SIAM J. COMPUT.
Vol. 46, No. 3, pp. 1087–1110

THE COMPLEXITY OF GENERAL-VALUED CSPs*

VLADIMIR KOLMOGOROV†, ANDREI KROKHIN‡, AND MICHAL ROLÍNEK†

Abstract. An instance of the valued constraint satisfaction problem (VCSP) is given by a finite set of variables, a finite domain of labels, and a sum of functions, each function depending on a subset of the variables. Each function can take finite values specifying costs of assignments of labels to its variables or the infinite value, which indicates an infeasible assignment. The goal is to find an assignment of labels to the variables that minimizes the sum. We study, assuming that $\mathrm{P} \neq \mathrm{NP}$, how the complexity of this very general problem depends on the set of functions allowed in the instances, the so-called constraint language. The case when all allowed functions take values in $\{0, \infty\}$ corresponds to ordinary CSPs, where one deals only with the feasibility issue, and there is no optimization. This case is the subject of the algebraic CSP dichotomy conjecture predicting for which constraint languages CSPs are tractable (i.e., solvable in polynomial time) and for which they are NP-hard. The case when all allowed functions take only finite values corresponds to a finite-valued CSP, where the feasibility aspect is trivial and one deals only with the optimization issue. The complexity of finite-valued CSPs was fully classified by Thapper and Živný. An algebraic necessary condition for tractability of a general-valued CSP with a fixed constraint language was recently given by Kozik and Ochremiak. As our main result, we prove that if a constraint language satisfies this algebraic necessary condition, and the feasibility CSP (i.e., the problem of deciding whether a given instance has a feasible solution) corresponding to the VCSP with this language is tractable, then the VCSP is tractable. The algorithm is a simple combination of the assumed algorithm for the feasibility CSP and the standard LP relaxation. As a corollary, we obtain that a dichotomy for ordinary CSPs would imply a dichotomy for general-valued CSPs.

Key words. valued constraint satisfaction problem, complexity, dichotomy, fractional polymorphism

AMS subject classification. 68Q25

DOI. 10.1137/16M1091836

1. Introduction.

Computational problems from many different areas involve finding an assignment of labels to a set of variables, where that assignment must satisfy some specified feasibility conditions and/or optimize some specified objective function. In many such problems, the feasibility conditions are local and also the objective function can be represented as a sum of functions, each of which depends on some subset of the variables. Examples include: Gibbs energy minimization, Markov random fields (MRFs), conditional random fields (CRFs), min-sum problems, minimum cost homomorphism, constraint optimization problems (COPs), and valued constraint satisfaction problems (VCSPs) [7, 18, 40, 44, 53].

The constraint satisfaction problem provides a common framework for many theoretical and practical problems in computer science [19, 44]. An instance of the *constraint satisfaction problem* (CSP) consists of a collection of variables that must be assigned labels from a given domain subject to specified constraints [42]. The CSP is equivalent to the problem of evaluating conjunctive queries on databases [33], and

*Received by the editors August 31, 2016; accepted for publication (in revised form) February 8, 2017; published electronically June 29, 2017. An extended abstract of this work (without proofs) has appeared in FOCS 2015.

http://www.siam.org/journals/sicomp/46-3/M109183.html

Funding: The first and third authors are supported by the European Research Council under the European Union's Seventh Framework Programme (FP7/2007-2013)/ERC grant agreement 616160.

†IST Austria, Klosterneuberg, Austria (vnk@ist.ac.at, michal.rolinek@ist.ac.at).

‡School of Engineering, Durham University, Durham DH1 3LE, UK (andrei.krokhin@durham.ac.uk).

to the homomorphism problem for relational structures [23]. The CSP deals only with the feasibility issue: can all constraints be satisfied simultaneously?

There are several natural optimization versions of the CSP: Max CSP (or Min CSP) where the goal is to find the assignment maximizing the number of satisfied constraints (or minimizing the number of unsatisfied constraints) [15, 19, 30, 31], problems like Max-Ones and Min-Hom where the constraints must be satisfied and some additional function of the assignment is to be optimized [19, 32, 47], and, the most general version, *valued CSP* or VCSP (also known as soft CSP), where each combination of values for variables in a constraint has a cost and the goal is to minimize the aggregate cost [13, 17, 35, 49]. Thus, an instance of the VCSP amounts to minimizing a sum of functions, each depending on a subset of variables. By using infinite costs to indicate infeasible combinations, VCSP can model both feasibility and optimization aspects and so considerably generalizes all the problems mentioned above [13, 17, 39]. There is much activity and there are very strong results concerning various aspects of approximability of (V)CSPs (see, e.g., [5, 8, 12, 19, 21, 22, 26, 43] for a small sample), but in this paper we focus on solving VCSPs to optimality.

We assume throughout the paper that P $\neq$ NP. Since all the above problems are NP-hard in full generality, a major line of research in CSP tries to identify the tractable cases of such problems (see books/surveys [16, 19, 20, 39]), the primary motivation being the general picture rather than specific applications. The two main ingredients of a constraint are (a) variables to which it is applied and (b) relations/functions specifying the allowed combinations of values or the costs for all combinations. Therefore, the main types of restrictions on CSP are (a) *structural* where the hypergraph formed by sets of variables appearing in individual constraints is restricted [25, 41], and (b) *language-based* where the constraint language, i.e., the set of relations/functions that can appear in constraints, is fixed (see, e.g., [10, 16, 19, 23, 49]). The ultimate sort of results in these directions are *dichotomy* results, pioneered by [45], which characterize the tractable restrictions and show that the rest are as hard as the corresponding general problem (which cannot generally be taken for granted). The language-based direction is considerably more active than the structural one, there are many partial language-based dichotomy results, e.g., [9, 11, 17, 19, 30, 31, 36, 47], but many central questions are still open. In this paper, we study VCSPs with a fixed constraint language on a finite domain, and all further discussion concerns only such CSPs and VCSPs.

Related work. The CSP dichotomy conjecture, stating that each CSP is either tractable or NP-hard, was first formulated by Feder and Vardi [23]. The universal-algebraic approach to this problem was discovered in [10, 28, 29], and the precise boundary between the tractable cases and NP-hard cases was conjectured in algebraic terms in [10], in what is now known as the algebraic CSP dichotomy conjecture (see Conjecture 2.16). The hardness part was proved in [10], and it is the tractability part that is the essence of the conjecture. This conjecture is still open in full generality and is the object of much investigation, e.g., [2, 3, 4, 1, 6, 10, 11, 16, 27]. It is known to hold for domains with at most three elements [9, 45], for smooth digraphs [6], and for the case when all unary relations are available [1, 11]. The main two polynomial-time algorithms used for CSPs are one based on local consistency ("bounded width") and the other based on compact representation of solution sets ("few subpowers"), and their applicability (in pure form) is fully characterized in [2, 4] and [27], respectively.

At the opposite (to CSP) end of the VCSP spectrum are the finite-valued CSPs, in which functions do not take infinite values. In such VCSPs, the feasibility aspect is trivial, and one has to deal only with the optimization issue. One polynomial-time algorithm that solves tractable finite-valued CSPs is based on the so-called basic linear programming (BLP) relaxation, and its applicability (also for the general-valued case) was fully characterized in [35] (see Theorem 2.17). The complexity of finite-valued CSPs was completely classified in [49], where it is shown that all finite-valued CSPs not solvable by BLP are NP-hard.

For general-valued CSPs, full classifications are known for the Boolean case (i.e., when the domain is two-element) [17] and also for the case when all 0-1-valued unary cost functions are available [36]. The algebraic approach to the CSP was extended to VCSPs in [13, 14, 17, 37], and was also key to much progress. An algebraic necessary condition for a VCSP to be tractable was recently proved by Kozik and Ochremiak in [37], where this condition was also conjectured to be sufficient (see Theorem 2.14 and Conjecture 2.15 below). This conjecture can be called the algebraic VCSP dichotomy conjecture, and it is a generalization of the corresponding conjecture for CSP. A large family of VCSPs satisfying the necessary condition from [37] has recently been shown tractable via a low-level Sherali–Adams hierarchy relaxation [48].

Our proof uses the technique of "lifting a language" introduced in [34].

Our contribution. We completely classify the complexity of VCSPs with a fixed constraint language modulo the complexity of CSPs (see Theorem 3.3). Clearly, for a VCSP to be tractable, it is necessary that the corresponding feasibility CSP is tractable. We prove that any VCSP satisfying this necessary condition and the necessary condition of Kozik and Ochremiak is tractable. The polynomial-time algorithm that solves such VCSP is a simple combination of the (assumed) polynomial-time algorithm for the feasibility CSP and BLP (see Theorem 3.4). Thus, our dichotomy theorem generalizes the dichotomy for finite-valued CSPs from [49], and, with the help of the CSP tractability result from [4], it also implies the tractability of VCSPs shown tractable in [48, 50].

Our classification result has the following several unexpected features. One is that the algorithm that solves all tractable VCSPs uses feasibility checking only as a black box. The other is that the algorithm is simply feasibility preprocessing followed by BLP — this was unexpected, for example, because higher levels of the Sherali-Adams hierarchy were used in [48] to prove tractability of a wide class of VCSPs. Finally, the proof of our result avoids structural universal algebra present in most CSP classifications and in [37, 38].

Our result says that any dichotomy for CSP (not necessarily the one predicted by the algebraic CSP dichotomy conjecture) will imply a dichotomy for VCSP. However, if the algebraic CSP dichotomy conjecture holds, then the necessary algebraic condition of Kozik and Ochremiak guarantees tractability of the feasibility CSP (see [37]), implying that this algebraic condition alone is necessary and sufficient for tractability of a VCSP, and also that all the intractable VCSPs are NP-hard. In particular, the algebraic CSP dichotomy conjecture implies the algebraic VCSP dichotomy conjecture.

On the technical level, some of our proofs (e.g., those in section 7) use techniques established in [35, 49], while others (e.g., all of section 6) introduce new technical ideas.

Our result is the culmination of research into complexity classification of language-based VCSPs in the sense that its scope cannot be widened, the yet unclassified part of the VCSP landscape is the (nonvalued) CSP. One could, of course, extend the classification framework by looking at other forms of algorithmic tractability, say, approximation algorithms or fixed-parameter tractability, and such extensions will have many open questions. It is also interesting to obtain tighter and more explicit characterizations for important special cases of VCSP (as done in [50], for example), by deriving them from our main result or otherwise.

2. Preliminaries.

2.1. Valued constraint satisfaction problems. Throughout the paper, let D be a fixed finite set and let $\overline{\mathbb{Q}} = \mathbb{Q} \cup \{\infty\}$ denote the set of rational numbers with (positive) infinity.

DEFINITION 2.1. *We denote the set of all functions $f : D^n \to \overline{\mathbb{Q}}$ by $\mathcal{F}_D^{(n)}$ and let $\mathcal{F}_D = \bigcup_{n\geq 1} \mathcal{F}_D^{(n)}$. We will often call the functions in $\mathcal{F}_D$* cost functions *over D. For every cost function $f \in \mathcal{F}_D^{(n)}$, let* $\operatorname{dom} f = \{x \mid f(x) < \infty\}$. *Note that* $\operatorname{dom} f$ *can be considered both as an n-ary relation and as a n-ary function such that* $\operatorname{dom} f(x) = 0$ *if and only if $f(x)$ is finite.*

We will call the set D the *domain*, elements of D *labels* (for variables), and say that the cost functions in $\mathcal{F}_D$ take *values*. Note that in some papers on VCSP, e.g., [13, 48], cost functions are called weighted relations.

DEFINITION 2.2. *An instance of the* valued constraint satisfaction problem *(VCSP) is a function from D^V to $\overline{\mathbb{Q}}$ given by*

$$f_{\mathcal{I}}(x) = \sum_{t\in T} f_t\left(x_{v(t,1)}, \ldots, x_{v(t,n_t)}\right), \tag{1}$$

where V is a finite set of variables, T is a finite set of constraints, and each constraint is specified by a cost function f_t of arity n_t and indices $v(t,k)$, $k = 1, \ldots, n_t$. The goal is to find an assignment *(or* labeling*) $x \in D^V$ that minimizes $f_{\mathcal{I}}$. The value of an optimal assignment is denoted by* $\operatorname{Opt}(\mathcal{I})$.

