

Proof. TOPROVE 18 $\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad$ $\square$

Theorem 5.6, along with Theorem 4.9, implies Theorem 1.8 for dual discriminator structures.

6 Proof of Theorem 5.5

We consider the complexity of $\mathrm{IMP}_d(\Gamma)$ for constraint languages Γ where $\mathsf{Pol}(\Gamma)$ (see Definition 5.2) includes a *semilattice* operation ψ (either meet or join). There are two kinds of semilattice operations (see e.g. [22]). A *join*-semilattice, also known as an *upper*-semilattice, refers to a partially ordered set that possesses a *join* (or least upper bound) for every nonempty finite subset. Conversely, a *meet*-semilattice, or *lower*-semilattice, is a partially ordered set characterized by having a *meet* (or greatest lower bound) for any nonempty finite subset. Algebraically, semilattices can be defined as pairs $\mathcal{D} = (D, \phi)$, where D is a domain and ϕ is the semilattice operation *join* or *meet*. Note that both operations are associative, commutative and idempotent binary operations. In mathematics, the symbol for the join (meet) operation in a semilattice is often denoted by the symbol $\vee$ ($\wedge$). Any such operation induces a partial order ($\preceq$) (and its corresponding inverse order) in which the result of the operation for any two elements represents the least upper bound (or greatest lower bound) of those elements in relation to the established partial order.

The input to $\mathrm{IMP}_d(\Gamma)$ consists of any given set of polynomials that defines the combinatorial ideal $I_{\mathcal{C}}$ (see Definition 5.3) corresponding to a semilattice closed language Γ:

$$f_{R_1}(X_{R_1}), \ldots, f_{R_\ell}(X_{R_\ell}), f_D(x_1), \ldots, f_D(x_n). \tag{25}$$

We want to show that PC is capable of computing the d-truncated Gröbner basis (in grlex order) in polynomial time for any fixed d.

Theorem 5.5 proof outline. Schematically, Theorem 5.5 is proven by the following arguments:

(i) First we prove Theorem 5.5 for the Boolean case, where the domain $D = \{0, 1\}$. That is, we show that bounded-degree PC computes the d-truncated Gröbner basis for the Boolean domain. The known [45] algorithm to efficiently compute the d-truncated Gröbner basis consists of "guessing" the truncated Gröbner basis in polynomial time. Here, the main technical difficulty is that this guessing "trick" is not immediately simulable in an efficient way by PC.

We show that the latter is possible. The algorithm in [45] essentially reduces the IMP for a given polynomial f in the $2d$-truncated Gröbner basis to the (contrapositive) problem of checking whether "non-vanishing assignments" of variables for f belong to the variety. In this work, we are able to PC-derive f by polynomially formulating "non-vanishing assignments" into an infeasible system of Horn-type polynomials. We then combine algebraic and logical reasoning, leading us to efficient PC refutations of the new system by means of simulation of refutation proofs. By accurately using the PC ability for refutation, we can then retrieve a PC derivation of f. This technique may be of independent interest.

(ii) We reduce the general case (with arbitrary finite domain D) to the Boolean case. The reduction is achieved by encoding the domain D using strings over $\{0, 1\}$. The encoding is given by a novel bijective map that preserves the semilattice structure. The strength of our bijection is that it ensures a one-to-one correspondence between the solution spaces of the original CSP over D and the reduced Boolean problem, which allows us to reduce the (search version of) $\text{IMP}_d(\Gamma)$ to the (search version of) $\text{IMP}_{O(d)}(\Gamma^{01})$, where Γ^{01} is a Boolean constraint language derived from Γ. Crucially, the preservation of the semilattice structure ensures that $\text{IMP}_{O(d)}(\Gamma^{01})$ remains solvable in polynomial time by PC.

More details on the second point are given below.

1. Show that any instance $\mathcal{C} = (X, D, C)$ of $\text{CSP}(\Gamma)$ is reducible to an instance C^{01} of $\text{CSP}(\Gamma^{01})$, where Γ^{01} is a finite constraint language over $\{0, 1\}$ and so that there exists a $\phi \in \text{Pol}(\Gamma^{01})$ that is a semilattice (Min or Max polymorphisms) (see Section 6.2.2). The idea is that we can "encode" $\mathcal{C}$ in binary and that the encoding function is invertible (see Section 6.2.1).

2. Show $\text{IMP}_d(\Gamma)$ is reducible to $\text{IMP}_{(|D|-1)d}(\Gamma^{01})$, where Γ^{01} is a constraint language over the Boolean domain $\{0, 1\}$ and there exists a semilattice $\phi \in \text{Pol}(\Gamma^{01})$ (Min or Max polymorphism) (see Section 6.2.3). In addition, the reduction ensures that the varieties in the two different domains are in one-to-one correspondence.

3. By Corollary 6.8, we can solve $\text{IMP}_{(|D|-1)d}(\Gamma^{01})$ by bounded-degree PC.

4. Our reduction guarantees that we can recover a bounded-degree PC proof in the finite domain from the bounded-degree PC proof over the Boolean domain (see Section 6.2.4). More precisely, we show how PC proofs of degree $(|D|-1)d$ in the Boolean domain translate into the PC proofs of degree $O(d)$ in the finite domain, thus proving Theorem 5.5.

Remark 6.1. *We emphasize that a reduction to the Boolean domain case has also been used to prove that the decision version of $\text{IMP}_d(\Gamma)$ is tractable for such constraint languages Γ (see [15, Th. 5.10] for more details). However the mapping used in [15] is by the means of pp-interpretability (see [50]), and it is not guaranteed that one can recover proofs for the finite domain under this reduction. In particular, the very first obstacle is given by the fact the mapping π in the definition of pp-interpretability [15, Def. 3.12] is not guaranteed to be a bijection. In fact, this difficulty of transforming the Boolean case proof to the finite general domain case led to the development of a specific method [15, Th. 6.5] for the search version of the problem. Moreover, as previously noted at the beginning of Section 5, it is far from evident that it can be simulated by PC. In the following, we show that the standard bounded-degree PC approach is sufficient. This simplifies known approaches and unifies them into one simple PC-based approach.*

6.1 Min/Max polymorphisms

Two important classes of polymorphisms that played a fundamental role in the celebrated dichotomy theorem by Schaefer [58] are the Min/Max polymorphisms. In fact, Min (Max) is a polymorphism of the (dual) problem HORN-SAT [35]. A Boolean language Γ is invariant under semilattice operations given by (component-wise) the Max operation (logical OR) or the Min operation (logical AND). The semilattice polymorphism is a well-known generalization of Min/Max polymorphisms for the general finite domain. In this section we show that PC is d-complete and efficient for polynomial systems that are closed with respect to the Min polymorphism (a similar proof holds for Max).

In the remainder of this section we focus on system of polynomials $C_1, \ldots, C_m$ that are Min closed, namely each polynomial constraint C_i, along with domain polynomials, has solutions that are closed with respect to the Min polymorphism. These polynomials are defined in Definition 5.3.

6.1.1 Min polymorphism

We will make use of the following definition from [45].

Definition 6.2. *For a given set $X = \{x_1, \ldots, x_n\}$ of variables and for any set $S \subseteq [n]$ possibly empty, $\alpha \in \{0, \pm 1\}$, let a* **term** *be defined as* [4]

$$\tau^+(S) \stackrel{\text{def}}{=} \alpha \prod_{i \in S} x_i, \quad \text{*POSITIVE TERM*}$$

$$\tau^-(S) \stackrel{\text{def}}{=} \alpha \prod_{i \in S} (x_i - 1). \quad \text{*NEGATIVE TERM*}$$

For $S_1, S_2 \subseteq [n]$ and $i \in [n]$, let a **2-terms polynomial** *be a polynomial that is the sum of two terms or it is $\pm(x_i^2 - x_i)$. We say that a set G of polynomials is* **2-terms structured** *if each polynomial from G is a 2-terms polynomial.*

We further distinguish between the following special 2-terms polynomials:

$$\mathcal{T}^+ \stackrel{\text{def}}{=} \{\tau^+(S_1) + \tau^+(S_2) \mid S_1, S_2 \subseteq [n]\} \cup \{\pm(x_i^2 - x_i) \mid i \in [n]\}, \quad \text{*POSITIVE 2-TERMS*}$$

$$\mathcal{T}^- \stackrel{\text{def}}{=} \{\tau^-(S_1) + \tau^-(S_2) \mid S_1, S_2 \subseteq [n]\} \cup \{\pm(x_i^2 - x_i) \mid i \in [n]\}. \quad \text{*NEGATIVE 2-TERMS*}$$

Remark 6.3. Γ *is a finite language. It follows that each given polynomial constraint C_i (as defined in Definition 5.3) has a constant number of feasible solutions. Therefore PC can efficiently derive any polynomial vanishing over the solutions of C_i, and in particular it can derive any vanishing 2-term polynomial.*

Lemma 6.4. *If Min $\in$ Pol(Γ) then PC can compute the truncated reduced Gröbner basis $\mathcal{G}_d$ in $n^{O(d)}$ time (where n is the number of variables), for any degree $d \in \mathbb{N}$.*

Proof. TOPROVE 19 □

Lemma 6.5. *If $f = p - q$ belongs to $\mathcal{G}_d$ then the following polynomials*

$$p_j = p(x_j - 1) \quad \forall j \in Q, \quad q_i = q(x_i - 1) \quad \forall i \in P \tag{26}$$

belong to the combinatorial ideal $I_{\mathcal{C}}$ and there is a $O(d)$-PC proof of this fact.

