The Ideal Membership Problem and Abelian Groups

Andrei A. Bulatov
School of Computing Science, Simon Fraser University, Burnaby, Canada

Akbar Rafiey
School of Computing Science, Simon Fraser University, Burnaby, Canada

Abstract

Given polynomials $f_0, f_1, \ldots, f_k$ the Ideal Membership Problem, IMP for short, asks if f_0 belongs to the ideal generated by $f_1, \ldots, f_k$. In the search version of this problem the task is to find a proof of this fact. The IMP is a well-known fundamental problem with numerous applications, for instance, it underlies many proof systems based on polynomials such as Nullstellensatz, Polynomial Calculus, and Sum-of-Squares. Although the IMP is in general intractable, in many important cases it can be efficiently solved.

Mastrolilli [SODA'19] initiated a systematic study of IMPs for ideals arising from Constraint Satisfaction Problems (CSPs), parameterized by constraint languages, denoted IMP(Γ). The ultimate goal of this line of research is to classify all such IMPs accordingly to their complexity. Mastrolilli achieved this goal for IMPs arising from CSP(Γ) where Γ is a Boolean constraint language, while Bulatov and Rafiey [arXiv'21] advanced these results to several cases of CSPs over finite domains. In this paper we consider IMPs arising from CSPs over "affine" constraint languages, in which constraints are subgroups (or their cosets) of direct products of Abelian groups. This kind of CSPs include systems of linear equations and are considered one of the most important types of tractable CSPs. Some special cases of the problem have been considered before by Bharathi and Mastrolilli [MFCS'21] for linear equation modulo 2, and by Bulatov and Rafiey [arXiv'21] to systems of linear equations over $\mathsf{GF}(p)$, p prime. Here we prove that if Γ is an affine constraint language then IMP(Γ) is solvable in polynomial time assuming the input polynomial has bounded degree.

2012 ACM Subject Classification Mathematics of computing → Combinatoric problems; Mathematics of computing → Gröbner bases and other special bases

Keywords and phrases Polynomial Ideal Membership, Constraint Satisfaction Problems, Polymorphisms, Gröbner Bases, Abelian Groups

Digital Object Identifier 10.4230/LIPIcs.STACS.2022.18

Related Version *Full Version*: `https://arxiv.org/pdf/2201.05218.pdf` [16]

Funding *Andrei A. Bulatov*: Research supported by an NSERC Discovery Grant.
Akbar Rafiey: Research supported by NSERC.

1 Introduction

The Ideal Membership Problem. Representing combinatorial problems by polynomials and then using algebraic techniques to approach them is one of the standard methods in algorithms and complexity. The Ideal Membership Problem (IMP for short) is an important algebraic framework that has been instrumental in such an approach. The IMP underlies many proof systems based on polynomials such as Nullstellensatz, Polynomial Calculus, and Sum-of-Squares, and therefore plays an important role in such areas as proof complexity and approximation.

Let $\mathbb{F}$ be a field and $\mathbb{F}[x_1, \ldots, x_n]$ the ring of polynomials over $\mathbb{F}$. Given polynomials f_0, $f_1, \ldots, f_k \in \mathbb{F}[x_1, \ldots, x_n]$ the IMP asks if f_0 belongs to the ideal $\langle f_1, \ldots, f_k \rangle$ generated by $f_1, \ldots, f_k$. This fact is usually demonstrated by presenting a *proof*, that is, a collection of

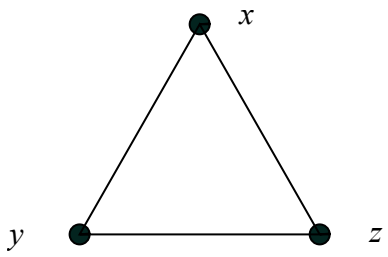

Figure 1 Graph 2-colorability.

polynomials $h_1, \ldots, h_k$ such that the following polynomial identity holds $f_0 = h_1 f_1 + \cdots + h_k f_k$. Many applications require the ability to produce such a proof. We refer to the problem of finding a proof of membership as the *search IMP*. Note that by the Hilbert Basis Theorem any ideal of $\mathbb{F}[x_1, \ldots, x_n]$ can be represented by a finite set of generators meaning that the above formulation of the problem covers all possible ideals of $\mathbb{F}[x_1, \ldots, x_n]$.

The general IMP is a difficult problem and it is not even obvious whether or not it is decidable. The decidability was established in [23, 32, 33]. Then Mayr and Meyer [29] were the first to study the complexity of the IMP. They proved an exponential space lower bound for the membership problem for ideals generated by polynomials with integer and rational coefficients. Mayer [28] went on establishing an exponential space upper bound for the IMP for ideals over $\mathbb{Q}$, thus proving that such IMPs are **EXPSPACE**-complete. The source of hardness here is that a proof that $f_0 \in \langle f_1, \ldots, f_k \rangle$ may require polynomials of exponential degree. In the cases when the degree of a proof has a linear bound in the degree of f_0, the IMP can be solved more efficiently. (There is also the issue of exponentially long coefficients that we will mention later.)

Combinatorial Ideals. To illustrate the connection of the IMP to combinatorial problems we consider the following simple example. We claim that the graph in Fig. 1 is 2-colorable if and only if polynomials $x(1-x), y(1-y), z(1-z), x+y-1, x+z-1, y+z-1$ have a common zero. Indeed, denoting the two possible colors 0 and 1, the first three polynomials guarantee that the only zeroes this collection of polynomials can have are such that $x, y, z \in \{0, 1\}$. Then the last three polynomials make sure that in every common zero the values of x, y, z are pairwise different, and so correspond to a proper coloring of the graph. Of course, the graph in the picture is not 2-colorable, and by the Weak Nullstellensatz this is so if and only if the constant polynomial 1 belongs to the ideal generated by the polynomials above. A proof of that can be easily found

$$1 = (-4)\,[x(x-1)] + (2x-1)\,([x+y-1] - [y+z-1] + [x+z-1])\,.$$

The example above exploits the connection between polynomial ideals and sets of zeroes of polynomials, also known as *affine varieties*. While this connection does not necessarily holds in the general case, as Hilbert's Nullstellensatz requires certain additional properties of ideals, it works for so called *combinatorial ideals* that arise from the majority of combinatorial problems similar to the example above. The varieties corresponding to combinatorial ideals are finite, and the ideals themselves are zero-dimensional and radical. These properties make the IMP significantly easier, in particular, it can be solved in single exponential time [20]. Also, Hilbert's Strong Nullstellensatz holds in this case, which means that if the IMP is restricted to radical ideals, it is equivalent to (negation of) the question: given $f_0, f_1, \ldots, f_k$ does there exist a zero of $f_1, \ldots, f_k$ that is not a zero of f_0.

The special case of the IMP with $f_0 = 1$ has been studied for combinatorial problems in the context of lower bounds on Polynomial Calculus and Nullstellensatz proofs, see e.g. [4, 17, 22]. A broader approach of using polynomials to represent finite-domain constraints has been explored in [18, 26]. Clegg et al., [18], discuss a propositional proof system based on a bounded degree version of Buchberger's algorithm [9] for finding proofs of unsatisfiability. Jefferson et al., [26] use a modified form of Buchberger's algorithm that can be used to achieve the same benefits as the local-consistency algorithms which are widely used in constraint processing.

Applications in other proof systems. The bit complexity of various (semi)algebraic proof systems is another link that connects approximation algorithms and the IMP. As is easily seen, if the degree of polynomials $h_1, \dots, h_k$ in a proof $f_0 = h_1 f_1 + \dots + h_k f_k$ is bounded, their coefficients can be found by representing this identity through a system of linear equations. A similar approach is used in other (semi)algebraic proof systems such as Sum-of-Squares (SOS), in which bounded degree proofs can be expressed through an SDP program. Thus, if in addition to low degree the system of linear equations or the SDP program has a solution that can be represented with a polynomial number of bits (thus having low *bit complexity*), a proof can be efficiently found.

However, O'Donnell [30] proved that low degree of proofs does not necessarily imply its low bit complexity. He presented a collection of polynomials that admit bounded degree SOS proofs of nonnegativity, all such proofs involve polynomials with coefficients of exponential length. This means that the standard methods of solving SDPs such as the Ellipsoid Method would take exponential time to complete. Raghavendra and Weitz [31] suggested some sufficient conditions on combinatorial ideals that guarantee a low bit complexity SOS proof exists whenever a low degree one does. Two of these conditions hold for the majority of combinatorial problems, and the third one is so called k-effectiveness of the IMP part of the proof. In [15], we showed that for problems where the IMP part is of the form IMP(Γ) (to be introduced shortly) only one of the first two conditions remains somewhat nontrivial and k-effectiveness can be replaced with the requirement that a variation of IMP(Γ) is solvable in polynomial time.