DEFINITION 2.3. *Any set $\Gamma \subseteq \mathcal{F}_D$ is called a* valued constraint language *over D, or simply a* language. *We will denote by* VCSP(Γ) *the class of all VCSP instances in which the constraint functions f_t are all contained in Γ. Instances of* VCSP(Γ) *will sometimes be called just* Γ-instances.

This framework subsumes many other frameworks studied earlier and captures many specific well-known problems, including k-SAT, GRAPH k-COLOURING, MAX CUT, MIN VERTEX COVER, and others (see [39]). Note that if every function in Γ takes values in $\{0, \infty\}$ (such functions are often called *crisp*) then VCSP(Γ) is a pure feasibility problem, commonly known as CSP(Γ).

The main goal of our line of research is to classify the complexity of problems VCSP(Γ). Problems CSP(Γ) and VCSP(Γ) are called tractable if, for each finite $\Gamma' \subseteq \Gamma$, VCSP(Γ') is tractable. Also, VCSP(Γ) is called NP-hard if, for some finite $\Gamma' \subseteq \Gamma$, VCSP(Γ') is NP-hard. One advantage of defining tractability in terms of finite subsets is that the tractability of a valued constraint language is independent of whether the cost functions are represented explicitly (say, via full tables of values, or via tables for the finite-valued parts) or implicitly (via oracles). Following [10], we say that VCSP(Γ) is *globally tractable* there is a polynomial-time algorithm solving VCSP(Γ), assuming all functions in instances are given by full tables of values. For CSPs, there is no example of CSP(Γ) that is tractable, but not globally tractable, and it is conjectured in [10] that no such CSP(Γ) exists.

2.2. Polymorphisms, expressibility, cores. Let $\mathcal{O}_D^{(m)}$ denote the set of all operations $g : D^m \to D$ and let $\mathcal{O}_D = \cup_{m\geq 1}\mathcal{O}_D^{(m)}$. When D is clear from the context, we will sometimes write simply $\mathcal{O}^{(m)}$ and $\mathcal{O}$.

Any language Γ defined on D can be associated with a set of operations on D, known as the polymorphisms of Γ, which allow one to combine (often in a useful way) several feasible assignments into a new one.

DEFINITION 2.4. *An operation $g \in \mathcal{O}_D^{(m)}$ is a* polymorphism *of a cost function $f \in \mathcal{F}_D$ if, for any $x^1, x^2, \ldots, x^m \in \operatorname{dom} f$, we have that $g(x^1, x^2, \ldots, x^m) \in \operatorname{dom} f$ where g is applied component-wise.*

For any valued constraint language Γ over a set D, we denote by $\mathrm{Pol}(\Gamma)$ *the set of all operations on D which are polymorphisms of every $f \in \Gamma$.*

Example 2.5. Let $f \in \mathcal{F}_{\{0,1\}}^{(n)}$ be such that $f(1, \ldots, 1, 0) = \infty$ and $f(a_1, \ldots, a_n) = 0$ otherwise. It corresponds to the Horn clause $(x_1 \vee \ldots \vee x_{n-1} \vee \overline{x_n})$. Then it is well known and easy to see that the binary operation $\min \in \mathcal{O}_{\{0,1\}}$ is a polymorphism of f.

Clearly, if g is a polymorphism of a cost function f, then g is also a polymorphism of $\operatorname{dom} f$. For $\{0, \infty\}$-valued functions, which naturally correspond to relations, the notion of a polymorphism defined above coincides with the standard notion of a polymorphism for relations. Note that the projections (aka dictators), i.e., operations of the form $e_n^i(x_1, \ldots, x_n) = x_i$, are polymorphisms of all valued constraint languages. Polymorphisms play the key role in the algebraic approach to the CSP, but, for VCSPs, more general constructs are necessary, which we now define.

DEFINITION 2.6. *An m-ary* fractional operation *ω on D is a probability distribution on $\mathcal{O}_D^{(m)}$. The support of ω is defined as* $\mathrm{supp}(\omega) = \{g \in \mathcal{O}_D^{(m)} \mid \omega(g) > 0\}$.

DEFINITION 2.7. *An m-ary fractional operation ω on D is said to be a* fractional polymorphism *of a cost function $f \in \mathcal{F}_D$ if, for any $x^1, x^2, \ldots, x^m \in \operatorname{dom} f$, we have*

$$\sum_{g \in \mathrm{supp}(\omega)} \omega(g) f\left(g(x^1, \ldots, x^m)\right) \leq \frac{1}{m}\left(f(x^1) + \cdots + f(x^m)\right). \tag{2}$$

For a constraint language Γ, $\mathrm{fPol}(\Gamma)$ *will denote the set of all fractional operations that are fractional polymorphisms of each function in Γ. Also, let* $\mathrm{fPol}^+(\Gamma) = \{g \in \mathcal{O}_D \mid g \in \mathrm{supp}(\omega), \omega \in \mathrm{fPol}(\Gamma)\}$.

The intuition behind the notion of fractional polymorphism is that it allows one to combine several feasible assignments into new feasible assignments so that the expected value of a new assignment (nonstrictly) improves the average value of the original assignments.

Example 2.8. Suppose that ω is a binary fractional operation on $D = \{0, 1\}$ such that $\omega(\min) = \omega(\max) = 1/2$. Then it is well known and easy to check that the finite-valued functions with fractional polymorphism ω are the submodular functions. Moreover, functions with this fractional polymorphism that are not necessarily finite-valued precisely correspond to submodular functions defined on a ring family.

More examples of fractional polymorphisms can be found in [39, 35, 49].

We remark that, in some papers (e.g., in [13]), fractional polymorphisms (and closely related objects called weighted polymorphisms) are defined as rational-valued functions, which is sufficient for analyzing the complexity of VCSPs with finite constraint languages. However, real-valued fractional polymorphisms are necessary to analyze infinite constraint languages [24, 38, 49].

The key observation in the algebraic approach to (V)CSP is that neither the complexity nor the algebraic properties of a language Γ change when functions "expressible" from Γ in a certain way are added to it.

DEFINITION 2.9. *For a constraint language* Γ, *let* $\langle\Gamma\rangle$ *denote the set of all functions* $f(x_1,\dots,x_k)$ *such that, for some instance* $\mathcal{I}$ *of* VCSP(Γ) *with objective function* $f_{\mathcal{I}}(x_1,\dots,x_k,x_{k+1},\dots,x_n)$, *we have*

$$f(x_1,\dots,x_k)=\min_{x_{k+1},\dots,x_n} f_{\mathcal{I}}(x_1,\dots,x_k,x_{k+1},\dots,x_n).$$

We then say that Γ expresses f, *and call* $\langle\Gamma\rangle$ *the* expressive power *of* Γ.

LEMMA 2.10. [14, 17] *Let* $f\in\langle\Gamma\rangle$. *Then*

1. *if* $\omega\in\mathrm{fPol}(\Gamma)$ *then* ω *is a fractional polymorphism of* f *and of* $\mathrm{dom}\, f$;
2. VCSP(Γ) *is tractable if and only if* VCSP($\Gamma\cup\{f,\mathrm{dom}\, f\}$) *is tractable;*
3. VCSP(Γ) *is NP-hard if and only if* VCSP($\Gamma\cup\{f,\mathrm{dom}\, f\}$) *is NP-hard.*

The dichotomy problem for VCSPs can be reduced to a class of constraint languages called rigid cores, defined below. Apart from reducing the cases that need to be considered, this reduction enabled the use of much more powerful results from universal algebra than what can be done without this restriction (see, e.g., [38]).

For a subset $D'\subseteq D$, let $u_{D'}$ be the function defined as follows: $u_{D'}(d)=0$ if $d\in D'$ and $u_{D'}(d)=\infty$ otherwise. We write u_d for $u_{\{d\}}$. Let $\mathcal{C}_D=\{\{u_d\}\mid d\in D\}$.

LEMMA 2.11. [38] *For any valued constraint language* Γ' *on a finite set* D', *there is a subset* $D\subseteq D'$ *and a valued constraint language* Γ *on* D *such that* $\mathcal{C}_D\subseteq\Gamma$ *and the problems* VCSP(Γ') *and* VCSP(Γ) *are polynomial-time equivalent.*

This language Γ is called the *rigid core* of Γ', and it can be obtained from Γ' as follows. Let g' be a unary operation on D' with minimum $|g'(D')|$ among all unary operations $g'\in\mathrm{fPol}^+(\Gamma')$. Then D is set to be $g'(D')$ and Γ is set to be $\{f|_D : f\in\Gamma'\}\cup\mathcal{C}_D$. Thus, the intuition behind moving to the rigid core is that (a) one removes labels from the domain that can always be (uniformly) replaced in any solution to an instance without increasing its value, and (b) one allows constraints of the form u_d that can be used to fix labels for variables, leading to applicability of more powerful algebraic results.

2.3. Cyclic and symmetric operations. Several types of operations play a special role in the algebraic approach to (V)CSP.

DEFINITION 2.12. *An operation* $g\in\mathcal{O}_D^{(m)}$, $m\geq 2$, *is called*

- idempotent *if* $g(x,\dots,x)=x$ *for all* $x\in D$;
- Taylor *if, for each* $1\leq i\leq m$, *it satisfies an identity of the form* $g(\triangle_1,\triangle_2,\dots,\triangle_m)=g(\square_1,\square_2,\dots,\square_m)$ *where all* $\triangle_j,\square_j$ *are in* $\{x,y\}$ *and* $\triangle_i\neq\square_i$.
- cyclic *if* $g(x_1,x_2,\dots,x_m)=g(x_2,\dots,x_m,x_1)$ *for all* $x_1,\dots,x_m\in D$;
- symmetric *if* $g(x_1,x_2,\dots,x_m)=g(x_{\pi(1)},x_{\pi(2)},\dots,x_{\pi(m)})$ *for all* $x_1,\dots,x_m\in D$, *and any permutation* π *on* $[m]$.

A fractional operation ω *is said to be idempotent/cyclic/symmetric if all operations in* $\mathrm{supp}(\omega)$ *have the corresponding property.*

It is well known and easy to see that all polymorphisms and fractional polymorphisms of a rigid core are idempotent.

The following lemma is contained in the proof of Theorem 50 in [38].

LEMMA 2.13. *Let* Γ *be a rigid core on a set* D. *Then the following are equivalent:*

1. $\mathrm{fPol}^+(\Gamma)$ *contains a Taylor operation of arity at least* 2;
2. Γ *has a cyclic fractional polymorphism of (some) arity at least* 2;
3. Γ *has a cyclic fractional polymorphism of every prime arity* $p>|D|$.

The following theorem is Corollary 51 from [38].

THEOREM 2.14. [38] *Let* Γ *be a valued constraint language that is a rigid core. If* $\mathrm{fPol}^+(\Gamma)$ *does not contain a Taylor operation then* $\mathrm{VCSP}(\Gamma)$ *is NP-hard.*

Kozik and Ochremiak state a conjecture (which they attribute to L. Barto) that the above theorem describes all NP-hard valued constraint languages, and all other languages are tractable. Using Lemma 2.13, we restate the original conjecture via cyclic fractional polymorphisms.

CONJECTURE 2.15. [37] *Let* Γ *be a valued constraint language that is a rigid core. If* Γ *has a cyclic fractional polymorphism of arity at least* 2, *then* $\mathrm{VCSP}(\Gamma)$ *is tractable.*

Note that, for a finite core Γ (but with fixed D), the above condition can be checked in polynomial time. Indeed, if $p > |D|$ is some fixed prime number, then it is sufficient to check for a cyclic fractional polymorphism of arity p. Such polymorphisms, by definition, are solutions to a system of linear inequalities. Since the number of cyclic operations of arity p on D is constant, the system will have size polynomial in Γ and its feasibility can be decided by linear programming.