Proof. TOPROVE 20 □

---

[4]The empty product has the value 1.

Lemma 6.6. *If $f = p - q$ belongs to $\mathcal{G}_d$, then $f = 0$ admits a degree-$O(d)$ PC proof.*

Proof. TOPROVE 21 $\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad$ $\square$

By symmetric arguments we obtain the following.

Lemma 6.7. *If $\mathsf{Max} \in \mathsf{Pol}(\Gamma)$ then PC can calculate the reduced truncated basis Gröbner $\mathcal{G}_d$ in $n^{O(d)}$ time (where n is the number of variables), for any degree $d \in \mathbb{N}$.*

Corollary 6.8. *Let Γ be a finite constraint language over $\{0, 1\}$ that is closed under a 2-element semilattice operation polymorphism. Then PC can compute the reduced truncated Gröbner basis $\mathcal{G}_d$ in $n^{O(d)}$ time (where n is the number of variables), for any degree $d \in \mathbb{N}$.*

6.2 Generalizing to finite domain semilattice

In the following, we generalize Corollary 6.8 to constraint languages over finite domains that are closed under a semilattice polymorphism and obtain the proof of Theorem 5.5. We begin by recalling the definition of semilattice operations. We then present the main arguments used to prove Theorem 5.5, followed by their details.

6.2.1 Binary encoding

Let $\mathcal{D} = (D, \psi)$ be a semilattice, with ψ being a semilattice polymorphism (see Definition 5.2) of the constraint language Γ. Then, it is known that $\mathcal{D}$ can be encoded in binary form in such a way that it is a subalgebra of $\mathcal{B}^k$ for some $k \in \mathbb{N}$, where $\mathcal{B} = (\{0, 1\}, \phi)$ is a 2-element semilattice [50]. In the following we show how to encode the elements of D in binary in a proper form such that (i) the just mentioned property [50] is satisfied, and (ii) it will allow us to recover proofs over the finite domain D from the Boolean domain, a property that is not guaranteed by the approach considered in [15]. The encoding μ is very "natural" and it is described in the proof of the following lemma.

Lemma 6.9. *Let $\mathcal{D} = (D, \psi)$ be a finite semilattice where ψ is a meet-semilattice (join-semilattice) operation. Then there is a mapping $\mu : D \to \{0, 1\}^{(|D|-1)}$ such that Min (Max) is a polymorphism of the Boolean relation $D^{01} = \{\mu(d_1), \ldots, \mu(d_{|D|})\}$ and $\mu : D \to D^{01}$ is bijective.*

Proof. TOPROVE 22 $\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad$ $\square$

6.2.2 Reducing CSP(Γ) over a finite domain to the Boolean domain

By Lemma 6.9 we obtain the following.

Lemma 6.10. *Let Γ be a constraint language over D that is closed with respect to a meet (join) semilattice operation. Let Γ^{01} be the constraint language over $\{0, 1\}$ that is obtained from Γ by replacing the values from D appearing in the relations from Γ with their corresponding binary encoding, as given by the mapping μ in Lemma 6.9. Then*

- *Min (Max) is a polymorphism of Γ^{01}.*

- *Any given instance $\mathcal{C} = (X, D, C)$ of CSP(Γ) is polynomial time reducible to an instance $\mathcal{C}^{01} = (Y, \{0, 1\}, C^{01})$ of CSP(Γ^{01}). Moreover, the solution sets $Sol(\mathcal{C})$ and $Sol(\mathcal{C}^{01})$ are in one-to-one correspondence.*

Proof. TOPROVE 23 $\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad$ $\square$

6.2.3 Reducing $\text{IMP}_d(\Gamma)$ over a finite domain to the Boolean domain

In the reduction considered in Lemma 6.10 every variable $x_i \in X$ is mapped to $(|D| - 1)$ new binary variables $y_{i1}, \ldots, y_{i(|D|-1)}$. In the following we reduce $\text{IMP}_d(\Gamma)$ to $\text{IMP}_{(|D|-1)d}(\Gamma^{01})$. This reduction, along with its corresponding "inversion" (see Section 6.2.4), will be used to prove Theorem 5.5, as summarized in Section 6.

The interpolating polynomial $\mathcal{P}$ in the Boolean domain. By a straightforward generalization of Lagrange interpolating polynomials (see e.g. [54], [25]), given $|D|$ distinct values $\mu(d_1), \ldots, \mu(d_{|D|}) \in \{0, 1\}^{(|D|-1)}$ (see Lemma 6.9) and corresponding values $d_1, \ldots, d_{|D|}$, there exists a polynomial $\mathcal{P}$ of degree at most $|D| - 1$ that interpolates the data, i.e. $\mathcal{P}(\mu(d_i)) = d_i$ for each $i = 1, \ldots, |D|$.

A reduction to the Boolean domain. As in the proof of Lemma 6.10, we want to map every variable $x_i \in X$ to a tuple of $(|D| - 1)$ new binary variables $y_{i1}, \ldots, y_{i(|D|-1)}$. Let $Y_i = y_{i1}, \ldots, y_{i(|D|-1)}$ and $Y = \{Y_1, \ldots, Y_n\}$.
To guarantee that each tuple Y_i assumes only values that correspond to valid encondings of elements in D, we consider the following "low" degree polynomial:

$$\mathcal{T}(y_1, \ldots, y_{(|D|-1)}) = \prod_{v_1, \ldots, v_{(|D|-1)} \in D^{01}} (1 - \prod_{j=1}^{(|D|-1)} (1 - v_j + y_j))).$$

Every $\mathcal{T}(Y_i)$ has degree $|D|(|D| - 1)$.
For any given $\text{CSP}(\Gamma)$-instance $\mathcal{C} = (X, D, C)$, the corresponding combinatorial ideal (see Definition 5.3)

$$\mathcal{F} = \{f_{R_1}(x_{1_{R_1}} \ldots, x_{k_{R_1}}), \ldots, f_{R_\ell}(x_{1_{R_\ell}} \ldots, x_{k_{R_\ell}}), f_D(x_1), \ldots, f_D(x_n)\} \tag{27}$$

$$\text{I}_\mathcal{C} = \langle \mathcal{F} \rangle \tag{28}$$

is mapped to

$$\mathcal{F}^{01} = \{f_{R_1}(\mathcal{P}(Y_{1_{R_1}}), \ldots, \mathcal{P}(Y_{k_{R_1}})), \ldots, f_{R_\ell}(\mathcal{P}(Y_{1_{R_\ell}}) \ldots, \mathcal{P}(Y_{k_{R_\ell}})),$$

$$\mathcal{T}(Y_1), \ldots, \mathcal{T}(Y_n), y_1^2 - y_1, \ldots, y_{n(|D|-1)}^2 - y_{n(|D|-1)}\} \tag{29}$$

$$\text{I}_\mathcal{C}^{01} = \langle \mathcal{F}^{01} \rangle. \tag{30}$$

Note that

- The polynomial constraints $\{\mathcal{T}(Y_i) = 0 \mid \forall i \in [n]\}$ (along with $y_i^2 - y_i = 0$) forces each tuple Y_i to take only the values from D^{01}.

- The polynomial constraint $f_{R_j}\left(\mathcal{P}(Y_{1_{R_j}}), \ldots, \mathcal{P}(Y_{k_{R_j}})\right) = 0$, for all $j \in [\ell]$, forces the tuples $Y_{i_{R_j}}$ to take only the values whose corresponding value (according to μ, see Lemma 6.9) in finite domain satisfy $f_{R_1}(x_{1_{R_1}} \ldots, x_{k_{R_1}})$.

- Let $u(x_1, \ldots, x_n)$ and consider $u^{01}(\mathcal{P}(Y_1), \ldots, \mathcal{P}(Y_n))$. Then

$$u(x) = 0 \iff u^{01}(Y) = 0 \iff (\mathcal{P}(Y_1), \ldots, \mathcal{P}(Y_n)) \in \mathbf{V}(\text{I}_\mathcal{C}).$$

Thus $\mathbf{V}(\text{I}_\mathcal{C})$ and $\mathbf{V}(\text{I}_{\mathcal{C}^{01}})$ are in one-to-one correspondence and $u \in \text{I}_\mathcal{C}$ if and only if $u^{01} \in \text{I}_\mathcal{C}^{01}$.