The IMP and the CSP. In this paper we consider IMPs that arise from a specific class of combinatorial problems, the Constraint Satisfaction Problem or the CSP for short. In a CSP we are given a set of variables, and a collection of constraints on the values that variables are allowed to be assigned simultaneously. The question in a CSP is whether there is an assignment to variables that satisfies all the constraints. The CSP provides a general framework for a wide variety of combinatorial problems, and it is therefore very natural to study the IMPs that arise from constraint satisfaction problems.

One of the major directions in the CSP research is the study of CSPs in which the allowed types of constraints are restricted. Such restrictions are usually represented by a *constraint language* that is a set of relations or predicates on a fixed set. The CSP parametrized by a constraint language Γ is denoted CSP(Γ).

Mastrolilli in [27] initiated a systematic study of IMPs that arise from problems of the form CSP(Γ), denoted IMP(Γ). More precisely, for a constraint language Γ over domain $D = \{0, \dots, d-1\} \subseteq \mathbb{F}$, in an instance of IMP($\Gamma$) we are given an instance $\mathcal{P}$ of CSP(Γ) with set of variables $X = \{x_1, \dots, x_n\}$, and a polynomial $f_0 \in \mathbb{F}[x_1, \dots, x_n]$. The question is whether or not f_0 belongs to the ideal $\mathcal{I}(\mathcal{P})$ of $\mathbb{F}[x_1, \dots, x_n]$, where the corresponding variety of $\mathcal{I}(\mathcal{P})$ equals the set of solutions of $\mathcal{P}$. Observe, that using Hilbert's Strong Nullstellensatz

the problem can also be reformulated as, whether there exists a solution to $\mathcal{P}$ that is not a zero of f_0. Sometimes we need to restrict the degree of the input polynomial, the IMP in which the degree of f_0 is bounded by d is denoted by $\mathrm{IMP}_d(\Gamma)$.

The complexity of the IMP. The main research question considered in [27] is to classify the problems $\mathrm{IMP}(\Gamma)$ according to their complexity. We [15] showed that in all known cases $\mathrm{IMP}_d(\Gamma)$ can be solved in polynomial time (for any fixed d) if and only if a Gröbner Basis can be efficiently constructed.

Mastrolilli [27] along with Mastrolilli and Bharathi [6] succeeded in characterizing the complexity of $\mathrm{IMP}_d(\Gamma)$ for constraint languages Γ over a 2-element domain. Their results are best presented using the language of polymorphisms. 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 multi-dimensional symmetry of relations from Γ. By $\mathsf{Pol}(\Gamma)$ we denote the set of all polymorphisms of Γ. As for the CSP, polymorphisms of Γ is what determines the complexity of $\mathrm{IMP}(\Gamma)$, see [15].

According to [27, 6], let Γ be a constraint language over $D = \{0, 1\}$ such that the *constant relations* $R_0, R_1 \in \Gamma$, where $R_i = \{(i)\}$. Then $\mathrm{IMP}_d(\Gamma)$ is polynomial time solvable if Γ is invariant under a semilattice or affine operation (of $\mathbb{Z}_2$), the problem $\mathrm{IMP}(\Gamma)$ is polynomial time solvable if Γ is invariant under a majority polymorphism. Otherwise $\mathrm{IMP}_0(\Gamma)$ is **coNP**-complete. This result has been improved in [15] (see also [5, 7]) by showing that $\mathrm{IMP}_d(\Gamma)$ remains polynomial time when Γ has an arbitrary semilattice polymorphism, not only on a 2-element set, an arbitrary dual-discriminator polymorphism, or an affine polymorphism of $\mathbb{Z}_p$, p prime.

Solving the IMP. The IMP is mostly solved using one of the two methods. The first one is the method of finding an IMP or SOS proofs of bounded degree using systems of linear equations or SDP programs. The other approach uses *Gröbner bases* and the standard polynomial division to verify whether a given polynomial has zero remainder when divided by generators of an ideal: if this is the case, the polynomial belongs to the ideal. However, constructing a Gröbner basis is not always feasible, as even though the original generating set is small, the corresponding Gröbner basis may be huge. Note however that to solve the IMP_d it suffices to construct a degree d Gröbner Basis, a.k.a d-truncated Gröbner Basis.

A more sophisticated approach was suggested in [15]. It involves reductions between problems of the form $\mathrm{IMP}(\Gamma)$ before arriving to one for which a Gröbner basis can be constructed in a relatively simple way. Moreover, [15] also introduces a slightly different form of the IMP, called the χIMP, in which the input polynomial has indeterminates as some of its coefficients, and the problem is to find values for those indeterminates (if they exist) such that the resulting polynomial belongs to the given ideal. We showed that χIMP is solvable in polynomial time for every known case of polynomial time solvable IMP, and that χIMP helps to solve the search version of the IMP.

▶ **Theorem 1** ([15]).

(1) *If Γ has a semilattice, dual-discriminator, or the affine polymorphism of $\mathbb{Z}_p$, p prime, then $\chi IMP_d(\Gamma)$ is solvable in polynomial time for every d.*

(2) *If $\chi IMP_d(\Gamma)$ is polynomial time solvable then for every instance $\mathcal{P}$ of $CSP(\Gamma)$ a d-truncated Gröbner basis of $\mathtt{I}(\mathcal{P})$ can be found in polynomial time.*

Our contribution

Affine operations. In this paper we consider IMPs over languages invariant under affine operations of arbitrary finite Abelian groups. This type of constraint languages played an important role in the study of the CSP for three reasons. First, it captures a very natural class of problems. Problems CSP(Γ) where Γ is invariant under an affine operation of a finite field $\mathbb{F}$ can be expressed by systems of linear equations over $\mathbb{F}$ and therefore admit a classic solution algorithm such as Gaussian elimination or coset generation. In the case of a general Abelian group $\mathbb{A}$ the connection with systems of linear equations is more complicated, although it is still true that every instance of CSP(Γ) in this case can be thought of as a system of linear equations with coefficients from some ring – the ring of endomorphisms of $\mathbb{A}$.

Second, it has been observed that there are two main algorithmic approaches to solving the CSP. The first one is based on the local consistency of the problem. CSPs that can be solved solely by establishing some kind of local consistency are said to have *bounded width* [14, 2]. The property to have bounded width is related to a rather surprising number of other seemingly unrelated properties, see e.g. [1, 34]. CSP algorithms of the second type are based on the *few subalgebras* property and achieve results similar to those of Gaussian elimination: they construct a concise representation of the set of all solutions of a CSP [11, 24]. Problems CSP(Γ) where Γ has an affine polymorphism were pivotal in the development of few subpowers algorithms, and, in a sense, constitute the main nontrivial case of them. Among the existing results on the IMP, IMP(Γ) for Γ invariant under a semilattice or majority polymorphism belong to the local consistency part of the algorithmic spectrum, while those for Γ invariant with respect to an affine operation are on the "few subalgebras" part of it. It is therefore important to observe differences in approaches to the IMP in these two cases.

Third, the few subalgebras algorithms [11, 24] when applied to systems of linear equations serve as an alternative to Gaussian elimination that also work in a more general situation and are less sensitive to the algebraic structure behind the problem. There is, therefore, a hope that studying IMPs with an affine polymorphism may teach us about proof systems that use the IMP and do not quite work in the affine case.

The main result of this paper is

▶ **Theorem 2.** *Let $\mathbb{A}$ be an Abelian group and Γ a constraint language such that the affine operation $x - y + z$ of $\mathbb{A}$ is a polymorphism of Γ. Then $IMP_d(\Gamma)$ can be solved in polynomial time for any d. Moreover, given an instance $(f_0, \mathcal{P})$ of $IMP_d(\Gamma)$ a (d-truncated) Gröbner basis of $\mathtt{I}(\mathcal{P})$ can be constructed in polynomial time.*

The tractability of affine IMPs. In [6, 7, 15, 27] IMPs invariant under an affine polymorphism are represented as systems of linear equations that are first transformed to a reduced row-echelon form using Gaussian elimination, and then further converted into a Gröbner basis of the corresponding ideal. If Γ is a constraint language invariant under the affine operation of a general Abelian group $\mathbb{A}$, none of these three steps work: an instance generally cannot be represented as a system of linear equations, Gaussian elimination does not work on systems of linear equations over an arbitrary Abelian group, and a reduced row-echelon form cannot be converted into a Gröbner basis. We therefore need to use a completely different approach, see Section 4. Given an instance $(f_0, \mathcal{P})$ of IMP(Γ) we use the Fundamental Theorem of Abelian groups and a generalized version of pp-interpretations for the IMP [15] to reduce $(f_0, \mathcal{P})$ to an instance $(f_0', \mathcal{P}')$ of *multi-sorted* IMP(Δ) (see below), in which every variable takes values from a set of the form $\mathbb{Z}_{p^\ell}$, p prime. Then we replace the

domains $\mathbb{Z}_{p^\ell}$ of $(f'_0, \mathcal{P}')$ by sets of roots of unity that allows for a more concise representation of polynomials. Finally, we show that a (truncated) Gröbner Basis for the resulting problem can be efficiently constructed.