For the case when (possibly infinite) Γ consists only of $\{0,\infty\}$-valued functions, $\mathrm{VCSP}(\Gamma)$ is actually a CSP. For such Γ, *any* probability distribution on polymorphisms (of the same arity) is a fractional polymorphism. Then a theorem and a conjecture (the latter now known as the algebraic CSP dichotomy conjecture) equivalent to Theorem 2.14 and Conjecture 2.15 were given in [10]. One of several equivalent forms of the algebraic CSP dichotomy conjecture is as follows.

CONJECTURE 2.16. [10, 3] *Let* Γ *be a valued constraint language that is a rigid core and that consists of* $\{0,\infty\}$*-valued functions. If* Γ *has a cyclic polymorphism of arity at least* 2, *then* $\mathrm{VCSP}(\Gamma)$ *is tractable. Otherwise,* $\mathrm{VCSP}(\Gamma)$ *is NP-hard.*

In view of this, it is natural to call Conjecture 2.15 the *algebraic VCSP dichotomy conjecture.*

2.4. Basic LP relaxation. Symmetric operations are known to be closely related to LP-based algorithms for CSP-related problems. One algorithm in particular has been known to solve many VCSPs to optimality. This algorithm is based on the so-called *basic LP relaxation*, or BLP, defined as follows.

Let $\mathbb{M}_n = \{\mu \geq 0 \mid \sum_{x\in D^n}\mu(x) = 1\}$ be the set of probability distributions over labelings in D^n. We also denote $\Delta = \mathbb{M}_1$; thus, Δ is the standard $(|D|-1)$-dimensional simplex. The corners of Δ can be identified with elements in D. For a distribution $\mu \in \mathbb{M}_n$ and a variable $v \in \{1,\dots,n\}$, let $\mu_{[v]} \in \Delta$ be the marginal probability of distribution μ for v:

$$\mu_{[v]}(a) \;=\; \sum_{x\in D^n : x_v = a} \mu(x) \qquad \forall a \in D.$$

Given a VCSP instance $\mathcal{I}$ in the form (1), we define the value $\mathrm{BLP}(\mathcal{I})$ as follows:

$$\begin{aligned}
\text{(3)} \qquad \mathrm{BLP}(\mathcal{I}) \;=\; \min \sum_{t\in T}\sum_{x\in \operatorname{dom} f_t} & \mu_t(x) f_t(x) \\
\text{s.t.} \quad (\mu_t)_{[k]} = \alpha_{v(t,k)} & \qquad \forall t\in T, k\in\{1,\dots,n_t\},\\
\mu_t \in \mathbb{M}_{n_t} & \qquad \forall t \in T,\\
\mu_t(x) = 0 & \qquad \forall t\in T, x\notin \operatorname{dom} f_t,\\
\alpha_v \in \Delta & \qquad \forall v \in V.
\end{aligned}$$

If there are no feasible solutions then $\mathrm{BLP}(\mathcal{I}) = \infty$. The objective function and all constraints in this system are linear, therefore this is a linear program. Its size is polynomial in the size of $\mathcal{I}$, so $\mathrm{BLP}(\mathcal{I})$ can be found in time polynomial in $|\mathcal{I}|$.

We say that BLP *solves* $\mathcal{I}$ if $\mathrm{BLP}(\mathcal{I}) = \min_{x \in D^n} f_{\mathcal{I}}(x)$, and BLP solves $\mathrm{VCSP}(\Gamma)$ if it solves all instances $\mathcal{I}$ of $\mathrm{VCSP}(\Gamma)$. If BLP solves $\mathrm{VCSP}(\Gamma)$ and Γ is a rigid core, then the optimal solution for every instance can be found by using the standard self-reducibility method. In this method, one goes through the variables in some order, finding $d \in D$ for the current variable v such that instances $\mathcal{I}$ and $\mathcal{I} + u_d(v)$ have the same optimal value (which can be checked by BLP), updating $\mathcal{I} := \mathcal{I} + u_d(v)$, and moving to the next variable. At the end, the instance will have a unique feasible assignment whose value is the optimum of the original instance. Note that in this case $\mathrm{VCSP}(\Gamma)$ is globally tractable.

THEOREM 2.17 ([35]). *BLP solves* $\mathrm{VCSP}(\Gamma)$ *if and only if, for every* $m > 1$, Γ *has a symmetric fractional polymorphism of arity* m.

THEOREM 2.18 ([35, 49]). *Let* Γ *be a rigid core constraint language that is finite-valued. If* Γ *has a symmetric fractional polymorphism of arity* 2 *then BLP solves* $\mathrm{VCSP}(\Gamma)$*, and so* $\mathrm{VCSP}(\Gamma)$ *is tractable. Otherwise,* $\mathrm{VCSP}(\Gamma)$ *is NP-hard.*

3. Main result.

DEFINITION 3.1. *Let* $\mathcal{I}$ *be a VCSP instance over variables* V *with domain* D. *The* feasibility instance, $\mathrm{Feas}(\mathcal{I})$*, associated with* $\mathcal{I}$ *is a CSP instance obtained from* $\mathcal{I}$ *by replacing each constraint function* f_t *with* $\mathrm{dom}\, f_t$.

For a language Γ, let $\mathrm{Feas}(\Gamma) = \{\mathrm{dom}\, f \mid f \in \Gamma\}$. Then the instances of the problem $\mathrm{CSP}(\mathrm{Feas}(\Gamma))$ are the instances $\mathrm{Feas}(\mathcal{I})$ where $\mathcal{I}$ runs through all instances of $\mathrm{VCSP}(\Gamma)$.

DEFINITION 3.2. *Let* $\mathcal{I}$ *be a VCSP instance over variables* V *with domain* D. *For each variable* $v \in V$, *let* $D_v = \{d \in D \mid d = \sigma(v)$ *for some feasible solution* σ *for* $\mathcal{I}\}$. *Then* $(1,\infty)$-minimal instance $\bar{\mathcal{I}}$ *associated with* $\mathcal{I}$ *is the VCSP instance obtained from* $\mathcal{I}$ *by adding, for each* $v \in V$, *the constraint* $u_{D_v}(x_v)$.

Note that if Γ is a rigid core and the problem $\mathrm{CSP}(\mathrm{Feas}(\Gamma))$ is tractable, then, for any instance $\mathcal{I}$ of $\mathrm{VCSP}(\Gamma)$, one can construct the associated $(1,\infty)$-minimal instance in polynomial time. Indeed, to find out whether a given $d \in D$ is in D_v, one only needs to decide whether the CSP instance obtained from $\mathrm{Feas}(\mathcal{I})$ by adding the constraint $u_d(x_v)$ is satisfiable. Since Γ is a rigid core, the latter instance is also an instance of $\mathrm{CSP}(\mathrm{Feas}(\Gamma))$.

If Γ is a rigid core then, for $\mathrm{VCSP}(\Gamma)$ to be tractable, Γ must satisfy the assumption of Conjecture 2.15, and also, clearly, the feasibility part of the problem, $\mathrm{CSP}(\mathrm{Feas}(\Gamma))$, must be tractable. Our main result shows that if these necessary conditions are satisfied then $\mathrm{VCSP}(\Gamma)$ is indeed tractable.

THEOREM 3.3. *Let* Γ *be a valued constraint language over domain* D *that is a rigid core. If the following conditions hold then* $\mathrm{VCSP}(\Gamma)$ *is tractable:*

1. Γ *has a cyclic fractional polymorphism of arity at least* 2, *and*
2. $\mathrm{CSP}(\mathrm{Feas}(\Gamma))$ *is tractable.*

Otherwise, $\mathrm{VCSP}(\Gamma)$ *is not tractable.*

In Theorem 3.3, the intractability part for (absence of) the first condition follows from Theorem 2.14, and it is obvious for the second condition. The tractability part follows from Theorem 3.4 below.

THEOREM 3.4. *Let Γ be an arbitrary language that has a cyclic fractional polymorphism of arity at least* 2*. If $\mathcal{I}$ is an instance of* VCSP(Γ) *and $\bar{\mathcal{I}}$ is its associated* $(1,\infty)$*-minimal instance, then* $\mathrm{Opt}(\mathcal{I}) = \mathrm{BLP}(\bar{\mathcal{I}})$.

Indeed, if Γ is a rigid core satisfying conditions (1) and (2) from Theorem 3.3 and $\mathcal{I}$ is an instance of VCSP(Γ) then the equality $\mathrm{Opt}(\mathcal{I}) = \mathrm{BLP}(\bar{\mathcal{I}})$ means that we can efficiently find the optimum value for $\mathcal{I}$ by constructing $\bar{\mathcal{I}}$ (which we can do efficiently because Γ is a rigid core and CSP(Feas(Γ)) is tractable) and then applying BLP to $\bar{\mathcal{I}}$. Then we can find an optimal assignment for $\mathcal{I}$ by self-reduction (see the discussion before Theorem 2.17).

Recall the notion of global tractability from section 2.1. The algorithm that we just described gives the following.

COROLLARY 3.5. *Let Γ be a valued constraint language over domain D that is a rigid core. If*

1. *Γ has a cyclic fractional polymorphism of arity at least* 2*, and*
2. CSP(Feas(Γ)) *is globally tractable,*

then VCSP(Γ) *is globally tractable.*

It also follows from Theorem 3.4 that, for every language Γ that has a cyclic fractional polymorphism of arity at least 2, VCSP(Γ) is polynomial-time equivalent to CSP(Feas(Γ)). In particular, *any* complexity classification of CSPs, whether it is the dichotomy as predicted by Conjecture 2.16 or anything else, gives a complexity classification of VCSPs.

Let us now discuss how Theorem 3.3 can be combined with known CSP complexity classifications to obtain new, previously unknown, VCSP classifications which are tighter than Theorem 3.3.

As we explained in section 1, if the algebraic CSP dichotomy conjecture holds, then condition (2) in Theorem 3.3 can be omitted and all intractable VCSPs are NP-hard. Since this conjecture holds when $|D| \leq 3$ [9, 45] or when D is arbitrary finite, but Γ contains all unary crisp functions [1, 11], we get the following corollaries.

COROLLARY 3.6. *Let $|D| \leq 3$ and let Γ be a valued constraint language that is a rigid core on D. If Γ has a cyclic fractional polymorphism then the problem* VCSP(Γ) *is tractable, otherwise it is NP-hard.*

For the case $|D| = 2$, the tractable cases can be characterized by six specific cyclic fractional polymorphisms [17], and it was shown in [38] that the presence of any cyclic fractional polymorphism (when $|D| = 2$) implies the presence of one of those six. Also, Corollary 3.6 generalizes results from [51, 52] where the dichotomy was shown for the special case when $|D| = 3$ and all noncrisp functions in Γ are unary. The specific conditions for tractability in [51, 52] have not been shown to be directly implied by the presence of a cyclic fractional polymorphism, though.

COROLLARY 3.7. *Let Γ be a valued constraint language on D that contains all unary crisp functions. If Γ has a cyclic fractional polymorphism then the problem* VCSP(Γ) *is tractable, otherwise it is NP-hard.*

Corollary 3.7 generalizes a result from [51] where the dichotomy was shown for the special case when Γ includes all unary crisp functions and all noncrisp functions in Γ are unary. Again, the specific condition for tractability in [51] is not known to be directly implied by the presence of a cyclic fractional polymorphism.

It is shown in [38] how Theorem 3.3 implies the dichotomy results (including specific conditions for tractability) for the finite-valued case from [49] (Theorem 2.18) and for the case when Γ contains all unary functions taking values in $\{0, 1\}$ [36]. The algorithm for the tractable case in [36] is somewhat similar in spirit to our algorithm, and actually inspired the latter.