- If $\deg(u) = d$, then $\deg(u^{01}) \le (|D| - 1)d$.

Note that the set of satisfying assignments $Sol(\mathcal{C})$ corresponds to the variety $\mathbf{V}(\mathrm{I}_\mathcal{C})$ of $\mathrm{I}_\mathcal{C}$, i.e. $Sol(\mathcal{C}) = \mathbf{V}(\mathrm{I}_\mathcal{C})$.

Corollary 6.11. $\mathrm{IMP}_d(\Gamma)$ *is polynomial time reducible to* $\mathrm{IMP}_{(|D|-1)d}(\Gamma^{01})$, *where* Γ^{01} *if a finite constraint language over* $\{0,1\}$ *that is closed under a 2-element semilattice operation polymorphism, and such that* $\mathbf{V}(\mathrm{I}_\mathcal{C})$ *and* $\mathbf{V}(\mathrm{I}_{\mathcal{C}^{01}})$ *are in one-to-one correspondence.*

6.2.4 Mapping the Boolean PC proof back to finite domain

Let $|D| = O(1)$. By Lemma 6.4, we can solve $\mathrm{IMP}_{(|D|-1)d}(\Gamma^{01})$ by bounded-degree PC in $n^{O(d)}$ time. In the following we show the existence of a polynomial time bounded-degree PC proof for $\mathrm{IMP}_{(|D|-1)d}(\Gamma^{01})$ implies a polynomial time bounded-degree PC proof for $\mathrm{IMP}_d(\Gamma)$, and hence the proof of Theorem 5.5 follows.

To begin, we introduce the following terminology.

Definition 6.12. *Let* $\mathrm{I} \subseteq \mathbb{Q}[x_1, \ldots, x_n]$ *be an ideal, and let* $f, g \in \mathbb{Q}[x_1, \ldots, x_n]$. *We say that* f *and* g *are* **congruent modulo** I, *written* $f \cong g \mod \mathrm{I}$, *if* $f - g \in \mathrm{I}$.

The interpolating polynomial $\mathcal{Q}_j$ in finite domain. For any given value $d_i \in D$ and $j \in [(|D| - 1)]$, we need a polynomial function $\mathcal{Q}_j(d_i)$ that returns the j-th bit of $\mu(d_i) = b_{i1}, \ldots, b_{i(|D|-1)}$, i.e. $\mathcal{Q}_j(d_i) = b_{ij}$. By Lagrange interpolating polynomials (see e.g. [54], [25]), given $|D|$ distinct values $d_1, \ldots, d_{|D|} \in D$ and corresponding values $b_{1j}, \ldots, b_{|D|j}$, there exists a polynomial $\mathcal{Q}_j$ of degree $|D| - 1$ that interpolates the data, i.e. $\mathcal{Q}_j(d_i) = b_{ij}$ for each $j = 1, \ldots, (|D| - 1)$.

Lemma 6.13.   *1.* $x \cong \mathcal{P}(\mathcal{Q}_1(x), \ldots, \mathcal{Q}_{(|D|-1)}(x)) \mod \langle f_D(x) \rangle$. [5]

    2. $\mathcal{T}(\mathcal{Q}_1(x), \mathcal{Q}_2(x), \ldots, \mathcal{Q}_{(|D|-1)}(x)) \cong 0 \mod \langle f_D(x) \rangle$.

    3. $\mathcal{Q}_j(x)^2 - \mathcal{Q}_j(x) \cong 0 \mod \langle f_D(x) \rangle$

Proof. TOPROVE 24 □

Proof. TOPROVE 25 □

7 Proof of Theorem 5.6

In this section, we focus on $\mathrm{IMP}_d(\Gamma)$, where Γ is a language closed under the *dual discriminator* polymorphism and show the proof of Theorem 5.6. The dual discriminator is a well-known majority operation [34, 3] and is often used as a starting point in many CSP-related classifications [3]. For a finite domain D, a ternary operation f is called a majority operation if $f(a, a, b) = f(a, b, a) = f(b, a, a) = a$ for all $a, b \in D$.

Definition 7.1. *The* dual discriminator *on a domain* D, *denoted by* ∇, *is a majority operation such that* $\nabla(a, b, c) = a$ *for pairwise distinct* $a, b, c \in D$.

---

[5] Recall $\langle f_D(x) \rangle$ is the ideal generated by the domain polynomial $f_D(x)$ of x.

The input for $\text{IMP}_d(\Gamma)$ consists of any given set of polynomials that defines the combinatorial ideal $\text{I}_\mathcal{C}$ (see Definition 5.3) corresponding to a dual discriminator closed language:

$$f_{R_1}(X_{R_1}), \ldots, f_{R_\ell}(X_{R_\ell}), f_D(x_1), \ldots, f_D(x_n). \tag{31}$$

We want to show that PC is capable of computing the full Gröbner basis (in grlex order) in polynomial time.

We will assume that the solution set is non-empty, as the search version of $\text{IMP}_0(\Gamma)$ can be solved by PC in polynomial time using "local-consistency" algorithms, valid for the dual-discriminator, as shown in [36].

Theorem 5.6 proof structure. The central idea is to adapt the algorithms presented in [6, 8] and [15] to design a PC algorithm that runs in polynomial time for any given domain D. The main arguments are as follows.

(i) For any given instance of $\text{IMP}_d(\Gamma)$, consider the corresponding $\text{CSP}(\Gamma)$ input instance $\mathcal{C} = (X, D, C)$ (see Definition 5.4). It is known that any instance $\mathcal{C} = (X, D, C)$ of $\text{CSP}(\Gamma)$ can be reduced to an equivalent $\text{CSP}(\Gamma)$ instance with only binary constraints (that is, constraints with at most two variables in their scope). In this transformed instance, the constraints are organized into three categories: *permutation* constraints, *complete* constraints, or *two-fan* constraints. This restructuring aligns with the classification introduced in prior work (see [20]). We derive binary constraints according to the CSP classification.

(ii) Consider the input combinatorial ideal $\text{I}_\mathcal{C}$ associated with the given instance of $\text{IMP}_d(\Gamma)$ (see Eq. (24)). The next phase consists in the decomposition of the combinatorial ideal $\text{I}_\mathcal{C}$ into a collection of simpler ideals. Each of these simpler ideals arises from the structured binary constraints considered above. The advantage here is that these individual ideals have Gröbner bases that can be derived efficiently through the PC algorithm.

(iii) Finally, we combine the Gröbner bases corresponding to the simpler ideals into a single Gröbner basis for the entire combinatorial ideal $\text{I}_\mathcal{C}$. This approach allows us to efficiently compute a solution to the original problem in polynomial time.

Schematically, Theorem 5.6 is proven by the following arguments:

1. In Section 7.1, by using the known result for CSPs mentioned in (i), $\text{IMP}_d(\Gamma)$ can be shown to be equivalent to $\text{IMP}_d(\Pi)$, where Π is a binary constraint language. To achieve this, we start from the given input polynomials (31) and derive bivariate polynomials describing the mentioned binary constraints. The derivation can be done in polynomial time and entirely within the framework of PC.

2. In Section 7.2.2, it is shown that a Gröbner basis for the combinatorial ideal generated by the *permutation* constraints I_{perm} can be calculated in polynomial time by PC. In particular, we define new constraints CPC_i that arise from "chaining" together permutation constraints, with the property that, if X_i and X_j are the variables in CPC_i and CPC_j respectively, then $X_i \cap X_j = \emptyset$. It follows that $\text{I}_{perm} = \sum_i \text{I}_{CPC_i}$, and a set of generators for each I_{CPC_i} is found.

3. In Section 7.2.3, similar to the permutation constraints case, a set of generators for the combinatorial ideal I_{CF} generated by the *complete* and *two-fan* constraints is found. In particular, we find a Gröbner basis for I_{CF}.

35

4. In Section 7.2.4, we construct generators for simpler ideals by combining those of I_{CPC_p} and I_{CF}. This combination preserve the structure that $\mathrm{I}_{\mathcal{C}} = \sum_{p \in J} CPC_p + \mathrm{I}_{CF}$. We will show that a Gröbner basis for I_{Cc} can be computed with bounded degree in polynomial time.

We will show how to convert the ad-hoc algorithm in [6, 8] into a standard PC algorithm. Moreover, techniques from [15] are implemented for the case of the complete and two-fan constraints. This algorithm can be used to solve $\mathrm{IMP}_d(\Gamma)$ in polynomial time and leads to Theorem 5.6.