Multi-sorted IMPs. In order to prove Theorem 2 we introduce two techniques new to the IMP research, although the first one has been extensively used for the CSP. The first technique is multi-sorted problems mentioned above, where every variable has its own domain of values. This framework is standard for the CSP, and also works very well for the IMP, as long as the domain of each variable can be embedded into the field of real or complex numbers. However, many concepts used in proofs and solution algorithms such as pp-definitions, pp-interpretations, polymorphisms have to be significantly adjusted, and several existing results have to be reproved in this more general setting. However, in spite of this extra work, the multi-sorted IMP may become the standard framework in this line of research.

A general approach to χIMP. In [15] we introduced χIMP, a variation of the IMP, in which given a CSP instance $\mathcal{P}$ and a polynomial f_0 some of whose coefficients are unknown, the goal is to find values of the unknown coefficients such that the resulting polynomial f'_0 belongs to $\mathcal{I}(\mathcal{P})$; or report such values do not exist. This framework has been instrumental in finding a Gröbner basis and therefore solving the search version of the IMPs mentioned earlier, as well as in establishing connections between the IMP and other proof systems such as SOS. We again use χIMP to prove the second part of Theorem 2. In order to do that we improve the approach in two ways. First, we adapt it for multi-sorted problems. Second, while in [15] reductions for χIMP are proved in an ad hoc manner, here we develop a unifying construction based on substitution reductions that covers all the useful cases so far.

2 Preliminaries

Ideals and varieties. We follow the same notation and terminology as [15, 19, 27]. Let $\mathbb{F}$ denote an arbitrary field and $\mathbb{F}[x_1, \ldots, x_n]$ be the ring of polynomials over the field $\mathbb{F}$ and indeterminates $x_1, \ldots, x_n$. Sometimes it will be convenient not to assume any specific ordering or names of the indeterminates. In such cases we use $\mathbb{F}[X]$, where X is a set of indeterminates, and treat points in $\mathbb{F}^X$ as mappings $\varphi : X \to \mathbb{F}$. The value of a polynomial $f \in \mathbb{F}[X]$ is then written as $f(\varphi)$. The *ideal* of $\mathbb{F}[x_1, \ldots, x_n]$ generated by a finite set of polynomials $\{f_1, \ldots, f_m\}$ in $\mathbb{F}[x_1, \ldots, x_n]$ is defined as $\langle f_1, \ldots, f_m \rangle \stackrel{\text{def}}{=} \left\{ \sum_{i=1}^{m} t_i f_i \mid t_i \in \mathbb{F}[x_1, \ldots, x_n] \right\}$. For a set of points $S \subseteq \mathbb{F}^n$ its *vanishing ideal* is the set of polynomials defined as

$$\mathbf{I}(S) \stackrel{\text{def}}{=} \{f \in \mathbb{F}[x_1, \ldots, x_n] \mid f(a_1, \ldots, a_n) = 0 \ \ \forall (a_1, \ldots, a_n) \in S\}.$$

For an ideal $\mathcal{I} \subseteq \mathbb{F}[x_1, \ldots, x_n]$ its *affine variety* is the set of common zeros of all the polynomials in $\mathcal{I}$. This is denoted by $\mathbf{V}(\mathcal{I})$ and is formally defined as

$$\mathbf{V}(\mathcal{I}) = \{(a_1, \ldots, a_n) \in \mathbb{F}^n \mid f(a_1, \ldots, a_n) = 0 \quad \forall f \in \mathcal{I}\}.$$

The (multi-sorted) CSP. In the majority of theoretical studies of the CSP all variables are assumed to have the same domain, this type of CSPs are known as *one-sorted CSPs*. However, for various purposes, mainly for more involved algorithms such as in [10, 35] one might consider CSPs where different variables of a CSP have different domains, this type of CSPs are known as *multi-sorted CSPs* [12]. Definitions below are from [12].

▶ **Definition 3.** *For any finite collection of finite domains $\mathcal{D} = \{D_t \mid t \in T\}$, and any list of indices $(t_1, t_2, \ldots, t_m) \in T^m$, a subset R of $D_{t_1} \times D_{t_2} \times \cdots \times D_{t_m}$, together with the list $(t_1, t_2, \ldots, t_m)$, is called a* multi-sorted *relation over $\mathcal{D}$ with arity m and signature $(t_1, t_2, \ldots, t_m)$. For any such relation R, the signature of R is denoted $\sigma(R)$.*

As an example consider $\mathcal{D} = \{D_1, D_2\}$ with $D_1 = \{0, 1\}$, $D_2 = \{0, 1, 2\}$. Then $\mathbb{Z}_6$, which is the direct sum of $\mathbb{Z}_2$ and $\mathbb{Z}_3$, $\mathbb{Z}_2 \oplus \mathbb{Z}_3$, can be viewed as a multi-sorted relation over $\mathcal{D}$ of arity 2 with signature $(1, 2)$.

Given any set of multi-sorted relations, we can define a corresponding class of multi-sorted CSPs. Let Γ be a set of multi-sorted relations over a collection of sets $\mathcal{D} = \{D_t \mid t \in T\}$. The multi-sorted constraint satisfaction problem over Γ, denoted MCSP(Γ), is defined to be the decision problem with instance $\mathcal{P} = (X, \mathcal{D}, \delta, \mathcal{C})$, where X is a finite set of variables, $\delta : X \to T$, and $\mathcal{C}$ is a set of constraints where each constraint $C \in \mathcal{C}$ is a pair $\langle \mathbf{s}, R \rangle$, so that

- $\mathbf{s} = (x_1, \ldots, x_{m_C})$ is a tuple of variables of length m_C, called the constraint scope;
- R is from Γ with arity m_C and signature $(\delta(x_1), \ldots, \delta(x_{m_c}))$, called the constraint relation.

The goal is to decide whether or not there exists a solution, i.e. a mapping $\varphi : X \to \cup_{D \in \mathcal{D}} D$, with $\varphi(x) \in D_{\delta(x)}$, satisfying all of the constraints. We will use $\mathsf{Sol}(\mathcal{P})$ to denote the (possibly empty) set of solutions of the instance $\mathcal{P}$.

The ideal-CSP correspondence. For an instance $\mathcal{P} = (X, \mathcal{D}, \delta, \mathcal{C})$ of MCSP(Γ) we wish to map $\mathsf{Sol}(\mathcal{P})$ to an ideal $\mathcal{I}(\mathcal{P}) \subseteq \mathbb{F}[X]$ ($\mathbb{F}$ is supposed to contain $\cup_{D \in \mathcal{D}} D$, and therefore usually is considered to be a numerical field) such that $\mathsf{Sol}(\mathcal{P}) = \mathbf{V}(\mathcal{I}(\mathcal{P}))$. The (radical) ideal $\mathcal{I}(\mathcal{P})$ of $\mathbb{F}[x_1, \ldots, x_n]$ whose corresponding variety equals the set of solutions of $\mathcal{P}$ is constructed as follows. First, for every x_i the ideal $\mathcal{I}(\mathcal{P})$ contains a *domain* polynomial $f_{\mathcal{D}}(x_i) = \prod_{a \in D_{\delta(x_i)}} (x_i - a)$ whose zeroes are precisely the elements of $D_{\delta(x_i)}$ (this ensures that $\mathcal{I}(\mathcal{P})$ is radical). Then for every constraint $R(x_{i_1}, \ldots, x_{i_k})$, where R is a predicate on $\mathcal{D}$, the ideal $\mathcal{I}(\mathcal{P})$ contains a polynomial $f_R(x_{i_1}, \ldots, x_{i_k})$ that interpolates R, that is, for $(x_{i_1}, \ldots, x_{i_k})$ it holds $f_R(x_{i_1}, \ldots, x_{i_k}) = 0$ if and only if $R(x_{i_1}, \ldots, x_{i_k})$ is true. This model generalizes a number of constructions used in the literature to apply Nullstellensatz or SOS proof systems to combinatorial problems, see, e.g., [4, 17, 22, 31]. If $\mathcal{D} = \{D\}$ in the above definitions then we obtain the definitions for the one-sorted CSP and IMP. Moreover, as observed for the one-sorted case [27, 15], due to the presence of domain polynomials we have $\mathbf{V}(\mathcal{I}(\mathcal{P})) = \emptyset \Leftrightarrow 1 \in \mathcal{I}(\mathcal{P}) \Leftrightarrow \mathcal{I}(\mathcal{P}) = \mathbb{F}[X]$.