Let us now explain how Theorem 3.3 implies the tractability result from [48] (stated below). An idempotent operation $g \in \mathcal{O}_D$ of arity at least 2 satisfying $g(y, x, x \ldots, x, x) = g(x, y, x, \ldots, x, x) = \cdots = g(x, x, x, \ldots, x, y)$ for all $x, y \in D$ is called a *weak near-unanimity* operation. The tractability result result from [48] states that if $\mathrm{fPol}^+(\Gamma)$ contains weak near-unanimity operations of all but finitely many arities, then $\mathrm{VCSP}(\Gamma)$ is tractable (in fact, via a specific algorithm based on Sherali–Adams hierarchy, which does not follow from our results). This condition on $\mathrm{fPol}^+(\Gamma)$ is well known in the algebraic approach to the CSP; it characterizes (when appropriately formulated) CSPs of bounded width [4]. So assume that $\mathrm{fPol}^+(\Gamma)$ satisfies this condition. Since $\mathrm{fPol}^+(\Gamma) \subseteq \mathrm{Pol}(\Gamma)$, the set $\mathrm{Pol}(\Gamma)$ also contains these operations, so $\mathrm{CSP}(\mathrm{Feas}(\Gamma))$ is tractable by [4]. Moreover, by [3], $\mathrm{fPol}^+(\Gamma)$ then also contains a cyclic operation of arity at least 2. Now (the proof of) Theorem 50 of [38] implies that Γ has a cyclic fractional polymorphism of arity at least 2, and then tractability of $\mathrm{VCSP}(\Gamma)$ follows from Theorem 3.3.

We remark that some known VCSP classifications with tighter and more explicit characterizations of tractability can be easily derived from our main result; e.g., the classification for the Boolean case ($|D| = 2$) can be easily derived following the lines of section 8 of [13]. However, it might take additional effort to derive some others — for example, the dichotomy result from [50] was proved without using our theorem, and it is not known how to derive it from our main result.

4. Proof of Theorem 3.4: Reduction to a block-finite language. We will prove Theorem 3.4 by constructing, from a given (feasible) instance $\mathcal{I}$, a finite valued constraint language Γ' on some finite set D' and an instance $\mathcal{I}'$ of $\mathrm{VCSP}(\Gamma')$ such that $\mathrm{Opt}(\mathcal{I}) = \mathrm{Opt}(\bar{\mathcal{I}}) = \mathrm{Opt}(\mathcal{I}') = \mathrm{BLP}(\mathcal{I}') = \mathrm{BLP}(\bar{\mathcal{I}})$. The first equality is immediate from the definition $\bar{\mathcal{I}}$, the second one will follow trivially from the construction of Γ' and $\mathcal{I}'$, and the last equality holds by Lemma 4.1 below, while the key equality $\mathrm{Opt}(\mathcal{I}') = \mathrm{BLP}(\mathcal{I}')$ will follow from the fact that BLP solves $\mathrm{VCSP}(\Gamma')$ that we prove, using Theorem 2.17, in Theorem 4.4. The construction is inspired by [34], where a similar technique of "lifting" a language was used in a different context.

Let V be the set of variables of instance $\mathcal{I}$, and let

$$f_{\mathcal{I}}(x) \;=\; \sum_{t \in T} f_t(x_{v(t,1)}, \ldots, x_{v(t,n_t)}) \qquad \forall x : V \to D \tag{4}$$

be its objective function. For the $(1, \infty)$-minimal instance $\bar{\mathcal{I}}$, the objective function is

$$f_{\bar{\mathcal{I}}}(x) \;=\; \sum_{t \in T} f_t(x_{v(t,1)}, \ldots, x_{v(t,n_t)}) + \sum_{v \in V} u_{D_v}(x_v) \qquad \forall x : V \to D. \tag{5}$$

Now let $D'_v = \{(v, a) \mid a \in D_v\}$ be a unique copy of D_v. We now define a new language Γ' over domain $D' = \bigcup_{v \in V} D'_v$ as follows:

$$\Gamma' = \bigcup_{t \in T} \left\{ f_t^{\langle v(t,1), \ldots, v(t,n_t) \rangle}, \mathrm{dom}\, f_t^{\langle v(t,1), \ldots, v(t,n_t) \rangle} \right\} \cup \bigcup_{v \in V} \left\{ u_{D'_v} \right\} \cup \{=_{D'}\},$$

where functions $u_{D'_v}$ are as defined above, $=_{D'}$ is the binary $\{0, \infty\}$-valued function corresponding to the equality relation, and, for an n-ary function f over D and variables $v_1, \ldots, v_n \in V$, we define function $f^{\langle v_1, \ldots, v_n \rangle} : (D')^n \to \overline{\mathbb{Q}}$ as follows:

$$f^{\langle v_1,\ldots,v_n\rangle}(x) = \begin{cases} f(\hat{x}) & \text{if } x = ((v_1,\hat{x}_1),\ldots,(v_n,\hat{x}_n)) \\ \infty & \text{otherwise} \end{cases} \qquad \forall x \in (D')^n.$$

The above mentioned instance $\mathcal{I}'$ of VCSP(Γ') is obtained from $\bar{\mathcal{I}}$ by replacing each function f_t with $f_t^{\langle v(t,1),\ldots,v(t,n_t)\rangle}$ and replacing each function u_{D_v} with $u_{D'_v}$.

It is straightforward to check that there is a one-to-one correspondence between the sets of feasible solutions to BLP relaxations for $\mathcal{I}'$ and $\bar{\mathcal{I}}$, and that this correspondence also preserves the values of the solutions.

LEMMA 4.1. *We have* $\text{BLP}(\mathcal{I}') = \text{BLP}(\bar{\mathcal{I}})$.

LEMMA 4.2. *If* Γ *has a cyclic fractional polymorphism of arity* $m > 1$ *then* Γ' *has the same property.*

Proof. Let ω be a cyclic fractional polymorphism of Γ. Fix an arbitrary element $d' \in D'$. For each operation $g \in \text{supp}(\omega)$, define the operation g' on D' as follows:

$$g'(x_1,\ldots,x_m) = \begin{cases} (v, g(\hat{x}_1,\ldots,\hat{x}_m)) & \text{if } x_1 = (v,\hat{x}_1),\ldots,x_m = (v,\hat{x}_m) \text{ for some } v \in V, \\ d' & \text{otherwise.} \end{cases}$$

Clearly, each operation g' is cyclic. Consider the fractional operation ω' on D' such that $\omega(g') = \omega(g)$ for all $g \in \mathit{supp}(\omega)$. It is straightforward to check that ω' is a fractional polymorphism of Γ'. ∎

To prove Theorem 3.4, it remains to show that $\text{Opt}(\mathcal{I}') = \text{BLP}(\mathcal{I}')$. We will prove the more general fact that BLP solves VCSP(Γ'). The properties of the language Γ' that we use for this (apart from having a cyclic fractional polymorphism) are given below in Definition 4.3.

DEFINITION 4.3. *A finite language* Γ *is called* block-finite *if its domain* D *can be partitioned into disjoint subsets* $\{D_v \mid v \in V\}$ *such that*

(a) *For any* $a \in D_v$ *with* $v \in V$ *there exists a polymorphism* $g_a \in \mathcal{O}^{(1)}$ *of* Feas(Γ) *such that* $g_a(b) = a$ *for all* $b \in D_v$.

(b) *For any* n*-ary function* $f \in \Gamma$*, the relation* dom f *(viewed as a function* $D^n \to \{0,\infty\}$*) belongs to* Γ*. Furthermore, the binary equality relation on* D*, denoted as* $=_D\colon D^2 \to \{0,\infty\}$*, also belongs to* Γ*.*

(c) *Any* n*-ary function* $f \in \Gamma - \{=_D\}$ *satisfies* dom $f \subseteq D_{v_1} \times \cdots \times D_{v_n}$ *for some* $v_1,\ldots,v_n \in V$*.*

It is easy to see that the language Γ' defined in the previous section is block-finite. It obviously has properties (b) and (c), and it has property (a) because the instance $\mathcal{I}'$ is $(1,\infty)$-minimal. Indeed, if $a = (v,d) \in D'_v$ then, by definition, $\mathcal{I}$ has a feasible solution $\sigma : V \to D$ with $\sigma(v) = d$. Define function g_a as follows: for each $a' = (v',d') \in D'$, set $g_a(a') = (v', \sigma(v'))$. It is easy to check that g_a has the required properties.

From now on, we forget about the original language Γ from the previous section and about the specific language Γ' and work with an arbitrary block-finite language that has a cyclic fractional polymorphism of arity at least 2. For simplicity, we denote our language by Γ. Note that Γ is not necessarily a (rigid) core, but this property is not required in Theorem 2.17. By Theorem 2.17, in order to prove Theorem 3.4, it remains to show the following.

THEOREM 4.4. *Suppose that a block-finite language* Γ *admits a cyclic fractional polymorphism* ν *of arity at least* 2*. Then, for every* $m \geq 2$*,* Γ *admits a symmetric fractional polymorphism* ω_m^{sym} *of arity* m*.*

In the rest of the paper we prove Theorem 4.4. This will be done in two steps: (i) using the existence of ν, prove the existence of $\omega_2^{\mathtt{sym}}$; (ii) using the existence of $\omega_{m-1}^{\mathtt{sym}}$ for some $m \geq 3$, prove the existence of $\omega_m^{\mathtt{sym}}$. The claim will then follow by induction on m.

Note that for finite-valued languages step (i) was proved in [49] (or rather a very closely related statement), while step (ii) was established in [35]. However, in both cases it was essential that the language is finite-valued. The arguments in [35, 49] seem to break down when infinities are allowed. For example, we were unable to extend the approach in [49] that exploits the connectivity of a certain graph on D. To deal with block-finite languages, we will introduce (in section 6) a new technical tool where we first prove, via Farkas's lemma, the existence of a certain function with special properties in $\langle\Gamma\rangle$.

5. A graph of generalized operations. In this section we describe a basic tool that will be used for constructing new fractional polymorphisms, namely a graph of generalized operations introduced in [35].

Let $\mathcal{O}^{(m\to m)}$ be the set of mappings $\mathbf{g} : D^m \to D^m$ and let $\mathbb{1} \in \mathcal{O}^{(m\to m)}$ be the identity mapping. Consider a sequence x of m labelings $x \in [D^n]^m$; this means that $x = (x^1, \ldots, x^m)$ where $x^i \in D^n$ (think of x as an $m\times n$ matrix whose rows are $x^1, \ldots, x^m$). For an n-ary function f, we define $f^m(x) = \frac{1}{m}(f(x^1)+\cdots+f(x^m))$ (thus $f^m(x)$ is the average value of f on the rows of x). For a mapping $\mathbf{g} = (g_1, \ldots, g_m) \in \mathcal{O}^{(m\to m)}$, we also denote $x^{\mathbf{g}i} = g_i(x)$ for $i \in [m]$ and $\mathbf{g}(x) = (x^{\mathbf{g}1}, \ldots, x^{\mathbf{g}m})$ (so $g(x)$ is an $m \times n$ matrix where row i is obtained by column-wise application of g_i to x).