We define the sets of constraints

$$C_P = \{C_{ij} \in C \,|\, C_{ij} \text{ is a } permutation \text{ constraint}\}$$
$$C_{CF} = \{C_{ij} \in C \,|\, C_{ij} \text{ is a } complete \text{ or a } two\text{-}fan \text{ constraint}\}.$$

Note that that $\mathcal{C} = C_P \cup C_{CF}$. Therefore,

$$\mathrm{I}_{\mathcal{C}} = \mathrm{I}_{C_P} + \mathrm{I}_{C_{CF}}. \tag{32}$$

The idea is to find a set of generators for each addenda in the sum on the RHS and the to combine these polynomials together in a single Gröbner basis for $\mathrm{I}_{\mathcal{C}}$. We recall that for having the identity Eq. (32), by radicality, it is sufficient to have that

$$\mathbf{V}\left(I_{\mathcal{C}}\right) = \mathbf{V}\left(I_{C_P}\right) \cap \mathbf{V}\left(I_{C_{CF}}\right).$$

7.1 Binary constraints

Let Γ be a language over a finite domain D closed under the dual-discriminator polymorphism, i.e. $\nabla \in Pol(\Gamma)$. Let $\mathcal{C} = (X, D, C)$ be an instance of $\mathrm{CSP}(\Gamma)$. In general, if a language Γ is closed under a majority polymorphism μ, any instance of $CSP(\Gamma)$ is equivalent to an instance that has only binary constraints.

Consider a relation R with arity m and let $J \subseteq [m]$ be a set of indices. We denote $X[J] = (x_j)_{j \in J}$ the subset of variables with indices in J, and similarly we denote $pr_J(R)$ the projection of R to the components with indices in J, i.e. the tuples $(a_j)_{j \in J}$ such that there exists a n-tuple $(b_1, \ldots, b_n) \in R$ with $a_j = b_j$ for all $j \in J$.

Proposition 7.2 ([34])**.** *Let R be a relation of arity m that is closed under a majority operation, and let C be any constraint (X, R) constraining the variables in X with relation R. For any problem $\mathcal{P}$ containing the constraint C, the problem $\mathcal{P}'$ obtained by replacing C with the set of binary constraints*

$$\{((X[i], X[j]), pr_{i,j}(R)) \,|\, 1 \le i \le j \le m\}$$

has exactly the same solutions as $\mathcal{P}$.

Moreover, if the language Γ is closed under the dual-discriminator polymorphism, i.e. $\nabla \in Pol(\Gamma)$, the binary constraints C_{ij} are well-structured into three types.

Proposition 7.3. *[20] Suppose $\nabla \in Pol(\Gamma)$. Then each constraints $C_{ij} = \langle (x_i, x_j), R_{ij} \rangle$ is one of the following three types.*

1. *Permutation constraint: $R_{ij} = \{(a, \pi(a)) \,|\, a \in D_i\}$ for some $D_i \subseteq D$ and some bijection $\pi : D_i \to D_j$, where $D_j \subseteq D$.*

2. *Complete constraint: $R_{ij} = D_i \times D_j$ for some $D_i, D_j \subseteq D$.*

3. *Two-fan constraint:* $R_{ij} = \{(\{a\} \times D_j) \cup (D_i \times \{b\})\}$ *for some* $D_i, D_j \subseteq D$ *and* $a \in D_i, b \in D_j$.

Remark 7.4. *We emphasize here that given an instance of a CSP $\mathcal{C} = (X, D, C)$ whose language Γ is dual-discriminator closed, we can derive bivariate polynomials describing the binary constraints C_{ij} in polynomial time entirely within the framework of* PC.

Indeed, first we note that Γ is a finite constraint language. Let $arity(R)$ denote the arity of R. Then $M := \max_{R \in \Gamma} arity(R) = O(1)$, i.e. the maximum arity of a relation in Γ is constant.

Second, for any constraint we can easily find a set of generators for its combinatorial ideal. Let $T = R^T(x_{i_1}, \ldots, x_{i_m})$ be a m-ary constraint from C. Let $\mathcal{P}^T$ be a set of polynomials such that $Sol(T) = \mathbf{V}(\mathcal{P}^T)$ and such that the domain polynomials are in $\mathcal{P}^T$ for each variable appearing in any of the polynomials. Then the combinatorial ideal of T is equal to the ideal generated by $\mathcal{P}^T$, i.e. $I_T = \langle \mathcal{P}^T \rangle$.

Third, since $arity(R^T) \leq M$ is bounded by a constant, it follows that the reduced Gröbner basis of $\langle \mathcal{P}^T \rangle$ can be calculated in constant time with respect to the number n of variables in X. Indeed, finding the reduced Gröbner basis is in general an EXPSPACE-complete problem (see [47, 46]) but only with respect to the number of variables in $\mathcal{P}^T$, which however is bounded by M and so independent from n.

Lastly, we observe that any set of polynomials describing a binary constraint can be derived from the generators of the initial (non-binary) constraint. Indeed, by Proposition 7.2 we can describe the solution set of T by using binary constraints T_{ij}. Let $\mathcal{P}^T \subseteq \mathbb{R}[x_1, \ldots, x_n]$ be a set of polynomial generators such that $Sol(T) = \mathbf{V}(\mathcal{P}^T) \subseteq D^m$ and similarly let $\mathcal{P}^{T_{ij}} \subseteq \mathbb{R}[x_i, x_j]$ such that $Sol(T_{ij}) = \mathbf{V}(\mathcal{P}^{T_{ij}}) \subseteq D^2$. Note that the polynomial rings over which $\mathcal{P}^T$ and $\mathcal{P}^{T_{ij}}$ differ in the variables. Moreover, if $q \in \langle \mathcal{P}^{T_{ij}} \rangle$, then $q = 0$ over $\mathbf{V}(\mathcal{P}^{T_{ij}})$, but if we interpret $q \in \mathbb{R}[x_1, \ldots, x_n]$, then also $q = 0$ over $\mathbf{V}(\mathcal{P}^T)$ and therefore $q \in \langle \mathcal{P}^T \rangle$. Thus, q can be derived by PC *with bounded degree and bounded coefficients from $\mathcal{P}^T$.*

Note that if there are two constraints T^1 and T^2 that contain variables x_i and x_j in their scope, then by Proposition 7.2 the initial CSP *can be equivalently formulated with some binary constraint T_{ij}. By the same reasoning as in the previous case, a set of generators for $\mathcal{P}^{T_{ij}}$ can be derived by combining together polynomials in $\mathcal{P}_{ij}^{T_1}$ and $\mathcal{P}_{ij}^{T_2}$, the restrictions to variables x_i and x_j of $\mathcal{P}^{T_1}$ and $\mathcal{P}^{T_2}$ respectively.*

In light of the above results and remarks, we can assume without loss of generality that all the constraints in the CSP instance $\mathcal{C}$ are binary, and that the generators for the combinatorial ideal $I_{\mathcal{C}}$ are sets of polynomials $\mathcal{P}_{ij}$ with the property $Sol(C_{ij}) = \mathbf{V}(\mathcal{P}_{ij})$. Furthermore, the points of any bivariate variety $\mathbf{V}(\mathcal{P}_{ij})$ arise from either a permutation, a complete or a two-fan constraint.

7.2 Generating sets

In this section, we walk through the proof sketched at the beginning of Section 7. We start by presenting some derivations that can be efficiently made by PC. We will refer these derivations in the subsequent sections. We then proceed to find generating sets for ideals arising from permutation and from complete and two-fan constraints. Finally, we will combine these generators together and find a Gröbner basis of $I_{\mathcal{C}}$.

7.2.1 Derivation schemes

We present and prove five derivation schemes that can be performed by Polynomial Calculus in polynomial time. This subsection serves as a reference for later discussions and can be skipped at a first read.

Throughout this section x will denote a variable and $D_f, D_g, D_h \subseteq D$.