In the general Ideal Membership Problem we are given an ideal $\mathcal{I} \subseteq \mathbb{F}[x_1, \ldots, x_n]$, usually by some finite generating set, and a polynomial f_0. The question then is to decide whether or not $f_0 \in \mathcal{I}$. If $\mathcal{I}$ is given through a CSP instance, we can be more specific.

▶ **Definition 4.** *The* Ideal Membership Problem *associated with a constraint language Γ over a set $\mathcal{D}$ is the problem IMP(Γ) in which the input is a pair $(f_0, \mathcal{P})$ where $\mathcal{P} = (X, \mathcal{D}, \delta, \mathcal{C})$ is a MCSP(Γ) instance and f_0 is a polynomial from $\mathbb{F}[X]$. The goal is to decide whether f_0 lies in the ideal $\mathtt{I}(\mathcal{P})$. We use IMP$_d$(Γ) to denote IMP(Γ) when the input polynomial f_0 has degree at most d.*

We say that IMP(Γ) is *tractable* if it can be solved in polynomial time, and IMP(Γ) is *d-tractable* if IMP$_d$(Γ) can be solved in polynomial time for every d.

IMP and Gröbner Bases. The Gröbner Basis G of an ideal is a set of generators with some particular properties that allow for efficient solving of the IMP. If we restrict ourselves to the polynomials of degree at most d then we obtain a *d-truncated Gröbner Basis*. The d-truncated Gröbner Basis G_d of G is defined as $G_d = G \cap \mathbb{F}[x_1, \ldots, x_n]_d$ where $\mathbb{F}[x_1, \ldots, x_n]_d$

denotes the subset of polynomials of degree at most d. To solve IMP_d it suffices to compute a d-truncated Gröbner Basis. This is because, for the input polynomial f_0 of degree d, the only polynomials from G that can possibly divide f_0 are those from G_d. Moreover, the remainders of such divisions have degree at most d.

3 Multi-sorted CSPs and IMP

We study multi-sorted CSPs in the context of the IMP and provide a reduction for multi-sorted languages that are pp-interpretable. This in particular is useful in this paper as it provides a reduction between languages that are invariant under an affine polymorphism over an arbitrary Abelian group and languages over several cyclic p-groups.

3.1 Primitive-positive definability and interpretability

Primitive-positive (pp-) definitions have proved to be instrumental in the study of the CSP [25, 13] and of the IMP as well [15]. Here we introduce the definition of pp-definitions and the more powerful construction, pp-interpretations, in the multi-sorted case, and prove that, similar to the one-sorted case [15], they give rise to reductions between IMPs.

▶ **Definition 5** (pp-definability). *Let Γ be a multi-sorted constraint language on a collection of sets $\mathcal{D} = \{D_t \mid t \in T\}$. A* primitive-positive (pp-) formula *in the language Γ is a first order formula L over variables X that uses predicates from Γ, equality relations, existential quantifier, and conjunctions, and satisfies the condition:*

- *if $R_1(x_1, \ldots, x_k), R_2(y_1, \ldots, y_\ell)$ are atomic formulas in L with signatures σ_1, σ_2 and such that x_i, y_j is the same variable, then $\sigma_1(i) = \sigma_2(j)$.*

The condition above determines the signature $\sigma : X \to T$ of L.

Let Δ be another multi-sorted language over $\mathcal{D}$. We say that Γ pp-defines Δ (or Δ is pp-definable from Γ) if for each (k-ary) relation (predicate) $R \in \Delta$ there exists a pp-formula L over variables $\{x_1, \ldots, x_m, x_{m+1}, \ldots, x_{m+k}\}$ such that

$$R(x_{m+1}, \ldots, x_{m+k}) = \exists x_1 \ldots \exists x_m L,$$

and if σ, σ' are the signatures of L and R, respectively, then $\sigma' = \sigma_{|_{\{m+1,\ldots,m+k\}}}$.

Multi-sorted primitive-positive (pp-) interpretations are also similar to the one-sorted case [15], but require a bit more care.

▶ **Definition 6** (pp-interpretability). *Let Γ, Δ be multi-sorted constraint languages over finite collections of sets $\mathcal{D} = \{D_t \mid t \in T\}, \mathcal{E} = \{E_s \mid s \in S\}$, respectively, and Δ is finite. We say that Γ pp-interprets Δ if for every $s \in S$ there exist $i_{s,1}, \ldots, i_{s,\ell_s} \in T$, a set $F_s \subseteq D_{i_{s,1}} \times \cdots \times D_{i_{s,\ell_s}}$, and an onto mapping $\pi_s : F_s \to E_s$ such that Γ pp-defines the following relations*

1. *the relations F_s, $s \in S$,*
2. *the π_s-preimage of the equality relations on E_s, $s \in S$, and*
3. *the π-preimage of every relation in Δ,*

where by the π-preimage of a k-ary relation $Q \subseteq E_{s_1} \times \cdots \times E_{s_k}$ over $\mathcal{E}$ we mean the m-ary relation $\pi^{-1}(Q)$ over $\mathcal{D}$, with $m = \sum_{i=1}^{k} \ell_{s_i}$, defined by

$$\pi^{-1}(Q)(x_{1,1}, \ldots, x_{1,\ell_{s_1}}, x_{2,1}, \ldots, x_{2,\ell_{s_2}}, \ldots, x_{k,1}, \ldots, x_{k,\ell_{s_k}}) \qquad \text{is true}$$

if and only if

$$Q(\pi_{s_1}(x_{1,1}, \ldots, x_{1,\ell_{s_1}}), \ldots, \pi_{s_k}(x_{k,1}, \ldots, x_{k,\ell_{s_k}})) \qquad \text{is true.}$$

▶ **Example 7.** Suppose $\mathcal{D} = \{\mathbb{Z}_2, \mathbb{Z}_3\}$ and $\mathcal{E} = \{\mathbb{Z}_6\}$. Now, any relation on $\mathcal{E}$ is pp-interpretable in a language in $\mathcal{D}$ via $F = \mathbb{Z}_2 \times \mathbb{Z}_3$ and $\pi : F \to \mathbb{Z}_6$ as $\pi(0,0) = 0,\ \pi(1,2) = 1,\ \pi(0,1) = 2,\ \pi(1,0) = 3,\ \pi(0,2) = 4,\ \pi(1,1) = 5$.

As in the one-sorted case, pp-definitions and pp-interpretations give rise to reductions between IMPs. The proof of the following theorem is similar to that of Theorems 3.11 and 3.15 in [15].

▶ **Theorem 8.** *Let Γ, Δ be multi-sorted constraint languages over collections of sets $\mathcal{D} = \{D_t \mid t \in T\}, \mathcal{E} = \{E_s \mid s \in S\}$, respectively.*

(1) *If Γ pp-defines Δ, then IMP(Δ) [IMP$_d$(Δ)] is polynomial time reducible to IMP(Γ) [respectively, to IMP$_d$(Γ)]*

(2) *If Γ pp-interprets Δ, then IMP$_d$(Δ) is polynomial time reducible to IMP$_{O(d)}$(Γ).*

3.2 Polymorphisms and multi-sorted polymorphisms

One of the standard methods to reason about constraint satisfaction problems is to use polymorphisms. Here we only give the necessary basic definitions. For more details the reader is referred to [3, 13]. Let R be an (n-ary) relation on a set D and f a (k-ary) operation on the same set, that is, $f : D^k \to D$. Operation f is said to be a *polymorphism* of R, or R is *invariant* under f, if for any $\mathbf{a}_1, \dots, \mathbf{a}_k \in R$ the tuple $f(\mathbf{a}_1, \dots, \mathbf{a}_k)$ belongs to R, where f is applied component-wise, that is,

$$f(\mathbf{a}_1, \dots, \mathbf{a}_k) = (f(a_{1,1}, \dots, a_{1,k}), \dots, f(a_{n,1}, \dots, a_{n,k})),$$

and $\mathbf{a}_i = (a_{1,i}, \dots, a_{n,i})$. The set of all polymorphisms of R is denoted $\mathsf{Pol}(R)$. For a constraint language Γ by $\mathsf{Pol}(\Gamma)$ we denote the set of all operations that are polymorphisms of every relation from Γ.

Polymorphisms provide a link between constraint languages and relations pp-definable in those languages. That is for a constraint language Γ and relation R on set A, the relation R is pp-definable in Γ if and only if $\mathsf{Pol}(\Gamma) \subseteq \mathsf{Pol}(R)$ [8, 21].