A probability distribution ρ over $\mathcal{O}^{(m\to m)}$ will be called a *(generalized) fractional polymorphism of* Γ *of arity* $m \to m$ if each function $f \in \Gamma$ satisfies

$$\sum_{\mathbf{g}\in\mathrm{supp}(\rho)} \rho(\mathbf{g}) f^m(\mathbf{g}(x)) \leq f^m(x) \qquad \forall x \in [\mathrm{dom}\, f]^m. \tag{6}$$

We will sometimes represent fractional polymorphisms of arity m and generalized fractional polymorphisms of arity $m \to m$ as vectors in $\mathbb{R}^{\mathcal{O}^{(m)}}$ and $\mathbb{R}^{\mathcal{O}^{(m\to m)}}$, respectively. For $g \in \mathcal{O}^{(m)}$ and $\mathbf{g} \in \mathcal{O}^{(m\to m)}$, we denote the corresponding characteristic vectors by χ_g and $\chi_{\mathbf{g}}$ respectively. It can be checked that a generalized fractional polymorphism ρ of arity $m \to m$ can be converted into a fractional polymorphism ρ' of arity m, as follows:

$$\rho' = \sum_{\mathbf{g}=(g_1,\ldots,g_m)\in\mathrm{supp}(\rho)} \frac{\rho(\mathbf{g})}{m}(\chi_{g_1} + \cdots + \chi_{g_m}).$$

We will use the following construction in several parts of the proof. Assume that we have some probability distribution ω with a finite support such that (i) each element $s \in \mathrm{supp}(\omega)$ corresponds to an element of $\mathcal{O}^{(m\to m)}$ denoted as $\mathbb{1}^s$, and (ii) this distribution satisfies the following property for each $f \in \Gamma$:

$$\sum_{s\in\mathrm{supp}(\omega)} \omega(s) f^m(\mathbb{1}^s(x)) \leq f^m(x) \qquad \forall x \in [\mathrm{dom}\, f]^m. \tag{7a}$$

Condition (7a) then can be rephrased as saying that vector $\sum_{s\in\mathrm{supp}(\omega)} \omega(s)\chi_{\mathbb{1}^s}$ is a fractional polymorphism of Γ of arity $m \to m$. We will also consider the following condition:

$$\sum_{s\in\mathrm{supp}(\omega)} \omega(s) f(x^{\mathbb{1}^s i}) \leq f^{m-1}(x_{-i}) \qquad \forall x \in [\mathrm{dom}\, f]^m, i \in [m], \tag{7b}$$

where $x_{-i} \in [\mathrm{dom}\, f]^{m-1}$ denotes the sequence of $m-1$ labelings obtained from x by removing the ith labeling. Note that condition (7b) implies (7a) (since summing (7b) over $i \in [m]$ and dividing by m gives (7a)). The second condition will be used only in one of the results; unless noted otherwise, ω is only assumed to satisfy (7a).

For a mapping $\mathbf{g} \in \mathcal{O}^{(m\to m)}$ denote $\mathbf{g}^s = \mathbb{1}^s \circ \mathbf{g}$. (This notation is consistent with the earlier one since $\mathbb{1}^s \circ \mathbb{1} = \mathbb{1}^s$ for any s). We use $\mathbf{g}^{s_1 \cdots s_k}$ to denote $(\cdots(\mathbf{g}^{s_1})^{\cdots})^{s_k} = \mathbb{1}^{s_k} \circ \cdots \circ \mathbb{1}^{s_1} \circ \mathbf{g}$. Next, define a directed graph $(\mathbb{G}, E)$ as follows:

- $\mathbb{G} = \{\mathbb{1}^{s_1 \cdots s_k} \mid s_1, \ldots, s_k \in \mathrm{supp}(\omega), k \ge 0\}$ is the set of all mappings that can be obtained from $\mathbb{1}$ by applying operations from $\mathrm{supp}(\omega)$;
- $E = \{(\mathbf{g}, \mathbf{g}^s) \mid \mathbf{g} \in \mathbb{G}, s \in \mathrm{supp}(\omega)\}$.

This graph can be decomposed into strongly connected components, yielding a directed acyclic graph (DAG) on these components. We define $\mathsf{Sinks}(\mathbb{G}, E)$ to be the set of those strongly connected components $\mathbb{H} \subseteq \mathbb{G}$ of $(\mathbb{G}, E)$ that are sinks of this DAG (i.e., have no outgoing edges). Any DAG has at least one sink, therefore $\mathsf{Sinks}(\mathbb{G}, E)$ is nonempty. We denote $\mathbb{G}^* = \bigcup_{\mathbb{H}\in \mathsf{Sinks}(\mathbb{G},E)} \mathbb{H} \subseteq \mathbb{G}$ and $Range_n(\mathbb{G}^*) = \{\mathbf{g}^*(x) \mid \mathbf{g}^* \in \mathbb{G}^*, x \in [D^n]^m\}$. Also, for a tuple $\hat{x} \in D^m$ we will denote $\mathbb{G}(\hat{x}) = \{\mathbf{g}(\hat{x}) | \mathbf{g} \in \mathbb{G}\} \subseteq D^m$.

The following facts can easily be shown (see Appendices A.1 and A.2).

Proposition 5.1.
(a) *If $\mathbf{g}, \mathbf{h} \in \mathbb{G}$ then $\mathbf{h} \circ \mathbf{g} \in \mathbb{G}$. Moreover, if $\mathbf{g} \in \mathbb{H} \in \mathsf{Sinks}(\mathbb{G}, E)$ then $\mathbf{h} \circ \mathbf{g} \in \mathbb{H}$.*
(b) *Consider connected components $\mathbb{H}', \mathbb{H} \in \mathsf{Sinks}(\mathbb{G}, E)$. For each $\mathbf{g}' \in \mathbb{H}'$ there exists $\mathbf{g} \in \mathbb{H}$ satisfying $\mathbf{g} \circ \mathbf{g}' = \mathbf{g}'$.*
(c) *For each $x \in Range_n(\mathbb{G}^*)$ and $\mathbb{H} \in \mathsf{Sinks}(\mathbb{G}, E)$ there exists $\mathbf{g} \in \mathbb{H}$ satisfying $\mathbf{g}(x) = x$.*

Proposition 5.2. *Suppose that $\hat{x} \in Range_1(\mathbb{G}^*)$ and $x \in \mathbb{G}(\hat{x})$.*
(a) *There holds $x \in Range_1(\mathbb{G}^*)$.*
(b) *There exists $\mathbf{g} \in \mathbb{G}$ such that $\mathbf{g}(x) = \hat{x}$.*

We now state main theorems related to the graph $(\mathbb{G}, E)$, that are slight extensions of the results in [35]. Their proofs use the same techniques as [35] and can be found in Appendices A.3, A.4, and A.5.

Theorem 5.3. *Let $\widehat{\mathbb{G}}$ be a subset of $\mathbb{G}$ satisfying the following property: for each $\mathbf{g} \in \mathbb{G}$ there exists a path in $(\mathbb{G}, E)$ from $\mathbf{g}$ to some node $\widehat{\mathbf{g}} \in \widehat{\mathbb{G}}$. Then there exists a fractional polymorphism ρ of Γ of arity $m \to m$ with $\mathrm{supp}(\rho) = \widehat{\mathbb{G}}$.*

We will use this result either for the set $\widehat{\mathbb{G}} = \mathbb{G}$ or for the set $\widehat{\mathbb{G}} = \mathbb{G}^*$; clearly, both choices satisfy the condition of the theorem. The first choice gives that Γ admits a fractional polymorphism ρ with $\mathrm{supp}(\rho) = \mathbb{G}$; therefore, if $\mathbf{g} \in \mathbb{G}$, $f \in \langle\Gamma\rangle$ and $x \in [\mathrm{dom}\, f]^m$ then $\mathbf{g}(x) \in [\mathrm{dom}\, f]^m$.

Theorem 5.4. *Consider function $f \in \langle\Gamma\rangle$ of arity n and labelings $x \in Range_n(\mathbb{G}^*) \cap [\mathrm{dom}\, f]^m$.*
(a) *There holds $f^m(\mathbf{g}(x)) = f^m(x)$ for any $\mathbf{g} \in \mathbb{G}$.*
(b) *Suppose that condition (7b) holds. Then there exists a probability distribution λ over $\mathbb{G}^*$ (which is independent of f, x) such that $f^{\lambda}_{i'}(x) = f^{\lambda}_{i''}(x)$ for any $i', i'' \in [m]$ where*

$$f^{\lambda}_i(x) = \sum_{\mathbf{g}\in\mathbb{G}^*} \lambda_{\mathbf{g}} f(x^{\mathbf{g}i}). \tag{8}$$

6. Constructing special functions. In this section, we construct special functions in $\langle\Gamma\rangle$ that play an important role in the proof of Theorem 4.4.

For a sequence $x = (x^1, \dots, x^m) \in D^m$ and a permutation π of $[m]$, we define $x^\pi = (x^{\pi(1)}, \dots, x^{\pi(m)})$. Similarly, for a mapping $\mathbf{g} = (g_1, \dots, g_m) \in \mathcal{O}^{(m\to m)}$ define $\mathbf{g}^\pi = (g_{\pi(1)}, \dots, g_{\pi(m)})$. Let Ω be the set of mappings $\mathbf{g} \in \mathcal{O}^{(m\to m)}$ that satisfy the following condition:

Equivalently, $g_{\pi(i)}(x) = g_i(x^\pi)$ for any $i \in [m]$.

(9) $$\mathbf{g}^\pi(x) = \mathbf{g}(x^\pi) \text{ for any } x \in D^m \text{ and any permutation } \pi \text{ of } [m].$$

PROPOSITION 6.1. *If* $\mathbf{g}, \mathbf{h} \in \Omega$, *then* $\mathbf{g} \circ \mathbf{h} \in \Omega$.

Proof. Just note that

$$(\mathbf{g} \circ \mathbf{h})^\pi(x) = \mathbf{g}^\pi(\mathbf{h}(x)) = \mathbf{g}(\mathbf{h}^\pi(x)) = \mathbf{g}(\mathbf{h}(x^\pi)) = (\mathbf{g} \circ \mathbf{h})(x^\pi)$$

for any $x \in D^m$. ◻

Consider all generalized fractional polymorphisms ω of Γ of arity $m \to m$ satisfying $\mathrm{supp}(\omega) \subseteq \Omega$. At least one such polymorphism exists, namely $\omega = \chi_{\mathbb{1}}$ where $\mathbb{1} \in \mathcal{O}^{(m\to m)}$ is the identity mapping. Among such ω's, pick one with the largest support. It exists due to the following observation: if ω', ω'' are generalized fractional polymorphisms of Γ of arity $m \to m$ then so is the vector $\omega = \frac{1}{2}[\omega' + \omega'']$, and $\mathrm{supp}(\omega) = \mathrm{supp}(\omega') \cup \mathrm{supp}(\omega'')$.

Let us apply the construction of section 5 starting with the chosen distribution ω, where for $\mathbf{g} \in \mathrm{supp}(\omega)$ we define operation $\mathbb{1}^{\mathbf{g}} \in \mathcal{O}^{(m\to m)}$ via $\mathbb{1}^{\mathbf{g}} = \mathbf{g}$. Let the resulting graph be $(\mathbb{G}, E)$. It is straightforward to check that condition (7a) holds: it simply expresses the fact that ω is a generalized fractional polymorphsism of Γ of arity $m \to m$.

PROPOSITION 6.2. *It holds that* $\mathrm{supp}(\omega) = \mathbb{G}$.

Proof. If $\mathbf{g} \in \mathrm{supp}(\omega)$ then $\mathbf{g} = \mathbb{1}^{\mathbf{g}} \in \mathbb{G}$. Conversely, suppose that $\mathbf{g} \in \mathbb{G}$. We can write $\mathbf{g} = \mathbb{1}^{\mathbf{g}_k} \circ \cdots \circ \mathbb{1}^{\mathbf{g}_1} = \mathbf{g}_k \circ \cdots \circ \mathbf{g}_1$ with $\mathbf{g}_1, \dots, \mathbf{g}_k \in \mathrm{supp}(\omega) \subseteq \Omega$. Since Ω is closed under composition by Proposition 6.1, we get $\mathbf{g} \in \Omega$. By Theorem 5.3 there exists a generalized fractional polymorphism ρ with $\mathrm{supp}(\rho) = \mathbb{G}$, and so $\mathbf{g} \in \mathrm{supp}(\rho)$. By maximality of ω we get $\mathbf{g} \in \mathrm{supp}(\omega)$. ◻

In the remainder of this section we prove the following theorem.

THEOREM 6.3. *For any* $\hat{x} \in D^m$ *there exists a function* $f \in \langle\Gamma\rangle$ *of arity* m *with* $\arg\min f = \mathbb{G}(\hat{x})$.