Lemma 7.5 (Derivation Scheme 1). *Let $h = \Pi_{a \in D_h}(x - a)$ and consider polynomials $f = h(x - \alpha)$ and $g = h(x - \beta)$ with $\alpha \neq \beta$. Then h can be PC-derived from f and g in polynomial time.*

Proof. TOPROVE 26 □

Lemma 7.6 (Derivation Scheme 2). *Let $f = \Pi_{a \in D_f}(x - a)$ and $g = \Pi_{b \in D_g}(x - b)$ with $D_f \cap D_g \neq \emptyset$. Then $h = \Pi_{c \in D_f \cap D_g}(x - c)$ can be PC-derived from f and g in polynomial time.*

Proof. TOPROVE 27 □

Lemma 7.7 (Derivation Scheme 3). *Let $f = \Pi_{a \in D_f}(x - a)$ and $g = \Pi_{b \in D_g}(x - b)(x^2 + \alpha)$ with $D_f, D_g \subseteq D$, $D_f \cap D_g \neq \emptyset$ and $\alpha \in \mathbb{R}$. Then $h = \Pi_{c \in D_f \cap D_g}(x - c)$ can be PC-derived from f and g in polynomial time.*

Proof. TOPROVE 28 □

Corollary 7.8 (Derivation Scheme 4). *Let $f = \Pi_{a \in D_f}(x - a)$ and $g = \Pi_{b \in D_g}(x - b)\Pi_{d \in F_g}(x^2 + \beta_d)$ with $D_f, D_g \subseteq D$ and some set of indices F_g, $D_f \cap D_g \neq \emptyset$ and $\beta_d \in \mathbb{R}$. Then $h = \Pi_{c \in D_f \cap D_g}(x - c)$ can be PC-derived from f and g in polynomial time.*

Lemma 7.9 (Derivation Scheme 5). *Let $\sigma_{ij} : D_i \to D_j$ be a bijection with $D_i, D_j \subseteq D$. Let f be the Lagrange interpolating polynomial that simulates $\sigma_{ji} = \sigma_{ij}^{-1}$ over D_j. Consider polynomials $p_1 := x_i - f(x_j)$, $p_2 := x_i - a$ and $p_3 = \Pi_{b \in D_j}(x_j - b)$ for some $a \in D_i$. Then $h = x_j - \sigma_{ij}(a)$ can be PC-derived from p_1 and p_2 in polynomial time.*

Proof. TOPROVE 29 □

7.2.2 Permutation constraints

Let $C_{ij} = R_{ij}(x_i, x_j) \in C_P$ be a permutation constraint where $R_{ij} = \{(a, \pi_{ij}(a) \,|\, a \in D_i\}$ for some $D_i, D_j \subseteq D$ and a bijection $\pi_{ij} : D_i \to D_j$. To the constraint C_{ij} it corresponds the set of polynomials $\{x_j - f(x_i), x_i - g(x_j), \Pi_{a \in D_i}(x_i - a), \Pi_{b \in D_j}(x_j - b)\}$, where f and g are the (Lagrange) polynomials interpolating the points $\{(a, \pi_{ij}(a))\}_{a \in D_i}$ and $\{(\pi_{ij}^{-1}(b), b)\}_{b \in D_j}$ respectively. Recall Remark 7.4, thus all these polynomials can be derived by PC in polynomial time.

Next we use a construction similar to the one in [6, 8] to define larger constraints called *chain permutation constraints* (CPCs) that combine multiple permutation constraints together. We maintain the notation. More precisely, it is possible to define constraints

$$CPC_p := R_p(X_p = \{x_{p_1}, \ldots, x_{p_r}\})$$

such that the solutions to the constraints C_P are also the solutions to the constraints $CPC := \{CPC_p \,|\, p \in J\}$ for some $J \subseteq [n]$. Moreover, the following property holds.

Lemma 7.10 ([6, 8]). *Let $CPC_p = R_p(X_p)$ and $CPC_q = R_q(X_q)$ with $p, q \in J$ be two CPCs. If $p \neq q$, then $X_p \cap X_q = \emptyset$.*

However, we do not really need to calculate any CPC: it suffices for us to derive a set of polynomials $\mathcal{P}^{CPC_p}$ such that $\mathbf{V}\left(\mathcal{P}^{CPC_p}\right) = Sol(CPC_p)$ for any $p \in J$. In order to do so, we define

$$\mathcal{P}^{CPC_p} := \bigcup_{i,j \in J_p} \mathcal{P}_{ij}$$

38

where $J_p \subseteq [n]$ is a set of indices such that for any pair of indices there exists a chain of pairs of indices "connecting" them. More precisely, let $i, j \in J_p$, then there exist pairs

$$\{l_1^1, l_2^1\}, \{l_1^2, l_2^2\}, \{l_1^3, l_2^3\}, \ldots, \{l_1^k, l_2^k\} \subseteq J_p$$

for some $k \in [n]$ such that $i \in \{l_1^1, l_2^1\}, j \in \{l_1^k, l_2^k\}$ and $|\{l_1^w, l_2^w\} \cap \{l_1^{w+1}, l_2^{w+1}\}| = 1$ for any $w = 1, \ldots, k-1$.

Let I_{CPC_p} be the combinatorial ideal associated with $CPC_p = R_p(X_p = \{x_{p_1}, \ldots, x_{p_r}\})$. Let $S_i = pr_i(R_p) \subseteq D$ be the i-th projection of relation R_p, i.e. the set of values x_i can assume for each valid solution in R_p. As a result of the construction of the CPCs, there exist bijections between any pair of variables in X_p. We denote $\sigma_{ij} : S_i \to S_j$ any such bijection between two variables $x_i, x_j \in X_p$.

Lemma 7.11. *Let $\mathcal{P}^{CPC_p}$ be a generating set of I_{CPC_p} for some $p \in J$. If $i, j \in J_p$, then there exist interpolating polynomials f and g simulating the bijections σ_{ij} and σ_{ji} respectively, i.e. $f(a) = \sigma_{ij}(a)$ for all $a \in D_i$ and similarly for g. It follows that $x_j - f(x_i), x_i - g(x_j) \in \langle \mathcal{P}^{CPC_p} \rangle$. Furthermore, $\deg(f), \deg(g) \leq |D| - 1$ and polynomials $x_j - f(x_i)$ and $x_i - g(x_j)$ can be derived by PC in polynomial time.*

Proof. TOPROVE 30 $\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad$ $\square$

Remark 7.12. *There are at most $|D|!$ different bijections between variables in X_p, implying that many variables are actually related by linear polynomials of the type $x_j - x_k$ for some $x_j, x_k \in X_p$. Indeed, consider variable $x_i \in X_p$ and consider all the bijections σ_{il} such that $x_l \in X_p$. Suppose there exist two variables $x_j, x_k \in X_p$ such that $\sigma_{ij} = \sigma_{ik}$. Then $x_j = x_k$ for all values in $S_j = S_k$. It follows that $\sigma_{jk} = \sigma_{kj} = id$. Thus the Lagrange interpolating polynomials f and g are $x_j - x_k$ and $x_k - x_j$ respectively.*

From the above remark it follows that the number of variables which are *not* linearly related is at most $|D|! = O(1)$, while for the remaining variables linear polynomials can be derived from $\mathcal{P}^{CPC_p}$. Finding a Gröbner basis for I_{CPC_p} thus reduces to finding the Gröbner basis of an ideal with at most $|D|!$ variables. Indeed, consider the set

$$\{x_i - x_j \mid x_i > x_j \text{ and } \sigma_{ij} \text{ is the identity function } \forall x_i, x_j \in X_p\}. \tag{33}$$

Let $\mathcal{S}_p$ be the reduced Gröbner basis of Eq. (33). Note that it can be derived by PC with bounded degree and bounded coefficients. Now define

$$M_p := \{LM(s) \mid s \in \mathcal{S}_p\}.$$

Therefore, by radicality it follows

$$\mathrm{I}_{CPC_p} = \langle \mathcal{T}_p \rangle + \langle \mathcal{D}_p \rangle + \langle \mathcal{S}_p \rangle. \tag{34}$$

where $\mathcal{T}_p$ is the set of interpolating polynomials for the bijections different from the identity $\sigma_{ij} \neq id$, and $\mathcal{D}_p = \{\Pi_{a \in S_i}(x_i - a) \mid x_i \in X_p \setminus M_p\}$ is the set of domain polynomials.

At this stage, finding a Gröbner basis for I_{CPC_p} would not be difficult, but we choose to defer this step to Section 7.2.4. As we will see, we will update the sets $\mathcal{D}_p$ so that it is easy to find a Gröbner basis for $\mathcal{T}_p \cup \mathcal{D}_p \cup \mathcal{S}_p \cup G$, where G is a set of generators arising from the complete and two-fan constraints.

7.2.3 Complete and two-fan constraints

We consider the set of constraints C_{CF} comprising of the complete and two-fan constraints. Let $G = \emptyset$. We will add polynomials to G until it represents the constraints in C_{CF}.

A constraint $C_{ij} = R(x_i, x_j)$ is *complete* whenever $R = D_i \times D_j$ with $D_i, D_j \subseteq D$. It is described by a pair of *partial domain polynomials* defined as

$$\Pi_{a \in D_i}(x_i - a), \qquad \Pi_{a \in D_j}(x_j - a).$$

For every complete constraint, we can derive such polynomial as seen in Remark 7.4 and add them to G.

A constraint $C_{ij} = R(x_i, x_j)$ is *two-fan* if $R = \{(\{a\} \times D_j) \cup (D_i \times \{b\})\}$ with $D_i, D_j \subseteq D$, $a \in D_i$ and $b \in D_j$. A two-fan constraint is described by polynomials

$$(x_i - a)(x_j - b), \quad \Pi_{c \in D_i}(x_i - c), \quad \Pi_{d \in D_j}(x_j - d).$$

We also add those to G.