▶ **Corollary 9** ([25, 15])**.** *Let Γ, Δ be constraint languages on a set D, Δ finite, and $\mathsf{Pol}(\Gamma) \subseteq \mathsf{Pol}(\Delta)$. Then CSP($\Delta$) is polynomial time reducible to CSP(Γ), and IMP(Δ) is polynomial time reducible to IMP(Γ).*

We will need a version of polymorphisms adapted to multi-sorted relations. Let $\mathcal{D} = \{D_t \mid t \in T\}$ be a collection of sets. A multi-sorted operation on $\mathcal{D}$ is a *functional symbol* f with associated *arity* k along with an interpretation f^{D_t} of f on every set $D_t \in \mathcal{D}$, which is a k-ary operation on D_t. A multi-sorted operation f is said to be a *(multi-sorted) polymorphism* of a multi-sorted relation $R \subseteq D_{t_1} \times \cdots \times D_{t_n}$, $t_1, \dots, t_n \in T$, if for any $\mathbf{a}_1, \dots, \mathbf{a}_k \in R$ the tuple

$$f(\mathbf{a}_1, \dots, \mathbf{a}_k) = (f^{D_{t_1}}(a_{1,1}, \dots, a_{1,k}), \dots, f^{D_{t_n}}(a_{n,1}, \dots, a_{n,k})) \in R.$$

▶ **Example 10.** Note that for the sake of defining a multi-sorted operation, the collection $\mathcal{D}$ does not have to be finite. Let $\mathcal{A}$ be the class of all finite Abelian groups and f a ternary functional symbol that is interpreted as the affine operation $f^{\mathbb{A}}(x, y, z) = x - y + z$ on every $\mathbb{A} \in \mathcal{A}$, where $+, -$ are operations of $\mathbb{A}$. Consider the multi-sorted binary relation $R \subseteq \mathbb{Z}_2 \times \mathbb{Z}_4$ over $\mathcal{D} = \{\mathbb{Z}_2, \mathbb{Z}_4\}$ given by $R = \{(0,1), (0,3), (1,0), (1,2)\}$. It is straightforward to verify that f is a polymorphism of R. For instance,

$$f\left(\begin{pmatrix}0\\1\end{pmatrix}, \begin{pmatrix}1\\0\end{pmatrix}, \begin{pmatrix}1\\2\end{pmatrix}\right) = \begin{pmatrix}0-1+1\\1-0+2\end{pmatrix} = \begin{pmatrix}0\\3\end{pmatrix} \in R.$$

To make sure f is a polymorphism of R we of course have to check every combination of pairs from R.

The connection between multi-sorted polymorphisms and pp-definitions is more complicated than that in the one-sorted case [12], and we do not need it here.

4 CSPs and IMPs over Abelian groups

In this section we outline a proof of our main result, Theorem 11.

▶ **Theorem 11.** *Let $\mathbb{A}$ be an Abelian group. Then $IMP_d(\Delta)$ is polynomial time decidable for any d and any finite constraint language Δ which is invariant under the affine operation of $\mathbb{A}$. Moreover, a proof of membership for $IMP_d(\Delta)$ can also be found in polynomial time.*

Let $\mathbb{A}$ be an Abelian group and Δ a constraint language invariant with respect to the operation $x-y+z$ of $\mathbb{A}$. We first show how a given instance $\mathcal{P}$ of $\mathrm{CSP}(\Delta)$ can be transformed in such a way that a Gröbner Basis of the resulting instance can be constructed. Then we use substitution reductions to extend this reduction to instances of $\mathrm{IMP}_d(\Delta)$.

Step 1: Reduction to a multisorted language over cyclic groups. As was mentioned in the introduction, the standard way to solve $\mathrm{CSP}(\Delta)$ and $\mathrm{IMP}_d(\Delta)$ for languages over $\mathbb{Z}_p$ is to represent instances as a system of linear equations. However, it is not always possible for general Abelian groups. For example, the relation R below over $\mathbb{Z}_2 \times \mathbb{Z}_2$ cannot be represented by a system of linear equations with coefficients from $\mathbb{Z}_2$. This is because there are only 8 linear equations over $\mathbb{Z}_2$ with two variables, and the pairs from R only satisfy the trivial one $0x + 0y = 0$, however, R is nontrivial.

$$R = \begin{pmatrix} (0,0) & (1,0) & (0,1) & (1,1) \\ (0,0) & (0,1) & (1,0) & (1,1) \end{pmatrix} \begin{matrix} \leftarrow x \\ \leftarrow y \end{matrix} \tag{1}$$

By the Fundamental Theorem of Abelian groups, $\mathbb{A}$ is a direct sum $\mathbb{Z}_{t_1} \oplus \cdots \oplus \mathbb{Z}_{t_s}$ where each t_i is a prime power and $\mathbb{Z}_{t_i}$ is a cyclic group of order t_i. Using this fact we construct a multisorted constraint language Γ over $\mathbb{Z}_{t_1}, \ldots, \mathbb{Z}_{t_s}$ such that Γ pp-interprets Δ and Γ is invariant under the (multisorted) operation $x - y + z$ of $\mathbb{Z}_{t_1}, \ldots, \mathbb{Z}_{t_s}$. Moreover, the construction can be amended in such a way that we may assume that t_i, t_j are relatively prime for any $i \neq j$. (However, in this case the direct sum of $\mathbb{Z}_{t_1}, \ldots, \mathbb{Z}_{t_s}$ is no longer $\mathbb{A}$.) The following example illustrates the construction.

▶ **Example 12.** Applying such a transformation to the relation R from equation (1) above, every element of $\mathbb{Z}_2 \times \mathbb{Z}_2$ is replaced with a pair of elements of $\mathbb{Z}_2$ in the straightforward way, and R itself is replaced with the 4-ary relation $R' = \{(0,0,0,0), (1,0,0,1), (0,1,1,0), (1,1,1,1)\}$.

By Theorem 8 we have

▶ **Lemma 13.** *For any d the problem $IMP_d(\Delta)$ is polynomial time reducible to $IMP_{O(d)}(\Gamma)$.*

Step 2: Decomposition of multisorted constraints. Fix an instance $\mathcal{P}$ of $\mathrm{CSP}(\Gamma)$. By the following result we may assume that every constraint of Γ is over variables of the same sort.

▶ **Proposition 14.** *Let $\mathcal{P}$ be an instance of $CSP(\Gamma)$, where Γ is a multi-sorted constraint language over $\mathcal{D} = \{\mathbb{Z}_{t_1}, \ldots, \mathbb{Z}_{t_s}\}$ invariant with respect to the affine polymorphism of $\mathbb{Z}_{t_1}, \ldots, \mathbb{Z}_{t_s}$, where $t_1, \ldots, t_s$ are relatively prime. Then $\mathcal{P}$ is equivalent to $\mathcal{P}'$ such that for every constraint $\langle \mathbf{s}, R \rangle$ of $\mathcal{P}'$, the variables in $\mathbf{s}$ are of the same sort. Moreover, the set of variables X of $\mathcal{P}'$ is the same as that of $\mathcal{P}$ and for any $x \in X$ its sort is the same in both $\mathcal{P}$ and $\mathcal{P}'$.*

Step 3: Constructing a system of linear equations. Step 2 allows us to consider only constraints over $\mathbb{Z}_{p^m}$, p prime. Generally, such relations cannot be represented by a system of linear equations of the form we need, i.e., reduced to a row-echelon form. However, it is possible if new variables are allowed.

▶ **Lemma 15.** *Let R be an n-ary relation invariant under the affine operation of $\mathbb{Z}_{p^m}$. Then there are k and $\alpha_{ij} \in \mathbb{Z}_{p^m}$, $i \in [n], j \in [k]$, such that*

$$R = \{(x_1, \dots, x_n) \mid x_i = \alpha_{i1}y_1 + \dots + \alpha_{ik}y_k, \text{ for } i \in [n], y_1, \dots, y_k \in \mathbb{Z}_{p^m}\}.$$

Lemma 15 allows us to represent an instance of $\mathrm{IMP}(\Gamma)$ as a system of linear equations as follows.

▶ **Proposition 16.** *Every instance $(f_0, \mathcal{P})$ of $IMP_d(\Delta)$ can be transformed to an instance $(f_0', \mathcal{P}')$ of $IMP_{O(d)}(\Gamma)$ satisfying the following conditions and such that $f_0 \in \mathtt{I}(\mathcal{P})$ if and only if $f_0' \in \mathtt{I}(\mathcal{P}')$.*

(1) *For every $i \in [s]$ there is a set $Y_i = \{y_{i,1}, \dots, y_{i,r_i}\}$ of variables of $\mathcal{P}'$ and $Y_i \cap Y_j = \emptyset$ for $i \neq j$.*

(2) *For every constraint $\langle \mathbf{s}, R\rangle$ of $\mathcal{P}'$ the following conditions hold:*

- **(a)** *there is $i \in [s]$ such that $\mathbb{Z}_{p_i^{m_i}}$ is the domain of every variable from $\mathbf{s}$;*
- **(b)** *R is represented by a system of linear equations of the form $x_j = \alpha_1 y_{i,1} + \dots + \alpha_{r_i} y_{i,r_i}$, $x_j \in \mathbf{s}$, over $\mathbb{Z}_{p_i^{m_i}}$.*

Let $\mathcal{L}_i$ denote the collection of all equations constructed in Proposition 16 for constraints over $\mathbb{Z}_{p_i^{m_i}}$.