Proof. Let Γ^+ be the set of pairs (f, x) with $f \in \Gamma$ and $x \in [\mathrm{dom}\, f]^m$. Let $\Omega' \subseteq \Omega$ be the set of mappings $\mathbf{g} \in \Omega$ that satisfy $\mathbf{g}(x) \in [\mathrm{dom}\, f]^m$ for all $(f, x) \in \Gamma^+$. Note that $\mathbb{G} = \mathrm{supp}(\omega) \subseteq \Omega'$. By the choice of ω, the following system does not have a solution with rational ρ:

(10a) $$\rho(\mathbf{g}) \geq 0 \qquad \forall \mathbf{g} \in \Omega',$$

(10b) $$\sum_{\mathbf{g}\in\Omega'} \rho(\mathbf{g}) f^m(x) - \sum_{\mathbf{g}\in\Omega'} \rho(\mathbf{g}) f^m(\mathbf{g}(x)) \geq 0 \qquad \forall (f, x) \in \Gamma^+,$$

(10c) $$\sum_{\mathbf{g}\in\Omega'-\mathbb{G}} -\rho(\mathbf{g}) < 0.$$

Next, we use the following well-known result (see, e.g., [46]).

LEMMA 6.4 (Farkas's lemma). *Let* A *be a* $p \times q$ *matrix and* b *be a* p*-dimensional vector. Then exactly one of the following is true:*

- *There exists* $\lambda \in \mathbb{R}^q$ *such that* $A\lambda = b$ *and* $\lambda \geq 0$.
- *There exists* $\mu \in \mathbb{R}^p$ *such that* $\mu^T A \geq 0$ *and* $\mu^T b < 0$.

If A *and* b *are rational then* λ *and* μ *can also be chosen in* $\mathbb{Q}^q$ *and* $\mathbb{Q}^p$*, respectively.*

By this lemma, the following system has a solution with rational $\lambda \geq 0$:

$$\lambda(\mathbf{g}) + \sum_{(f,x)\in\Gamma^+} \lambda(f,x)(f^m(x) - f^m(\mathbf{g}(x))) = 0 \qquad \forall \mathbf{g} \in \mathbb{G}, \tag{11a}$$

$$\lambda(\mathbf{g}) + \sum_{(f,x)\in\Gamma^+} \lambda(f,x)(f^m(x) - f^m(\mathbf{g}(x))) = -1 \qquad \forall \mathbf{g} \in \Omega' - \mathbb{G}. \tag{11b}$$

We will now define several instances of VCSP(Γ) where it will be convenient to use constraints with rational positive weights; these weights can always be made integer by multiplying the instances by an appropriate positive integer, which would not affect the reasoning, but make notation cumbersome.

We will define a Γ-instance $\mathcal{I}$ with $m|D|^m$ variables $\mathcal{V} = \{(i,z) \mid i \in [m], z \in D^m\}$. The labelings $\mathcal{V} \to D$ for this instance can be identified with mappings $\mathbf{g} = (g_1, \ldots, g_m) \in \mathcal{O}^{(m\to m)}$, if we define $\mathbf{g}(i,z) = g_i(z)$ for the coordinate $(i,z) \in \mathcal{V}$. We define the cost function of $\mathcal{I}$ as follows:

$$f_{\mathcal{I}}(\mathbf{g}) = \sum_{(f,x)\in\Gamma^+, \lambda(f,x)\neq 0} \lambda(f,x) f^m(\mathbf{g}(x)) \qquad \forall \mathbf{g} \in \mathcal{O}^{(m\to m)}. \tag{12}$$

From (11) we get

$$f_{\mathcal{I}}(\mathbb{1}) = f_{\mathcal{I}}(\mathbf{g}) - \lambda(\mathbf{g}) \ \leq \ f_{\mathcal{I}}(\mathbf{g}) \ < \ \infty \qquad \forall \mathbf{g} \in \mathbb{G}, \tag{13a}$$

$$f_{\mathcal{I}}(\mathbb{1}) < f_{\mathcal{I}}(\mathbf{g}) - \lambda(\mathbf{g}) \ \leq \ f_{\mathcal{I}}(\mathbf{g}) \ < \ \infty \qquad \forall \mathbf{g} \in \Omega' - \mathbb{G}. \tag{13b}$$

Furthermore, $f^m(\cdot)$ is invariant with respect to permuting its arguments, and thus

$$f_{\mathcal{I}}(\mathbf{g}) = f_{\mathcal{I}}(\mathbf{g}^\pi) \qquad \forall \mathbf{g} \in \mathcal{O}^{(m\to m)}, \text{ permutation } \pi \text{ of } [m]. \tag{13c}$$

Let T be the set of tuples (i,j,x,y) where $i,j \in [m]$, $x,y \in D^m$ and $i = \pi(j)$, $y = x^\pi$ for some permutation π of $[m]$. Define another Γ-instance $\mathcal{I}'$ with variables $\mathcal{V}$ and the cost function

$$f_{\mathcal{I}'}(\mathbf{g}) = f_{\mathcal{I}}(\mathbf{g}) \ + \sum_{(i,j,x,y)\in T} =_D (\mathbf{g}(i,x), \mathbf{g}(j,y)) + \sum_{(f,x)\in\Gamma^+} (\operatorname{dom} f)^m(\mathbf{g}(x)) \quad \forall \mathbf{g} \in \mathcal{O}^{(m\to m)}, \tag{14}$$

where $=_D$ is the equality relation on D. The instance $\mathcal{I}'$ is a Γ-instance because of condition (b) in the definition of a block-finite language. Note that the second term in (14) for $\mathbf{g} = (g_1, \ldots, g_m)$ equals 0 if $g_{\pi(j)}(x) = g_j(x^\pi)$ for all $j \in [m]$, $x \in D^m$ and permutation π of $[m]$. Otherwise the second term equals ∞. In other words, the second term is zero if and only if mapping $\mathbf{g}$ satisfies condition (9), i.e., if and only if $\mathbf{g} \in \Omega$. Similarly, the third term in (14) is zero if $\mathbf{g} \in \Omega'$, and ∞ if $\mathbf{g} \in \Omega - \Omega'$. We obtain that

$$f_{\mathcal{I}'}(\mathbb{1}) \leq f_{\mathcal{I}'}(\mathbf{g}) \ < \ \infty \qquad \forall \mathbf{g} \in \mathbb{G}, \tag{15a}$$

$$f_{\mathcal{I}'}(\mathbb{1}) < f_{\mathcal{I}'}(\mathbf{g}) \ < \ \infty \qquad \forall \mathbf{g} \in \Omega' - \mathbb{G}, \tag{15b}$$

$$f_{\mathcal{I}'}(\mathbf{g}) = \infty \qquad \forall \mathbf{g} \in \mathcal{O}^{(m\to m)} - \Omega'. \tag{15c}$$

These equations imply that $\mathbb{1} \in \arg\min f_{\mathcal{I}'} \subseteq \mathbb{G}$. We will show next that $\arg\min f_{\mathcal{I}'} = \mathbb{G}$.

For an index $k \in \mathbb{Z}$ let $\bar{k} \in [m]$ be the unique index with $\bar{k} - k = 0 \pmod m$. Let π_k be the cyclic permutation of $[m]$ with $\pi_k(1) = \bar{k}$. In particular, π_1 is the identity permutation. Also, for $k \in \mathbb{Z}$ let $e^k \in \mathcal{O}^{(m)}$ be the projection to the $\bar{k}$th coordinate. For a mapping $\mathbf{g} = (g_1, \ldots, g_m) \in \mathcal{O}^{(m\to m)}$ and a tuple $z \in D^m$ we will denote $\mathbf{g}(k, z) = \mathbf{g}(\bar{k}, z) \in D$. From the definition, for any permutation π of $[m]$ and any $(i, z) \in \mathcal{V}$ we have $\mathbf{g}^\pi(i, z) = (g_{\pi(1)}, \ldots, g_{\pi(m)})(i, z) = g_{\pi(i)}(z) = \mathbf{g}(\pi(i), z)$. In particular, $\mathbf{g}^{\pi_j}(i, z) = \mathbf{g}(i + j - 1, z)$.

From (13c) we have $f_{\mathcal{I}}(\mathbb{1}^{\pi_1}) = \cdots = f_{\mathcal{I}}(\mathbb{1}^{\pi_m}) = f_{\mathcal{I}}(\mathbb{1})$. Recall that $f_{\mathcal{I}} \in \langle\Gamma\rangle$ admits a generalized fractional polymorphism ω with $\mathrm{supp}(\omega) = \mathbb{G}$. Applying this polymorphism gives

$$\sum_{\mathbf{g}\in\mathrm{supp}(\omega)} \omega(\mathbf{g}) f^m_{\mathcal{I}}(\mathbf{g}(\mathbb{1}^{\pi_1}, \ldots, \mathbb{1}^{\pi_m})) \le f^m_{\mathcal{I}}(\mathbb{1}^{\pi_1}, \ldots, \mathbb{1}^{\pi_m}) = f_{\mathcal{I}}(\mathbb{1}). \tag{16}$$

Here we view $\mathbb{1}^{\pi_1}, \ldots, \mathbb{1}^{\pi_m}$ (and later $\mathbf{g}^{\pi_1}, \ldots, \mathbf{g}^{\pi_m}$) as labelings for the instance $\mathcal{I}$, while $\mathbf{g}$ is a mapping in $\mathcal{O}^{(m\to m)}$ acting on the first m labelings coordinate-wise. We claim that $\mathbf{g}(\mathbb{1}^{\pi_1}, \ldots, \mathbb{1}^{\pi_m}) = (\mathbf{g}^{\pi_1}, \ldots, \mathbf{g}^{\pi_m})$ for each $\mathbf{g} = (g_1, \ldots, g_m) \in \mathrm{supp}(\omega)$. Indeed, we need to show that $g_j(\mathbb{1}^{\pi_1}, \ldots, \mathbb{1}^{\pi_m}) = \mathbf{g}^{\pi_j}$ for each $j \in [m]$. Let us prove this for coordinate $(i, z) \in \mathcal{V}$. We can write

$$\begin{aligned} g_j(\mathbb{1}^{\pi_1}(i, z), \ldots, \mathbb{1}^{\pi_m}(i, z)) &= g_j(e^i(z), \ldots, e^{i+m-1}(z)) \\ &= g_j(z^{\pi_i}) \; = \; g_{\pi_i(j)}(z) \; = \; \mathbf{g}(i + j - 1, z) \; = \; \mathbf{g}^{\pi_j}(i, z), \end{aligned}$$

which proves the claim. We can now rewrite (16) as follows:

$$\sum_{\mathbf{g}\in\mathrm{supp}(\omega)} \omega(\mathbf{g}) f^m_{\mathcal{I}}(\mathbf{g}^{\pi_1}, \ldots, \mathbf{g}^{\pi_m}) \le f_{\mathcal{I}}(\mathbb{1}). \tag{17}$$

Using (13c) and the fact that $f_{\mathcal{I}}(\mathbf{g}) = f_{\mathcal{I}'}(\mathbf{g})$ for each $\mathbf{g} \in \mathrm{supp}(\omega) = \mathbb{G}$, we obtain

$$\sum_{\mathbf{g}\in\mathrm{supp}(\omega)} \omega(\mathbf{g}) f_{\mathcal{I}'}(\mathbf{g}) \le f_{\mathcal{I}'}(\mathbb{1}). \tag{18}$$

Since $\mathbb{1} \in \arg\min f_{\mathcal{I}'}$, we conclude that $\mathbb{G} = \mathrm{supp}(\omega) \subseteq \arg\min f_{\mathcal{I}'}$. Therefore, $\arg\min f_{\mathcal{I}'} = \mathbb{G}$.

We can finally prove Theorem 6.3. We define function $f \in \langle\Gamma\rangle$ with m variables as follows:

$$f(x) = \min_{\mathbf{g}\in\mathcal{O}^{(m\to m)}:\mathbf{g}(\hat{x})=x} f_{\mathcal{I}'}(\mathbf{g}) \qquad \forall x \in D^m.$$

Consider tuple $x \in D^m$. We have $x \in \arg\min f$ if and only if there exists $\mathbf{g} \in \arg\min f_{\mathcal{I}'} = \mathbb{G}$ with $\mathbf{g}(\hat{x}) = x$. The latter condition holds if and only if $x \in \mathbb{G}(\hat{x})$. ∎

7. Proof of Theorem 4.4.

We will prove the following result.