It might happen that there exists a variable x_i for which two partial domain polynomials have been added, say $\Pi_{c \in D_{i_1}}(x_i - c)$ and $\Pi_{d \in D_{i_2}}(x_i - d)$. In this case, we derive by Derivation Scheme 2. the polynomial $\Pi_{c \in D_i}(x_i - c)$ where $D_i = D_{i_1} \cap D_{i_2}$ and replace the two initial partial domain polynomials in G with this new one. If for some variable x_i no partial domain polynomial has been added to G, we add to G the full domain polynomial $\Pi_{a \in D}(x_i - a)$.

Lastly, we observe that we can consider the equivalent $(2,3)$-consistent version $\mathcal{C}' = (X, D, C')$ of the initial CSP $\mathcal{C} = (X, D, C)$. We follow along the algorithm presented in [15]. However, we expand on that result by presenting a PC simulation of the algorithm.

- Repeat until possible: consider three variables $x_i, x_j, x_k \in X$ and consider the set

$$T_{ij,k} = \{(a,b) \in R_{ij} \mid \nexists c \in D \text{ s.t } (a,c) \in R_{ik} \wedge (c,b) \in R_{kj}\}$$

  If $T_{ij,k} \neq \emptyset$ do the following. Let f and g be interpolating polynomials vanishing at R_{ik} and R_{kj} respectively, i.e. $f(\alpha, \beta) = 0$ if and only if $(\alpha, \beta) \in R_{ik}$, and similarly we define g. Note that $\deg(f), \deg(g) = O(|D|^2)$. Define $h(x_i, x_j, x_k) := f(x_i, x_k)g(x_k, x_j)$. Then, by definition, $h(a,b,c) \neq 0$ if $(\alpha, \beta) \in T_{ij,k}$ and for all $c \in D$. It follows that, as done in Remark 7.4, we can derive a bivariate polynomial $\tilde{h}(x_i, x_j)$ such that $\tilde{h}(a,b) = 0$ if and only if $(a,b) \in R_{ij} \setminus T_{ij,k}$. Add $\tilde{h}$ to G.

When the algorithm stops, we consider the CSP generated by the polynomials in G, i.e. for every $x_i, x_j \in X$ we consider the constraint $C' = \mathbf{V}_{ij}(x_i, x_j)$, where $\mathbf{V}_{ij}$ is the variety generated by the polynomials in x_i and x_j. It turns out that C' is the $(2,3)$-consistent version of C. Therefore, C' and C have the same solutions and ∇ is a polymorphism of C'. Moreover, the following holds.

Lemma 7.13. *[56, Lemma 4.1.5] Let G be defined as above. Then G is a Gröbner basis of I_{CF}.*

7.2.4 Combining I_{CPC_p} and I_{CF}

Next, we want to combine the generators of I_{CPC_p} and I_{CF}. For the moment we have

$$I_{\mathcal{C}} = \sum_{p \in J} I_{CPC_p} + I_{CF} = \sum_{p \in J}(\langle \mathcal{T}_p \rangle + \langle \mathcal{D}_p \rangle + \langle \mathcal{S}_p \rangle) + \langle G \rangle. \tag{35}$$

The remainder of this section completes the proof of Theorem 5.6.

Proof. TOPROVE 31 $\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad$ □

8 Conclusions and research directions

In this paper it is shown that for two classes of problems that generalize HORN-SAT and 2-SAT a PC proof of degree d can be found in time $n^{O(d)}$, if it exists (see also [6] for related results). This is obtained by first showing that a (truncated) Gröbner basis for the graded lexicographic order can be computed by PC in polynomial time for any fixed d (and therefore with polynomial bit complexity). By a simple polynomial division argument (see Appendix B), the latter implies that for these two classes there are no bit-complexity issues. Furthermore, both HORN-SAT and 2-SAT, along with their generalizations to finite domains—semilattice and dual-discriminator closed languages, respectively—fit within the framework of bounded width languages [35]. As a step towards understanding the boundary of tractability of the PC criterion, it would be interesting to explore how PC can be applied to solve the $\mathrm{IMP}_d(\Gamma)$ for bounded width languages. Moreover, results regarding the tractability of the IMP_d, even when using restricted form of algorithms such as those encapsulated in the Polynomial Calculus proof system, would be valuable on their own right.

Similar to SoS, it has often been stated that a PC refutation of degree d can be found in time $n^{O(d)}$, if it exists. For PC over finite fields, this is already clear from the algorithm provided in [19]. However, in the case of PC over reals or rationals, the search for proofs can potentially result in bit complexity issues as recently shown by Hakoniemi in [31]. Indeed, in [31] it is shown that there is a set of polynomial constraints Q_n on Boolean variables that has both SoS and PC over rationals refutations of degree 2, but for which any SoS or PC refutation over rationals must have exponential bit-complexity. The author remarks that the constraints in Q_n do not arise from any CNF, and raise the open question to understand whether the two measures of bit-complexity and monomial-size are polynomially equivalent for CNFs. Our PC criterion does not apply to other CNF problems like $3\mathrm{LIN}(2)$, where PC and SoS are known to be not complete for any fixed d. Moreover, we remark that $3\mathrm{LIN}(2)$ problems do not arise from bounded width languages [2]. As an intermediate step for the open question raised in [31], it would be interesting to understand the bit complexity of problems with these CNF constraints.

In this paper, we have made partial advancements in the understanding of the bit complexity of SoS, an issue that has only recently garnered attention and remains in its early stages of research. Since it was first raised 2017, progress has been relatively limited. In this section, we have offered some insights that we hope will stimulate further exploration and enhance our understanding of this fundamental problem.

References

[1] Lorenzo Baldi, Teresa Krick, and Bernard Mourrain. An effective positivstellensatz over the rational numbers for finite semialgebraic sets, 2024. URL: https://arxiv.org/abs/2410.04845, arXiv:2410.04845.

[2] Libor Barto and Marcin Kozik. Constraint satisfaction problems solvable by local consistency methods. *J. ACM*, 61(1), January 2014. doi:10.1145/2556646.

[3] Libor Barto, Andrei Krokhin, and Ross Willard. Polymorphisms, and How to Use Them. In Andrei Krokhin and Stanislav Zivny, editors, *The Constraint Satisfaction Problem: Complexity and Approximability*, volume 7 of *Dagstuhl Follow-Ups*, pages 1–44. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, Dagstuhl, Germany, 2017. doi:10.4230/DFU.Vol7.15301.1.

[4] Paul Beame, Russell Impagliazzo, Jan Krajícek, Toniann Pitassi, and Pavel Pudlák. Lower bound on Hilbert's Nullstellensatz and propositional proofs. In *35th Annual Symposium on Foundations of Computer Science, Santa Fe, New Mexico, USA, 20-22 November 1994*, pages 794–806, 1994.

[5] Christoph Berkholz. The Relation between Polynomial Calculus, Sherali-Adams, and Sum-of-Squares Proofs. In Rolf Niedermeier and Brigitte Vallée, editors, *35th Symposium on Theoretical Aspects of Computer Science (STACS 2018)*, volume 96 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 11:1–11:14, Dagstuhl, Germany, 2018. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik. `doi:10.4230/LIPIcs.STACS.2018.11`.

[6] Arpitha P. Bharathi and Monaldo Mastrolilli. Ideal membership problem for boolean minority and dual discriminator. In Filippo Bonchi and Simon J. Puglisi, editors, *46th International Symposium on Mathematical Foundations of Computer Science, MFCS 2021, August 23-27, 2021, Tallinn, Estonia*, volume 202 of *LIPIcs*, pages 16:1–16:20. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2021. `doi:10.4230/LIPIcs.MFCS.2021.16`.

[7] Arpitha P. Bharathi and Monaldo Mastrolilli. Ideal membership problem over 3-element csps with dual discriminator polymorphism. *SIAM J. Discret. Math.*, 36(3):1800–1822, 2022. `doi:10.1137/21M1397131`.

[8] Arpitha P. Bharathi and Monaldo Mastrolilli. Ideal membership problem for boolean minority and dual discriminator. *SIAM Journal on Discrete Mathematics*, 39(1):206–230, 2025. `doi:10.1137/23M1556010`.

[9] B. Buchberger. *Ein Algorithmus zum Auffinden der Basiselemente des Restklassenringes nach einem nulldimensionalen Polynomideal (An Algorithm for Finding the Basis Elements in the Residue Class Ring Modulo a Zero Dimensional Polynomial Ideal)*. PhD thesis, Mathematical Institute, University of Innsbruck, Austria, 1965. English translation in J. of Symbolic Computation, Special Issue on Logic, Mathematics, and Computer Science: Interactions. Vol. 41, Number 3-4, Pages 475–511, 2006.