▶ **Example 17.** The relation R' from Example 12 can be represented by the following system of linear equations that uses two extra parameters y_1, y_2:

$$x_1 = y_1, \quad x_2 = y_2, \quad x_3 = y_2, \quad x_4 = y_1.$$

Step 4: Reduction to roots of unity. Using Proposition 16 we can construct a Gröbner Basis of instance $\mathcal{P}$ of $\mathrm{CSP}(\Gamma)$ as follows. Note first of all that a system of linear equations over $\mathbb{Z}_{p_i^{m_i}}$ can be solved in polynomial time. This immediately tells us if $1 \in \mathcal{I}(\mathcal{P})$ or not, and we proceed only if $1 \notin \mathcal{I}(\mathcal{P})$. Let $x_{1,1}, \dots, x_{1,k_1}, \dots, x_{s,1}, \dots, x_{s,k_s}$ and $y_{1,1}, \dots, y_{1,r_1}, \dots, y_{s,1}, \dots, y_{s,r_s}$ be the variables of $\mathcal{P}$ and assume the lexicographic order $\succ_{\mathsf{lex}}$ with

$$\begin{aligned} x_{1,1} \succ_{\mathsf{lex}} \dots \succ_{\mathsf{lex}} x_{1,k_1} \succ_{\mathsf{lex}} \dots \succ_{\mathsf{lex}} x_{s,1} \succ_{\mathsf{lex}} \dots \succ_{\mathsf{lex}} x_{s,k_s} \\ \succ_{\mathsf{lex}} y_{1,1} \succ_{\mathsf{lex}} \dots \succ_{\mathsf{lex}} y_{1,r_1} \succ_{\mathsf{lex}} y_{2,1} \succ_{\mathsf{lex}} \dots \succ_{\mathsf{lex}} y_{2,r_2} \succ_{\mathsf{lex}} \dots \succ_{\mathsf{lex}} y_{s,r_s}. \end{aligned} \tag{2}$$

Since the systems $\mathcal{L}_i$ of linear equations do not share any variables we construct a Gröbner Basis for each of them independently. Then we show that the union of all these Gröbner Bases is indeed a Gröbner Basis for $\mathcal{I}(\mathcal{P})$. For each $\mathcal{L}_i$ we denote the corresponding ideal by $\mathcal{I}(\mathcal{L}_i)$.

Each linear system $\mathcal{L}_i$ is already in its reduced row-echelon form with $x_{i,j}$ as the leading monomial of the j-th equation, $1 \leq j \leq k_i$. Each linear equation can be written as $x_{i,j} + f_{i,j} = 0 \pmod{p_i^{m_i}}$ where $f_{i,j}$ is a linear polynomial over $\mathbb{Z}_{p_i^{m_i}}$. Hence, a generating set for $\mathcal{I}(\mathcal{L}_i)$ in an implicit form is as follows where the addition is modulo $\mathbb{Z}_{p_i^{m_i}}$,

$$G_i = \left\{x_{i,1} + f_{i,1}, \dots, x_{i,k_i} + f_{i,k_i}, \prod_{j \in \mathbb{Z}_{p_i^{m_i}}} (y_{i,1} - j), \dots, \prod_{j \in \mathbb{Z}_{p_i^{m_i}}} (y_{i,r_i} - j)\right\} \tag{3}$$

Unfortunately, a polynomial representation of $x_{i,j} + f_{i,j}$ is exponentially large, and so we need an extra step.

Let $U_{p_i^{m_i}} = \{\omega_i, \omega_i^2, \ldots, \omega_i^{(p_i^{m_i})} = \omega_i^0 = 1\}$ be the set of $p_i^{m_i}$-th roots of unity where ω_i is a primitive $p_i^{m_i}$-th root of unity. For a primitive $p_i^{m_i}$-th root of unity ω_i we have $\omega_i^a = \omega_i^b$ if and only if $a \equiv b \pmod{p_i^{m_i}}$. From $\mathcal{L}_i$ we construct a new CSP instance $\mathcal{L}'_i = (V, U_{p_i^{m_i}}, \widetilde{C})$ as follows. For each equation $x_{i,t} + f_{i,t} = 0 \pmod{p_i^{m_i}}$, where

$$f_{i,t} = \alpha_{i,t,1} y_{i,1} + \cdots + \alpha_{i,t,r_i} y_{i,r_i} + \alpha_{i,t},$$

we add the constraint $x_{i,t} - f'_{i,t} = 0$ (here subtraction is in $\mathbb{C}$) with

$$f'_{i,t} = \omega_i^{\alpha_{i,t}} \cdot \left(y_{i,1}^{\alpha_{i,t,1}} \cdot \ldots \cdot y_{i,r_i}^{\alpha_{i,t,r_i}} \right).$$

Moreover, the domain constraints are different. For each variable $x_{i,j}$, $j \in [k_i]$, or $y_{i,j}$, $j \in [r_i]$ the domain polynomial is $(x_{i,j})^{(p_i^{m_i})} - 1$, $(y_{i,j})^{(p_i^{m_i})} - 1$. However, we do not need the domain polynomials for variables $x_{i,j}$.

▶ **Lemma 18.** *The set of polynomials $G' = \cup_{1 \le i \le s} G'_i$, where*

$$G'_i = \left\{ x_{i,1} - f'_{i,1}, \ldots, x_{i,k_i} - f'_{i,k_i}, (y_{i,1})^{(p_i^{m_i})} - 1, \ldots, (y_{i,r_i})^{(p_i^{m_i})} - 1 \right\}$$

forms a Gröbner Basis for $\mathtt{I}(\mathcal{P}') = \mathbf{I}(\mathsf{Sol}(\mathcal{P}'))$ with respect to the lex *order* (2).

Step 5: Transforming the input polynomial. Given an instance $(f_0, \mathcal{P})$ of $\mathrm{IMP}_d(\Delta)$ Steps 1–4 transform $\mathcal{P}$ to an ideal over the set of roots of unity, for which a Gröbner Basis can be efficiently constructed. To complete a solution algorithm for $\mathrm{IMP}_d(\Delta)$ we need to demonstrate how to convert the input polynomial f_0.

To this end note that the reduction in Step 1 converted f_0 into a polynomial f'_0 over $x_{1,1}, \ldots, x_{1,k_1}, \ldots, x_{s,1}, \ldots, x_{s,k_s}$, see Theorem 8 and Lemma 13. Then for each $i \in [s]$ we define a univariate polynomial $\phi_i \in \mathbb{C}[X]$ that interpolates points $(\omega_i^0, 0), (\omega_i, 1), \ldots, (\omega_i^{(p_i^{m_i}-1)}, p_i^{m_i} - 1)$, that is, $\phi_i(a) = \omega_i^a$ for $a \in \mathbb{Z}_{p_i^{m_i}}$.

▶ **Lemma 19.** *Define polynomial $f''_0 \in \mathbb{C}[X]$ to be*

$$f''_0(x_{1,1}, \ldots, x_{1,k_1}, \ldots, x_{s,1}, \ldots, x_{s,k_s}) = f'_0\left(\phi_1^{-1}(x_{1,1}), \ldots, \phi_1^{-1}(x_{1,k_1}), \ldots, \phi_s^{-1}(x_{s,1}), \ldots, \phi_s^{-1}(x_{s,k_s})\right).$$

Then $f_0 \in \mathtt{I}(\mathcal{P})$ if and only if $f''_0 \in \mathtt{I}(\mathcal{P}')$.

If f_0 has degree at most d, the polynomial f''_0 has degree $O(d)$, and thus can be constructed in polynomial time. Therefore, Lemma 19 completes the proof of the first part of Theorem 11. The search version of $\mathrm{IMP}_d(\Delta)$ is discussed in the next section.

5 Search version and the substitution technique

In [15] we introduced a framework to bridge the gap between the decision and the search versions of the IMP. Indeed, this framework gives a polynomial time algorithm to construct a truncated Gröbner Basis provided that the search version of a variation of the IMP is polynomial time solvable. This variation is called χIMP and is defined as follows.