THEOREM 7.1. *Assume that one of the following holds:*

(a) $m = 2$ *and* Γ *admits a cyclic fractional polymorphism of arity at least* 2.

(b) $m \ge 3$ *and* Γ *admits a symmetric fractional polymorphism of arity* $m - 1$.

Let $f \in \langle\Gamma\rangle$ *be a function of arity* m *with* $\arg\min f = \mathbb{G}(\hat{x})$, *where* $\hat{x} \in Range_1(\mathbb{G}^*)$. *Then for every distinct pair of indices* $i, j \in [m]$ *there exists* $x \in \arg\min f$ *with* $x_i = x_j$.

We claim that this will imply Theorem 4.4. Indeed, we can use the following observation.

PROPOSITION 7.2. *Suppose that* $\hat{x} \in \mathit{Range}_1(\mathbb{G}^*)$*, and there exists* $x \in \mathbb{G}(\hat{x})$ *with* $x_i = x_j$ *for some* $i, j \in [m]$*. Then* $\hat{x}_i = \hat{x}_j$*.*

Proof. By Proposition 5.2(b), there exists $\mathbf{g} \in \mathbb{G}$ such that $\mathbf{g}(x) = \hat{x}$. Let π be the permutation of $[m]$ that swaps i and j. By the choice of x, we have $x^\pi = x$. We can write $\hat{x}_j = g_j(x) = g_{\pi(i)}(x) = g_i(x^\pi) = g_i(x) = \hat{x}_i$. This proves the claim. ∎

COROLLARY 7.3. *If the precondition of Theorem* 7.1 *holds, then* Γ *admits a symmetric fractional polymorphism of arity* m*.*

Proof. Using Theorem 6.3, Theorem 7.1 and Proposition 7.2, we conclude that for any $\hat{x} \in \mathit{Range}_1(\mathbb{G}^*)$ we have $\hat{x}_1 = \cdots = \hat{x}_m$. Indeed, by Theorem 6.3 there exists a function $f \in \langle\Gamma\rangle$ with $\mathbb{G}(\hat{x}) = \arg\min f$. Theorem 7.1 implies that the precondition of Proposition 7.2 holds for any distinct pair of indices $i, j \in [m]$, and therefore $\hat{x}_i = \hat{x}_j$.

By Theorem 5.3, there exists a generalized fractional polymorphism ρ of Γ of arity $m \to m$ with $\mathrm{supp}(\rho) = \mathbb{G}^*$. Vector $\sum_{\mathbf{g}=(g_1,\ldots,g_m)\in\mathbb{G}^*} \rho(\mathbf{g}) \frac{1}{m}[\chi_{g_1} + \cdots + \chi_{g_m}]$ is then an m-ary fractional polymorphism of Γ; all operations in its support are symmetric because $\mathbb{G}^* \subseteq \Omega$ and $\hat{x}_1 = \cdots = \hat{x}_m$ for any $\hat{x} \in \mathit{Range}_1(\mathbb{G}^*)$. ∎

It remains to prove Theorem 7.1. A proof of parts (a) and (b) of Theorem 7.1 is given in sections 7.1 and 7.2, respectively. In both parts we will need the following result; it exploits the fact that Γ is block-finite.

LEMMA 7.4. *Suppose that* $\hat{x} \in \mathit{Range}_1(\mathbb{G}^*)$*,* $x \in \mathbb{G}(\hat{x})$ *and* f *is an* m*-ary function in* $\langle\Gamma\rangle$ *with* $\arg\min f = \mathbb{G}(\hat{x})$*. Then* $(a, \ldots, a) \in \mathrm{dom}\, f$ *for any* $a \in \{x_1, \ldots, x_m\}$*.*

Proof. We say that a tuple $z \in D^m$ is *proper* if $z_1, \ldots, z_m \in D_v$ for some $v \in V$. We will show that x is proper; the lemma will then follow from condition (a) from the definition of a block-finite language and the fact that $x \in \mathrm{dom}\, f$.

Fix an arbitrary element $a \in D$, and define mapping $\mathbf{g} \in \mathcal{O}^{(m\to m)}$ as follows:

$$\mathbf{g}(z) = \begin{cases} z & \text{if } z \text{ is proper,} \\ (a, \ldots, a) & \text{otherwise.} \end{cases}$$

We claim that $\mathbf{g} \in \Omega$. Indeed, consider $z \in D^m$. If $\mathbf{g}(z) = z$, the condition (9) holds trivially. Otherwise, we can easily check that

$$\mathbf{g}^\pi(z) = (a, \ldots, a)^\pi = (a, \ldots, a) = \mathbf{g}(z^\pi),$$

and so the condition (9) holds either way.

Let us now show that the vector $\rho = \chi_{\mathbf{g}}$ is a generalized fractional polymorphism of Γ of arity $m \to m$. Checking inequality (6) for binary equality relation $f = (=_D)$ is straighforward. Consider function $f \in \Gamma - \{=_D\}$. Since Γ is block-finite, we have $\mathrm{dom}\, f \subseteq D_{v_1} \times \cdots \times D_{v_n}$ for some $v_1, \ldots, v_n \in V$. This implies that for any $x \in [\mathrm{dom}\, f]^m$ we have $\mathbf{g}(x) = x$ (this can be checked coordinate-wise). Therefore, we have an equality in (6).

By the results above we obtain that $\mathbf{g} \in \mathbb{G}$. We are now ready to prove that x is proper. Suppose that this is not true; then $\mathbf{g}(x) = (a, \ldots, a)$. We have $\hat{x} \in \mathit{Range}_1(\mathbb{G}^*)$ and $x \in \mathbb{G}(\hat{x})$, so by Proposition 5.2(a) we conclude that $x \in \mathit{Range}_1(\mathbb{G}^*)$. We also have $(a, \ldots, a) \in \mathbb{G}(x)$, so Proposition 7.2 gives that $x_1 = \cdots = x_m$. This means that x is proper, which contradicts the earlier assumption. ∎

7.1. Case $m = 2$: Proof of Theorem 7.1(a). We start with the following observation.

PROPOSITION 7.5. *If $(a, b) \in \mathbb{G}(\hat{x})$ then $(b, a) \in \mathbb{G}(\hat{x})$.*

Proof. Consider mapping $\bar{\mathbb{1}} = (e_2^2, e_2^1)$, where $e_2^k \in \mathcal{O}^{(2)}$ is the projection to the the kth variable. It can be checked that $\bar{\mathbb{1}} \in \Omega$, and $\chi_{\bar{\mathbb{1}}}$ is a generalized fractional polymorphism of Γ of arity $2 \to 2$. Therefore, $\bar{\mathbb{1}} \in \mathbb{G}$.

We have $(a, b) = \mathbf{g}(\hat{x})$ for some $\mathbf{g} \in \mathbb{G}$. We also have $(b, a) = (\bar{\mathbb{1}} \circ \mathbf{g})(\hat{x})$ and $\bar{\mathbb{1}} \circ \mathbf{g} \in \mathbb{G}$, and therefore $(b, a) \in \mathbb{G}(\hat{x})$. ∎

Denote $A = \{x_1 \mid x \in \mathbb{G}(\hat{x})\} \subseteq D$, and let a be an element in A that minimizes $f(a, a)$. Note that $(a, a) \in \operatorname{dom} f$ by Lemma 7.4. Condition $\arg\min f = \mathbb{G}(\hat{x})$ and Proposition 7.5 imply that $(a, b), (b, a) \in \arg\min f$ for some $b \in A$. By assumption, Γ admits a cyclic fractional polymorphism ν of some arity $r \geq 2$. Let us apply it to tuples $(a, b), (b, a), (a, a), \ldots, (a, a)$, where (a, a) is repeated $r - 2$ times:

$$\sum_{h \in \operatorname{supp}(\nu)} \nu(h) f(h(a, b, a, \ldots, a), h(b, a, a, \ldots, a)) \leq \frac{2}{r} f(a, b) + \frac{r-2}{r} f(a, a). \tag{19}$$

We have $h(a, b, a, \ldots, a) = h(b, a, a, \ldots, a)$ since ν is cyclic; denote this element as a_h. We claim that $a_h \in A$ for any $h \in \operatorname{supp}(\nu)$. Indeed, consider a unary function $u_A(x_1) = \min_{x_2} f(x_1, x_2)$. It can be checked that $\arg\min u_A = A$. Then the presence of u_A in $\langle \Gamma \rangle$ implies that after applying ν to $(a, b, a, \ldots, a)$ one gets

$$\sum_{h \in \operatorname{supp}(\nu)} \nu(h) u_A(a_h) \leq \frac{r-1}{r} u_A(a) + \frac{1}{r} u_A(b) = \min u_A,$$

and thus indeed $a_h \in \arg\min u_A = A$ for any $h \in \operatorname{supp}(\nu)$.

By the choice of a we have $f(a, a) \leq f(a_h, a_h)$ for any $h \in \operatorname{supp}(\nu)$. From (19) we thus get

$$f(a, a) \leq \frac{2}{r} f(a, b) + \frac{r-2}{r} f(a, a), \tag{20}$$

and so $f(a, a) \leq f(a, b)$, implying $(a, a) \in \arg\min f$.

7.2. Case $m \geq 3$: proof of theorem 7.1(b). We define binary function $\bar{f} \in \langle \Gamma \rangle$ as follows: $\bar{f}(a, b) = \min_{x \in D^m : x_i = a, x_j = b} f(x)$.

If $z = (z_1, \ldots, z_m)$ is some sequence of size m and k is an index in $[m]$ then we will use z_{-k} to denote the subsequence of z of size $m - 1$ obtained by deleting the kth element.

Let $\tilde{\omega}$ be a symmetric fractional polymorphism of Γ of arity $m - 1$. Following the construction in [35], we define graph $(\tilde{\mathbb{G}}, \tilde{E})$ as described in section 5, starting with the distribution $\tilde{\omega}$ where for $s \in \operatorname{supp}(\tilde{\omega})$ mapping $\mathbb{1}^s \in \mathcal{O}^{(m \to m)}$ is defined as follows:

$$\mathbb{1}^s(x) = (s(x_{-1}), \ldots, s(x_{-m})) \qquad \forall x \in D^m.$$

It can be checked that if $\mathbf{g} = (g_1, \ldots, g_m) \in \tilde{\mathbb{G}}$ and $s \in \operatorname{supp}(\tilde{\omega})$ then $\mathbf{g}^s = (s \circ \mathbf{g}_{-1}, \ldots, s \circ \mathbf{g}_{-m})$. It can also be checked that condition (7b) holds for any $f \in \Gamma$: it corresponds to the fractional polymorphism $\tilde{\omega}$ applied to $m - 1$ tuples $x_{-i} \in [\operatorname{dom} f]^{m-1}$.

PROPOSITION 7.6. *There holds $\tilde{\mathbb{G}} \subseteq \mathbb{G}$.*

Proof. We claim that $\mathbb{1}^s \in \Omega$ for any $s \in \operatorname{supp}(\tilde{\omega})$. Indeed, for a permutation π of $[m]$ and $x \in D^m$ we can write

$$\mathbb{1}^s(x^\pi) = \left(s(x^\pi_{-1}), \ldots, s(x^\pi_{-m})\right) = \left(s(x_{-\pi(1)}), \ldots, s(x_{-\pi(m)})\right) = (\mathbb{1}^s)^\pi(x),$$

where the second equality uses that s is symmetric. Since each $\mathbf{g} \in \tilde{\mathbb{G}}$ has the form $\mathbf{g} = \mathbb{1}^{s_k} \circ \cdots \circ \mathbb{1}^{s_1}$ for some $s_1, \ldots, s_k \in \operatorname{supp}(\tilde{\omega})$ and Ω is closed under composition by Proposition 6.1, we get $\tilde{\mathbb{G}} \subseteq \Omega$.