[10] Andrei Bulatov. Personal communication, 2023.

[11] Andrei A. Bulatov. A dichotomy theorem for nonuniform CSPs (best paper award). In *58th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2017, Berkeley, CA, USA, October 15-17, 2017*, pages 319–330, 2017.

[12] Andrei A. Bulatov. Constraint satisfaction problems: Complexity and algorithms. *ACM SIGLOG News*, 5(4):4–24, November 2018. `doi:10.1145/3292048.3292050`.

[13] Andrei A. Bulatov and Akbar Rafiey. On the complexity of csp-based ideal membership problems. *CoRR*, abs/2011.03700, 2020. URL: `https://arxiv.org/abs/2011.03700`, `arXiv:2011.03700`.

[14] Andrei A. Bulatov and Akbar Rafiey. The ideal membership problem and abelian groups. In Petra Berenbrink and Benjamin Monmege, editors, *39th International Symposium on Theoretical Aspects of Computer Science, STACS 2022, March 15-18, 2022, Marseille, France (Virtual Conference)*, volume 219 of *LIPIcs*, pages 18:1–18:16. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2022. `doi:10.4230/LIPIcs.STACS.2022.18`.

[15] Andrei A. Bulatov and Akbar Rafiey. On the complexity of csp-based ideal membership problems. In Stefano Leonardi and Anupam Gupta, editors, *STOC '22: 54th Annual ACM SIGACT Symposium on Theory of Computing, Rome, Italy, June 20 - 24, 2022*, pages 436–449. ACM, 2022. `doi:10.1145/3519935.3520063`.

[16] Samuel R Buss. Lower bounds on nullstellensatz proofs via designs. *Proof complexity and feasible arithmetics*, 39:59–71, 1996.

[17] Samuel R. Buss and Toniann Pitassi. Good degree bounds on Nullstellensatz refutations of the induction principle. *J. Comput. Syst. Sci.*, 57(2):162–171, 1998.

[18] Hubie Chen. A rendezvous of logic, complexity, and algebra. *ACM Comput. Surv.*, 42(1):2:1–2:32, December 2009. `doi:10.1145/1592451.1592453`.

[19] Matthew Clegg, Jeff Edmonds, and Russell Impagliazzo. Using the Groebner basis algorithm to find proofs of unsatisfiability. In *Proceedings of the Twenty-Eighth Annual ACM Symposium on the Theory of Computing, 1996*, pages 174–183, 1996.

[20] Martin C. Cooper, David A. Cohen, and Peter G. Jeavons. Characterising tractable constraints. *Artificial Intelligence*, 65(2):347–361, 1994. `doi:10.1016/0004-3702(94)90021-3`.

[21] David A. Cox, John Little, and Donal O'Shea. *Ideals, Varieties, and Algorithms: An Introduction to Computational Algebraic Geometry and Commutative Algebra*. Springer Publishing Company, Incorporated, 4th edition, 2015.

[22] B. A. Davey and H. A. Priestley. *Introduction to Lattices and Order*. Cambridge University Press, 2 edition, 2002.

[23] Jean-Charles Faugère, Patrizia M. Gianni, Daniel Lazard, and Teo Mora. Efficient Computation of Zero-dimensional Gröbner Bases by Change of Ordering. *Journal of Symbolic Computation*, 16(4):329 – 344, 1993. `doi:10.1006/jsco.1993.1051`.

[24] Noah Fleming, Pravesh Kothari, and Toniann Pitassi. Semialgebraic proofs and efficient algorithm design. *Foundations and Trends in Theoretical Computer Science*, 14(1-2):1–221, 2019. `doi:10.1561/0400000086`.

[25] Mariano Gasca and Thomas Sauer. Polynomial interpolation in several variables. *Advances in Computational Mathematics*, 12(4):377–410, March 2000.

[26] Michel X. Goemans and David P. Williamson. Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming. *J. ACM*, 42(6):1115–1145, nov 1995. `doi:10.1145/227683.227684`.

[27] Sander Gribling, Sven Polak, and Lucas Slot. A note on the computational complexity of the moment-sos hierarchy for polynomial optimization. In *Proceedings of the 2023 International Symposium on Symbolic and Algebraic Computation*, ISSAC '23, page 280–288, New York, NY, USA, 2023. Association for Computing Machinery. `doi:10.1145/3597066.3597075`.

[28] Dima Grigoriev. Linear lower bound on degrees of positivstellensatz calculus proofs for the parity. *Theoretical Computer Science*, 259(1):613–622, 2001. `doi:10.1016/S0304-3975(00)00157-2`.

[29] Dima Grigoriev, Edward A. Hirsch, and Dmitrii V. Pasechnik. Complexity of semialgebraic proofs. *Moscow Mathematical Journal*, 2(4):647–679, 2002.

[30] Dima Grigoriev and Nicolai N. Vorobjov Jr. Complexity of null-and positivstellensatz proofs. *Ann. Pure Appl. Log.*, 113(1-3):153–160, 2001. `doi:10.1016/S0168-0072(01)00055-0`.

[31] Tuomas Hakoniemi. Monomial size vs. bit-complexity in sums-of-squares and polynomial calculus. In *36th Annual ACM/IEEE Symposium on Logic in Computer Science, LICS 2021, Rome, Italy, June 29 - July 2, 2021*, pages 1–7. IEEE, 2021. `doi:10.1109/LICS52264.2021.9470545`.

[32] David Hilbert. Ueber die vollen invariantensysteme. *Mathematische Annalen*, 42:313–373, 1893. URL: `http://eudml.org/doc/157652`.

[33] Peter Jeavons. On the algebraic structure of combinatorial problems. *Theoretical Computer Science*, 200(1):185 – 204, 1998. `doi:10.1016/S0304-3975(97)00230-2`.

[34] Peter Jeavons, David Cohen, and Marc Gyssens. Closure properties of constraints. *J. ACM*, 44(4):527–548, July 1997. `doi:10.1145/263867.263489`.

[35] Peter G. Jeavons and Martin C. Cooper. Tractable constraints on ordered domains. *Artificial Intelligence*, 79(2):327–339, 1995. `doi:10.1016/0004-3702(95)00107-7`.

[36] Christopher Jefferson, Peter Jeavons, Martin J. Green, and M. R. C. van Dongen. Representing and solving finite-domain constraint problems using systems of polynomials. *Annals of Mathematics and Artificial Intelligence*, 67(3):359–382, Mar 2013. `doi:10.1007/s10472-013-9365-7`.

[37] Cédric Josz and Didier Henrion. Strong duality in lasserre's hierarchy for polynomial optimization. *Optimization Letters*, 10(1):3–10, January 2016.

[38] Adam Kurpisz, Samuli Leppänen, and Monaldo Mastrolilli. On the hardest problem formulations for the 0/1 lasserre hierarchy. *Math. Oper. Res.*, 42(1):135–143, 2017. `doi:10.1287/MOOR.2016.0797`.

[39] Adam Kurpisz, Samuli Leppänen, and Monaldo Mastrolilli. An unbounded sum-of-squares hierarchy integrality gap for a polynomially solvable problem. *Math. Program.*, 166(1–2):1–17, November 2017. `doi:10.1007/s10107-016-1102-7`.

[40] Adam Kurpisz, Samuli Leppänen, and Monaldo Mastrolilli. Tight sum-of-squares lower bounds for binary polynomial optimization problems. *ACM Trans. Comput. Theory*, 16(1), March 2024. `doi:10.1145/3626106`.

[41] Adam Kurpisz, Samuli Leppänen, and Monaldo Mastrolilli. Sum-of-squares hierarchy lower bounds for symmetric formulations. *Mathematical Programming*, 182(1-2):369 – 397, 2020. Cited by: 4. `doi:10.1007/s10107-019-01398-9`.

[42] Jean B. Lasserre. An explicit exact sdp relaxation for nonlinear 0-1 programs. In Karen Aardal and Bert Gerards, editors, *Integer Programming and Combinatorial Optimization*, pages 293–303, Berlin, Heidelberg, 2001. Springer Berlin Heidelberg.

[43] Monique Laurent. *Sums of Squares, Moment Matrices and Optimization Over Polynomials*, pages 157–270. Springer New York, New York, NY, 2009. `doi:10.1007/978-0-387-09686-5_7`.

[44] Victor Magron, Mohab Safey El Din, and Markus Schweighofer. Algorithms for weighted sum of squares decomposition of non-negative univariate polynomials. *Journal of Symbolic Computation*, 93:200–220, 2019. `doi:10.1016/j.jsc.2018.06.005`.

[45] Monaldo Mastrolilli. The complexity of the ideal membership problem for constrained problems over the boolean domain. *ACM Trans. Algorithms*, 17(4):32:1–32:29, 2021. `doi:10.1145/3449350`.