▶ **Definition 20** (χIMP). *Given an ideal* $\mathtt{I} \subseteq \mathbb{F}[x_1, \ldots, x_n]$ *and a vector of* ℓ *polynomials* $M = (g_1, \ldots, g_\ell)$*, the* χ*IMP asks if there exist coefficients* $\mathbf{c} = (c_1, \ldots, c_\ell) \in \mathbb{F}^\ell$ *such that* $\mathbf{c}M = \sum_{i=1}^{\ell} c_i g_i$ *belongs to the ideal* $\mathtt{I}$*. In the search version of the problem the goal is to find coefficients* $\mathbf{c}$*.*

The χIMP associated with a (multi-sorted) constraint language Γ over a set $\mathcal{D}$ is the problem $\chi\mathrm{IMP}(\Gamma)$ in which the input is a pair $(M, \mathcal{P})$ where $\mathcal{P}$ is a $\mathrm{CSP}(\Gamma)$ instance and M is a vector of ℓ polynomials. The goal is to decide whether there are coefficients $\mathbf{c} = (c_1, \ldots, c_\ell) \in \mathbb{F}^\ell$ such that $\mathbf{c}M$ lies in the combinatorial ideal $\mathcal{I}(\mathcal{P})$. We use $\chi\mathrm{IMP}_d(\Gamma)$ to denote $\chi\mathrm{IMP}(\Gamma)$ when the vector M contains polynomials of degree at most d.

▶ **Theorem 21** (Theorem 1 part (2) paraphrased). *Let* $\mathcal{H}$ *be a class of ideals for which the search version of* χIMP_d *is polynomial time solvable. Then there exists a polynomial time algorithm that constructs a d-truncated Gröbner Basis of an ideal* $\mathtt{I} \in \mathcal{H}$*,* $\mathtt{I} \subseteq \mathbb{F}[x_1, \ldots, x_n]$*, in time* $O(n^d)$*.*

The above theorem suggests that, in order to prove the second part of Theorem 11, it is sufficient to show that χIMP is polynomial time solvable for instances of CSP arising from constraint languages that are closed under the affine operation of an Abelian group.

It was shown in [15] that having a Gröbner Basis yields a polynomial time algorithm for solving the search version of χIMP (by using the *division algorithm* and solving a system of linear equations).

▶ **Theorem 22** ([15]). *Let* $\mathtt{I}$ *be an ideal, and let* $\{g_1, \ldots, g_s\}$ *be a Gröbner Basis for* $\mathtt{I}$ *with respect to some monomial ordering. Then the (search version of)* χIMP *is polynomial time solvable.*

Given the above theorem, to solve the χIMP one might reduce the problem at hand to a problem for which a Gröbner Basis can be constructed in a relatively simple way. This approach has been proven to be extremely useful in various cases studied in [15]. In that paper the reductions for χIMP are proved in an ad hoc manner. However, the core idea in all of them is a substitution technique. Here we provide a unifying construction based on *substitution reductions* that covers all the useful cases so far.

5.1 Reduction by substitution

We call a class of χIMPs *CSP-based* if its instances are of the form $(M, \mathcal{P})$, where $\mathcal{P}$ is a CSP instance over a fixed set D. Let $\mathcal{X}, \mathcal{Y}$ be restricted CSP-based classes of the χIMP. The classes $\mathcal{X}, \mathcal{Y}$ can be defined by various kinds of restrictions, for example, as $\chi\mathrm{IMP}(\Gamma), \chi\mathrm{IMP}(\Delta)$, but not necessarily. Let the domain of $\mathcal{X}$ be D and the domain of $\mathcal{Y}$ be E. Let also $\mu_1, \ldots, \mu_k$ be a collection of surjective functions $\mu_i : E^{\ell_i} \to D$, $i \in [k]$. Each mapping μ_i can be interpolated by a polynomial h_i. We call the collection $\{h_1, \ldots, h_k\}$ a *substitution collection*.

The problem $\mathcal{X}$ is said to be *substitution reducible* to $\mathcal{Y}$ if there exists a substitution collection $\{h_1, \ldots, h_k\}$ and a polynomial time algorithm A such that for every instance $(M, \mathcal{P})$ of $\mathcal{X}$ an instance constructed as follows belongs to $\mathcal{Y}$.

(1) Let X be the set of variables of $(M, \mathcal{P})$. For every $x \in X$ the algorithm A selects a polynomial h_{i_x} and a set of variables Y_x such that
  - **(a)** $|Y_x| = \ell_{i_x}$;
  - **(b)** for any $x, y \in X$ either $Y_x = Y_y$ or $Y_x \cap Y_y = \emptyset$;
  - **(c)** if $x_1, \ldots, x_r \in X$ are such that $Y_{x_1} = \cdots = Y_{x_r} = \{y_1, \ldots, y_{\ell_j}\}$ then for any solution φ of $\mathcal{P}$ there are values $a_1, \ldots, a_{\ell_j} \in E$ such that $\varphi(x_i) = h_{i_{x_i}}(a_1, \ldots, a_{\ell_j})$.

(2) If $M = (g_1, \dots, g_\ell)$ then $M' = (g'_1, \dots, g'_\ell)$, where for $g_i(x_1, \dots, x_t)$

$$g'_i = g_i(h_{i_{x_1}}(Y_{x_1}), \dots, h_{i_{x_y}}(Y_{x_t})).$$

(3) Let $Y = \bigcup_{x \in X} Y_x$. The instance $\mathcal{P}'$ is given by $(Y, E, \mathcal{C}')$, where for every constraint $\langle \mathbf{s}, R \rangle$, $\mathbf{s} = (x_1, \dots, x_t)$, $\mathcal{P}'$ contains the constraint $\langle \mathbf{s}', R' \rangle$ such that
- $\mathbf{s}' = (x_{1,1}, \dots, x_{1,\ell_{x_1}}, x_{2,1}, \dots, x_{t,\ell_{x_t}})$, where $Y_{x_j} = \{x_{j,1}, \dots, x_{j,\ell_j}\}$;
- R' is an ℓ-ary relation, $\ell = \ell_{x_1} + , \dots, + \ell_{x_t}$, such that $(a_{1,1}, \dots, a_{1,\ell_{x_1}}, a_{2,1}, \dots, a_{t,\ell_{x_t}}) \in R'$ if and only if $(h_{i_{x_1}}(a_{1,1}, \dots, a_{1,\ell_{x_1}}), \dots, h_{i_{x_t}}(a_{t,1}, \dots, a_{t,\ell_{x_t}})) \in R$.

▶ **Lemma 23.** *Let $\mathcal{X}, \mathcal{Y}$ be restricted CSP-based classes of the χIMP_d and χIMP_{rd}, respectively, $r \geq 1$. If $\mathcal{X}$ is substitution reducible to $\mathcal{Y}$ with a substitution collection $\{h_1, \dots, h_k\}$, and $r \geq \ell_i$ for each $i \in [k]$, then there is a polynomial time reduction from $\mathcal{X}$ to $\mathcal{Y}$.*

Since the search χIMP can be solved whenever a Gröbner Basis can be efficiently found, the above lemma provide a powerful tool for solving the χIMP. That is, if $\mathcal{X}$ is substitution reducible to $\mathcal{Y}$ and furthermore $\mathcal{Y}$ is such that it admits a polynomial time algorithm to construct a Gröbner Basis, then instances of $\mathcal{X}$ are solvable in polynomial time too.

▶ **Theorem 24.** *Let $\mathcal{X}, \mathcal{Y}$ be restricted CSP-based classes of the χIMP_d and χIMP_{rd}, $r \geq 1$ respectively, such that $\mathcal{X}$ is substitution reducible to $\mathcal{Y}$ with a substitution collection $\{h_1, \dots, h_k\}$ and $r \geq \ell_i$ for $i \in [k]$. Suppose there exists a polynomial time algorithm that for any instance $(M', \mathcal{P}')$ of $\mathcal{Y}$ constructs a (truncated) Gröbner Basis, then*
1. *there is a polynomial time algorithm that solves every instance $(M, \mathcal{P})$ of $\mathcal{X}$; and*
2. *there is a polynomial time algorithm that for any instance $(M, \mathcal{P})$ of $\mathcal{X}$ constructs a d-truncated Gröbner Basis for $\mathtt{I}(\mathcal{P})$.*

We point out that the second part of Theorem 24 follows from Theorem 21, that is, since every instance $(M, \mathcal{P})$ of $\mathcal{X}$ is polynomial time solvable, by Theorem 21, we can construct a d-truncated Gröbner Basis for $\mathcal{I}(\mathcal{P})$ in polynomial time.

If $\mathcal{X}, \mathcal{Y}$ are of the form $\chi\text{IMP}(\Gamma)$, Theorem 24 implies the following corollary, which covers virtually all the reductions suggested in [15].

▶ **Corollary 25.** *Let Δ and Γ be multi-sorted constraint languages over finite collection of sets $\mathcal{D} = \{D_t \mid t \in T\}$, $\mathcal{E} = \{E_s \mid s \in S\}$, respectively. Suppose Γ pp-interprets Δ and there exists a polynomial time algorithm that for any instance $(M', \mathcal{P}')$ of $\chi IMP_{O(d)}(\Gamma)$ constructs a (truncated) Gröbner Basis, then*
1. *there is a polynomial time algorithm that solves every instance $(M, \mathcal{P})$ of $\chi IMP_d(\Delta)$; and*
2. *there is a polynomial time algorithm that for any instance $(M, \mathcal{P})$ of $\chi IMP_d(\Delta)$ constructs a d-truncated Gröbner Basis for $\mathtt{I}(\mathcal{P})$.*

Given Corollary 25, we can prove the reductions in Steps 1–5 are reductions by substitution (see the full version [16]), thus by Theorem 24 we can construct a d-truncated Gröbner Basis which yields the search version of Theorem 11.

References

1 Albert Atserias and Joanna Ochremiak. Proof complexity meets algebra. *ACM Trans. Comput. Log.*, 20(1):1:1–1:46, 2019.

2 Libor Barto and Marcin Kozik. Constraint satisfaction problems solvable by local consistency methods. *J. ACM*, 61(1):3:1–3:19, 2014.

3 Libor Barto, Andrei A. Krokhin, and Ross Willard. Polymorphisms, and how to use them. In Andrei A. Krokhin and Stanislav Zivný, editors, *The Constraint Satisfaction Problem: Complexity and Approximability*, volume 7 of *Dagstuhl Follow-Ups*, pages 1–44. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2017.

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, FOCS 1994, Santa Fe, New Mexico, USA, 20-22 November 1994*, pages 794–806. IEEE Computer Society, 1994. doi:10.1109/SFCS.1994.365714.

5 Arpitha P. Bharathi and Monaldo Mastrolilli. Ideal membership problem and a majority polymorphism over the ternary domain. In Javier Esparza and Daniel Král', editors, *45th International Symposium on Mathematical Foundations of Computer Science, MFCS 2020, August 24-28, 2020, Prague, Czech Republic*, volume 170 of *LIPIcs*, pages 13:1–13:13. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2020. doi:10.4230/LIPIcs.MFCS.2020.13.

6 Arpitha P. Bharathi and Monaldo Mastrolilli. Ideal membership problem for boolean minority. *CoRR*, abs/2006.16422, 2020. arXiv:2006.16422.

7 Arpitha P. Bharathi and Monaldo Mastrolilli. Ideal membership problem for boolean minority and dual discriminator. In *46th International Symposium on Mathematical Foundations of Computer Science, MFCS 2021, August 23-27, 2021, Tallinn, Estonia*. To appear, 2021.

8 V.G. Bodnarchuk, L.A. Kaluzhnin, V.N. Kotov, and B.A. Romov. Galois theory for post algebras. i. *Kibernetika*, 3:1–10, 1969.

9 Bruno Buchberger. Bruno buchberger's phd thesis 1965: An algorithm for finding the basis elements of the residue class ring of a zero dimensional polynomial ideal. *Journal of Symbolic Computation*, 41(3):475–511, 2006. Logic, Mathematics and Computer Science: Interactions in honor of Bruno Buchberger (60th birthday). doi:10.1016/j.jsc.2005.09.007.

10 Andrei A. Bulatov. A dichotomy theorem for nonuniform csps. In Chris Umans, editor, *58th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2017, Berkeley, CA, USA, October 15-17, 2017*, pages 319–330. IEEE Computer Society, 2017.

11 Andrei A. Bulatov and Víctor Dalmau. A simple algorithm for mal'tsev constraints. *SIAM J. Comput.*, 36(1):16–27, 2006.

12 Andrei A. Bulatov and Peter Jeavons. An algebraic approach to multi-sorted constraints. In *Principles and Practice of Constraint Programming - CP 2003, 9th International Conference, CP 2003, Kinsale, Ireland, September 29 - October 3, 2003, Proceedings*, volume 2833 of *Lecture Notes in Computer Science*, pages 183–198. Springer, 2003. doi:10.1007/978-3-540-45193-8_13.

13 Andrei A. Bulatov, Peter Jeavons, and Andrei A. Krokhin. Classifying the complexity of constraints using finite algebras. *SIAM J. Comput.*, 34(3):720–742, 2005. doi:10.1137/S0097539700376676.

14 Andrei A. Bulatov, Andrei A. Krokhin, and Benoît Larose. Dualities for constraint satisfaction problems. In Nadia Creignou, Phokion G. Kolaitis, and Heribert Vollmer, editors, *Complexity of Constraints - An Overview of Current Research Themes [Result of a Dagstuhl Seminar]*, volume 5250 of *Lecture Notes in Computer Science*, pages 93–124. Springer, 2008.

15 Andrei A. Bulatov and Akbar Rafiey. On the complexity of csp-based ideal membership problems. *CoRR*, abs/2011.03700, 2020.

16 Andrei A. Bulatov and Akbar Rafiey. The ideal membership problem and abelian groups. *CoRR*, abs/2201.05218, 2022.

17 Samuel R. Buss and Toniann Pitassi. Good degree bounds on nullstellensatz refutations of the induction principle. In Steven Homer and Jin-Yi Cai, editors, *Proceedings of the 11th Annual IEEE Conference on Computational Complexity, CCC 1996, Philadelphia, Pennsylvania, USA, May 24-27, 1996*, pages 233–242. IEEE Computer Society, 1996. doi:10.1109/CCC.1996.507685.

18 Matthew Clegg, Jeff Edmonds, and Russell Impagliazzo. Using the groebner basis algorithm to find proofs of unsatisfiability. In Gary L. Miller, editor, *Proceedings of the 28th Annual ACM Symposium on the Theory of Computing, STOC 1996, Philadelphia, Pennsylvania, USA, May 22-24, 1996*, pages 174–183. ACM, 1996. doi:10.1145/237814.237860.

19 David Cox, John Little, and Donal OShea. *Ideals, varieties, and algorithms: an introduction to computational algebraic geometry and commutative algebra*. Springer Science & Business Media, 2013.

20 Alicia Dickenstein, Noaï Fitchas, Marc Giusti, and Carmen Sessa. The membership problem for unmixed polynomial ideals is solvable in single exponential time. *Discret. Appl. Math.*, 33(1-3):73–94, 1991. doi:10.1016/0166-218X(91)90109-A.

21 D. Geiger. Closed systems of function and predicates. *Pacific Journal of Mathematics*, pages 95–100, 1968.

22 Dima Grigoriev. Tseitin's tautologies and lower bounds for nullstellensatz proofs. In *39th Annual Symposium on Foundations of Computer Science, FOCS 1998, November 8-11, 1998, Palo Alto, California, USA*, pages 648–652. IEEE Computer Society, 1998. doi:10.1109/SFCS.1998.743515.

23 Grete Hermann. Die frage der endlich vielen schritte in der theorie der polynomideale. *Mathematische Annalen*, 95(1):736–788, 1926.

24 Pawel M. Idziak, Petar Markovic, Ralph McKenzie, Matthew Valeriote, and Ross Willard. Tractability and learnability arising from algebras with few subpowers. *SIAM J. Comput.*, 39(7):3023–3037, 2010.

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

26 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, March 2013. doi:10.1007/s10472-013-9365-7.

27 Monaldo Mastrolilli. The complexity of the ideal membership problem for constrained problems over the boolean domain. In *Proceedings of the 13th Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2019, San Diego, California, USA, January 6-9, 2019*, pages 456–475, 2019. doi:10.1137/1.9781611975482.29.

28 Ernst W. Mayr. Membership in plynomial ideals over Q is exponential space complete. In Burkhard Monien and Robert Cori, editors, *6th Annual Symposium on Theoretical Aspects of Computer Science, STACS 1989, Paderborn, FRG, February 16-18, 1989, Proceedings*, volume 349 of *Lecture Notes in Computer Science*, pages 400–406. Springer, 1989. doi:10.1007/BFb0029002.

29 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.

30 Ryan O'Donnell. SOS is not obviously automatizable, even approximately. In Christos H. Papadimitriou, editor, *8th Innovations in Theoretical Computer Science Conference, ITCS 2017, January 9-11, 2017, Berkeley, CA, USA*, volume 67 of *LIPIcs*, pages 59:1–59:10. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2017. doi:10.4230/LIPIcs.ITCS.2017.59.

31 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, July 10-14, 2017, Warsaw, Poland*, volume 80 of *LIPIcs*, pages 80:1–80:13. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2017. doi:10.4230/LIPIcs.ICALP.2017.80.

32 Fred Richman. Constructive aspects of noetherian rings. *Proceedings of the American Mathematical Society*, 44(2):436–441, 1974.

33 Abraham Seidenberg. Constructions in algebra. *Transactions of the American Mathematical Society*, 197:273–313, 1974.

34 Johan Thapper and Stanislav Zivný. The limits of SDP relaxations for general-valued csps. *ACM Trans. Comput. Theory*, 10(3):12:1–12:22, 2018.

35 Dmitriy Zhuk. A proof of CSP dichotomy conjecture. In Chris Umans, editor, *58th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2017, Berkeley, CA, USA, October 15-17, 2017*, pages 331–342. IEEE Computer Society, 2017.