[46] Ernst W. Mayr. Membership in polynomial ideals over q is exponential space complete. In B. Monien and R. Cori, editors, *STACS 89*, pages 400–406, Berlin, Heidelberg, 1989. Springer Berlin Heidelberg.

[47] Ernst W. Mayr and Albert R. Meyer. The complexity of the word problems for commutative semigroups and polynomial ideals. *Advances in Mathematics*, 46(3):305–329, 1982. `doi:10.1016/0001-8708(82)90048-2`.

[48] Ryan O'Donnell. SOS Is Not Obviously Automatizable, Even Approximately. In Christos H. Papadimitriou, editor, *8th Innovations in Theoretical Computer Science Conference (ITCS 2017)*, volume 67 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 59:1–59:10, Dagstuhl, Germany, 2017. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik. `doi:10.4230/LIPIcs.ITCS.2017.59`.

[49] Marilena Palomba, Lucas Slot, Luis Felipe Vargas, and Monaldo Mastrolilli. Computational complexity of sum-of-squares bounds for copositive programs, 2025. URL: `https://arxiv.org/abs/2501.03698`, `arXiv:2501.03698`.

[50] Dona Papert. Congruence Relations in Semi-Lattices. *Journal of the London Mathematical Society*, s1-39(1):723–729, 01 1964. `arXiv:https://academic.oup.com/jlms/article-pdf/s1-39/1/723/2721805/s1-39-1-723.pdf`, `doi:10.1112/jlms/s1-39.1.723`.

[51] Pablo A. Parrilo. Semidefinite programming relaxations for semialgebraic problems. *Mathematical Programming*, 96(2):293–320, 2003.

[52] Fedor Part, Neil Thapen, and Iddo Tzameret. First-order reasoning and efficient semi-algebraic proofs. In *2021 36th Annual ACM/IEEE Symposium on Logic in Computer Science (LICS)*, pages 1–13, 2021. `doi:10.1109/LICS52264.2021.9470546`.

[53] Gábor Pataki and Aleksandr Touzov. How do exponential size solutions arise in semidefinite programming? *SIAM Journal on Optimization*, 34(1):977–1005, 2024.

[54] G.M. Phillips. *Interpolation and Approximation by Polynomials*. CMS Books in Mathematics. Springer, 2003.

[55] Aaron Potechin. Sum of Squares Lower Bounds from Symmetry and a Good Story. In Avrim Blum, editor, *10th Innovations in Theoretical Computer Science Conference (ITCS 2019)*, volume 124 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 61:1–61:20, Dagstuhl, Germany, 2019. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. `doi:10.4230/LIPIcs.ITCS.2019.61`.

[56] Akbar Rafiey. *Constraint Satisfaction Problems and friends: Symmetries and algorithm design*. PhD thesis, August 2022. Publisher: Simon Fraser University. URL: `https://summit.sfu.ca/item/35622`.

[57] Prasad Raghavendra and Benjamin Weitz. On the Bit Complexity of Sum-of-Squares Proofs. In Ioannis Chatzigiannakis, Piotr Indyk, Fabian Kuhn, and Anca Muscholl, editors, *44th International Colloquium on Automata, Languages, and Programming (ICALP 2017)*, volume 80 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 80:1–80:13, Dagstuhl, Germany, 2017. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik. `doi:10.4230/LIPIcs.ICALP.2017.80`.

[58] Thomas J. Schaefer. The complexity of satisfiability problems. In *Proceedings of the Tenth Annual ACM Symposium on Theory of Computing*, STOC '78, pages 216–226, New York, NY, USA, 1978. ACM. `doi:10.1145/800133.804350`.

[59] Dmitry Sokolov. (Semi)algebraic proofs over ±1 variables. In Konstantin Makarychev, Yury Makarychev, Madhur Tulsiani, Gautam Kamath, and Julia Chuzhoy, editors, *Proccedings of the 52nd Annual ACM SIGACT Symposium on Theory of Computing, STOC 2020, Chicago, IL, USA, June 22-26, 2020*, pages 78–90. ACM, 2020. `doi:10.1145/3357713.3384288`.

[60] Johan Thapper and Stanislav Živný. The limits of sdp relaxations for general-valued csps. *ACM Transactions on Computation Theory (TOCT)*, 10(3):1–22, 2018.

[61] Marc R.C. van Dongen. *Constraints, Varieties, and Algorithms*. PhD thesis, Department of Computer Science, University College, Cork, Ireland, 2002. URL: `http://csweb.ucc.ie/~dongen/papers/UCC/02/thesis.pdf`.

[62] Benjamin Weitz. *Polynomial Proof Systems, Effective Derivations, and their Applications in the Sum-of-Squares Hierarchy*. PhD thesis, EECS Department, University of California, Berkeley, May 2017.

[63] Dmitriy Zhuk. A proof of CSP dichotomy conjecture (best paper award). In *58th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2017, Berkeley, CA, USA, October 15-17, 2017*, pages 331–342, 2017. `doi:10.1109/FOCS.2017.38`.

[64] Dmitriy Zhuk. A proof of the CSP dichotomy conjecture. *J. ACM*, 67(5):30:1–30:78, 2020. `doi:10.1145/3402029`.

A Refutation degree for Horn clauses

Consider the case all clauses are duals of Horn clauses (for simplicity, Horn clauses work out identically), namely at most one variable is negated per clause. We encode these clauses as a set of polynomial identities in a way that preserves their semantics over $\{0,1\}^n$ assignments. Namely, let $\mathcal{C} = C_1 \wedge C_2 \wedge \ldots \wedge C_m$ be a dual Horn clause formula. We encode each clause $C_i = \neg x_{i_1} \vee x_{i_2} \vee \ldots \vee x_{i_k}$ by introducing a polynomial identity $P_i : x_{i_1}(x_{i_2} - 1) \cdots (x_{i_k} - 1) = 0$. The set of $\{0,1\}$ assignments that satisfy the newly introduced set of polynomial identities is exactly the set of satisfying assignments to $\mathcal{C}$.

The refutation by PC works as follows. Take all the variables that are already known to be false, say set F. These variables belong to $I_{\mathcal{C}}$. Consider clause C_i and the corresponding polynomial identity $P_i : x_{i_1}(x_{i_2} - 1) \cdots (x_{i_k} - 1) = 0$. If $\{x_{i_2}, \ldots, x_{i_k}\} \subseteq F$ then it is easy show that $x_{i_1} \in I_{\mathcal{C}}$ since by using the aforementioned polynomial identity P_i we can express x_{i_1} as polynomial combination of the variables in $\{x_{i_2}, \ldots, x_{i_k}\}$.[6] So we have added a new variable to the set F of known false variables. If the set of F covers an entire clause with no negated variables, then we can derive that $1 \in I_{\mathcal{C}}$ and we are done. If at some point, neither is true, by setting all remaining variables to 1 we satisfy all the clauses. So if the Horn clauses were unsatisfiable we find a proof whose degree is at most the degree of the polynomial identities encoding the clauses.

B Complexity of Polynomial Division

Consider the polynomial ring $\mathbb{R}[x_1, \ldots, x_n]$ ordered according to the grlex order, with $x_1 > x_2 > \cdots > x_n$. We will study the complexity of the standard division algorithm for multivariate polynomials (see [21, Section 2]). In particular, we observe next that it is a polynomial time algorithm.

Lemma B.1. *Let $\mathcal{P} = \{p_1, \ldots, p_m\}$ be a set of polynomials in $\mathbb{R}[x_1, \ldots, x_n]$ and consider a polynomial $f \in \mathbb{R}[x_1, \ldots, x_n]$. Assume that $f, p_1, \ldots, p_m$ have degree at most d and bit complexity polynomial in n. Then f can be written as*

$$f = h_1 p_1 + \ldots h_m p_m + r,$$

with $r, h_1, \ldots, h_m$ having bit complexity polynomial in n.

Proof. TOPROVE 32 □

From this lemma it follows immediately that the existence of a "small" Gröbner basis implies that the IMP can be solved efficiently.

Corollary B.2. *Let $\mathcal{G}_{2d} = \{g_1, \ldots, g_s\}$ be a $2d$-truncated Gröbner basis of the polynomial ideal $I \subseteq \mathbb{R}[x_1, \ldots, x_n]$. Consider a polynomial r of degree at most $2d$. If the polynomials $r, g_1, \ldots, g_s$ have bit complexity polynomial in n, then the (search version) of the IMP_{2d} for r can be solved in time polynomial in n.*

Proof. TOPROVE 33 □

---

[6]For example if $P : x_1(x_2 - 1)(x_3 - 1) = 0$ and $x_2, x_3 \in I_{\mathcal{C}}$ then by the polynomial identity P we have that $x_1 = x_1 x_2(1 - x_3) + x_1 x_3$, implying that $x_1 \in I_{\mathcal{C}}$.