Applying Theorem 5.3 (with $\tilde{\mathbb{G}}$ as both $\mathbb{G}$ and $\widehat{\mathbb{G}}$), we obtain a generalized fractional polymorphism $\tilde{\rho}$ with $\operatorname{supp}(\tilde{\rho}) = \tilde{\mathbb{G}} \subseteq \Omega$. By maximality of ω we get the desired $\tilde{\mathbb{G}} = \operatorname{supp}(\tilde{\rho}) \subseteq \operatorname{supp}(\omega) = \mathbb{G}$. ∎

For each $\mathbf{g} \in \tilde{\mathbb{G}}$ and $k \in [m]$ let us define labeling $x^{[\mathbf{g}k]} \in D^2$ as follows: set $x = \mathbf{g}(\hat{x})$, and then

- If $k = i$, set $x^{[\mathbf{g}k]} = (x_i, x_j)$. We have $x^{[\mathbf{g}i]} \in \arg\min \bar{f}$ since $x \in \mathbb{G}(\hat{x}) = \arg\min f$.
- If $k = j$, set $x^{[\mathbf{g}k]} = (x_j, x_i)$.
- If $k \neq i$ and $k \neq j$, set $x^{[\mathbf{g}k]} = (x_k, x_k)$. We have $x^{[\mathbf{g}k]} \in \operatorname{dom} \bar{f}$ by Lemma 7.4.

Proposition 7.7. *Suppose that $\mathbf{g} \in \tilde{\mathbb{G}}$ and $\mathbf{g}^s = \mathbf{h}$ where $s \in \operatorname{supp}(\tilde{\omega})$ (so that $\mathbf{h} \in \tilde{\mathbb{G}}$). Then*

$$\mathbb{1}^s\left(x^{[\mathbf{g}1]}, \ldots, x^{[\mathbf{g}m]}\right) = \left(x^{[\mathbf{h}1]}, \ldots, x^{[\mathbf{h}m]}\right).$$

Proof. Denote $x = \mathbf{g}(\hat{x})$ and $y = \mathbf{h}(\hat{x})$. We have $y = \mathbb{1}^s(x)$, or $y_k = s(x_{-k})$ for any $k \in [m]$. Also,

$$x^{[\mathbf{g}k]} = \begin{cases} (x_i, x_j) & \text{if } k = i, \\ (x_j, x_i) & \text{if } k = j, \\ (x_k, x_k) & \text{if } k \neq i \text{ and } k \neq j, \end{cases} \qquad x^{[\mathbf{h}k]} = \begin{cases} (y_i, y_j) & \text{if } k = i, \\ (y_j, y_i) & \text{if } k = j, \\ (y_k, y_k) & \text{if } k \neq i \text{ and } k \neq j. \end{cases}$$

It can be checked coordinate-wise (using that s is symmetric) that

$$x^{[\mathbf{h}k]} = s((x^{[\mathbf{g}1]}, \ldots, x^{[\mathbf{g}m]})_{-k})$$

for any $k \in [m]$. This gives the claim. ∎

Denote $\tilde{\mathbb{G}}^* = \bigcup_{\mathbb{H} \in \mathsf{Sinks}(\tilde{\mathbb{G}}, \tilde{E})} \mathbb{H} \subseteq \tilde{\mathbb{G}}$. Let us fix an arbitrary $\tilde{\mathbf{g}} \in \tilde{\mathbb{G}}^*$, and define $\tilde{x} = (x^{[\tilde{\mathbf{g}}1]}, \ldots, x^{[\tilde{\mathbf{g}}m]}) \in [D^2]^m$.

Proposition 7.8. *For any $\mathbf{g} \in \tilde{\mathbb{G}}$ there holds $\mathbf{g} \circ \tilde{\mathbf{g}} \in \tilde{\mathbb{G}}$. Furthermore, $\mathbf{g}(\tilde{x}) = (x^{[(\mathbf{g}\circ\tilde{\mathbf{g}})1]}, \ldots, x^{[(\mathbf{g}\circ\tilde{\mathbf{g}})m]})$.*

Proof. The first claim is by Proposition 5.1(a); let us show the second one. Let $d(\mathbb{1}, \mathbf{g})$ be the shortest distance from $\mathbb{1}$ to $\mathbf{g}$ in the graph $(\tilde{\mathbb{G}}, \tilde{E})$. (By the definition of this graph, we have $0 \leq d(\mathbb{1}, \mathbf{g}) < \infty$ for any $\mathbf{g} \in \tilde{\mathbb{G}}$, and $\mathbb{1} \in \tilde{\mathbb{G}}$.) We will use induction on $d(\mathbb{1}, \mathbf{g})$. The base case $d(\mathbb{1}, \mathbf{g}) = 0$ (i.e. $\mathbf{g} = \mathbb{1}$) holds by construction. Suppose that the claim holds for all mappings $\mathbf{g} \in \tilde{\mathbb{G}}$ with $d(\mathbb{1}, \mathbf{g}) = k \geq 0$, and consider mapping $\mathbf{h} \in \tilde{\mathbb{G}}$ with $d(\mathbb{1}, \mathbf{h}) = k + 1$. There must exist mapping $\mathbf{g} \in \tilde{\mathbb{G}}$ and operation $s \in \operatorname{supp}(\tilde{\omega})$ such that $d(\mathbb{1}, \mathbf{g}) = k$ and $\mathbf{g}^s = \mathbf{h}$. Observe that $(\mathbf{g} \circ \tilde{\mathbf{g}})^s = \mathbb{1}^s \circ \mathbf{g} \circ \tilde{\mathbf{g}} = \mathbf{g}^s \circ \tilde{\mathbf{g}} = \mathbf{h} \circ \tilde{\mathbf{g}}$. We can thus write

$$\mathbf{h}(\tilde{x}) = (\mathbb{1}^s \circ \mathbf{g})(\tilde{x}) = \mathbb{1}^s(\mathbf{g}(\tilde{x})) \overset{(1)}{=} \mathbb{1}^s(x^{[(\mathbf{g}\circ\tilde{\mathbf{g}})1]}, \ldots, x^{[(\mathbf{g}\circ\tilde{\mathbf{g}})m]}) \overset{(2)}{=} (x^{[(\mathbf{h}\circ\tilde{\mathbf{g}})1]}, \ldots, x^{[(\mathbf{h}\circ\tilde{\mathbf{g}})m]}),$$

where (1) holds by the induction hypothesis and (2) is by Proposition 7.7. ∎

Proposition 7.9. *There holds* $\tilde{x} \in Range_2(\tilde{\mathbb{G}}^*) \cap [\mathrm{dom}\,\bar{f}]^m$.

Proof. By Proposition 5.1(b) there exists $\mathbf{g} \in \tilde{\mathbb{G}}^*$ with $\mathbf{g} \circ \tilde{\mathbf{g}} = \tilde{\mathbf{g}}$. Using Proposition 7.8, we can write $\mathbf{g}(\tilde{x}) = (x^{[(\mathbf{g}\circ\tilde{\mathbf{g}})1]}, \ldots, x^{[(\mathbf{g}\circ\tilde{\mathbf{g}})m]}) = (x^{[\tilde{\mathbf{g}}1]}, \ldots, x^{[\tilde{\mathbf{g}}m]}) = \tilde{x}$. This shows that $\tilde{x} \in Range_2(\tilde{\mathbb{G}}^*)$.

Now let us show $x^{[\tilde{\mathbf{g}}k]} \in \mathrm{dom}\,\bar{f}$ for each $k \in [m]$. It suffices to prove it for $k = j$ (for other indices k the claim holds by construction). We have $\tilde{\mathbf{g}} \in \mathbb{H}$ for some strongly connected component $\mathbb{H} \in \mathsf{Sinks}(\tilde{\mathbb{G}}, \tilde{E})$. There is a path from $\tilde{\mathbf{g}}$ to $\tilde{\mathbf{g}}$ in $(\mathbb{H}, E[\mathbb{H}])$, therefore there exists mapping $\mathbf{h} \in \mathbb{H} \subseteq \tilde{\mathbb{G}}^*$ and $s \in \mathrm{supp}(\tilde{\omega})$ with $\mathbf{h}^s = \tilde{\mathbf{g}}$. Define $x = (x^{[\mathbf{h}1]}, \ldots, x^{[\mathbf{h}m]})$, then by Proposition 7.7 we have $\mathbb{1}^s(x) = \tilde{x}$. In particular, $x^{[\tilde{\mathbf{g}}j]} = s(x_{-j})$. Also, we have $x_{-j} \in [\mathrm{dom}\,\bar{f}]^{m-1}$ by construction. Since Γ admits $\tilde{\omega}$ and $s \in \mathrm{supp}(\tilde{\omega})$, we conclude that $x^{[\tilde{\mathbf{g}}j]} \in \mathrm{dom}\,\bar{f}$. ∎

Pick $k \in [m] - \{i, j\}$. By Theorem 5.4(b) we obtain that there exists a probability distribution λ over $\tilde{\mathbb{G}}^*$ such that $\bar{f}_i^\lambda(\tilde{x}) = \bar{f}_k^\lambda(\tilde{x})$. Using Proposition 7.8, we can rewrite this condition as

$$\sum_{\mathbf{g}\in\tilde{\mathbb{G}}^*} \lambda_{\mathbf{g}} \bar{f}\left(x^{[(\mathbf{g}\circ\tilde{\mathbf{g}})i]}\right) = \sum_{\mathbf{g}\in\tilde{\mathbb{G}}^*} \lambda_{\mathbf{g}} \bar{f}\left(x^{[(\mathbf{g}\circ\tilde{\mathbf{g}})k]}\right).$$

Every tuple $x^{[(\mathbf{g}\circ\tilde{\mathbf{g}})i]}$ on the left-hand side belongs to $\arg\min \bar{f}$. Therefore, every tuple $x^{[(\mathbf{g}\circ\tilde{\mathbf{g}})k]}$ on the right-hand side corresponding to mapping $\mathbf{g} \in \tilde{\mathbb{G}}^*$ with $\lambda_{\mathbf{g}} > 0$ also belongs to $\arg\min \bar{f}$.

We proved that there exists $x \in \arg\min f$ with $x_i = x_j$.

Appendix A. Proofs for section 5.

In this section we prove the properties of graph $(\mathbb{G}, E)$ stated in section 5.

A.1. Proof of Proposition 5.1.

Part (a). We have $\mathbf{g} = \mathbb{1}^{s_1\cdots s_k}$ and $\mathbf{h} = \mathbb{1}^{s_{k+1}\cdots s_\ell}$ for some $s_1, \ldots, s_\ell \in \mathrm{supp}(\omega)$ and $0 \le k \le \ell$. Therefore, $\mathbf{h} \circ \mathbf{g} = [\mathbb{1}^{s_\ell} \circ \cdots \circ \mathbb{1}^{s_{k+1}}] \circ [\mathbb{1}^{s_k} \circ \cdots \circ \mathbb{1}^{s_1}] = \mathbb{1}^{s_1\cdots s_\ell} \in \mathbb{G}$. Also, $\mathbf{h} \circ \mathbf{g} = \mathbf{g}^{s_{k+1}\cdots s_\ell}$, and so there is path from $\mathbf{g}$ to $\mathbf{h} \circ \mathbf{g}$ in $(\mathbb{G}, E)$. Since no edges leave the strongly connected component $\mathbb{H}$, we obtain that if $\mathbf{g} \in \mathbb{H}$ then $\mathbf{h} \circ \mathbf{g} \in \mathbb{H}$.

Part (b). Pick $\hat{\mathbf{g}} \in \mathbb{H}$. Since $\mathbb{H}'$ is strongly connected, there is a path from $\hat{\mathbf{g}} \circ \mathbf{g}' \in \mathbb{H}'$ to $\mathbf{g}' \in \mathbb{H}'$ in $(\mathbb{G}, E)$, i.e., $\mathbf{g}' = [\hat{\mathbf{g}} \circ \mathbf{g}']^{s_1\cdots s_k} = \mathbf{h} \circ \hat{\mathbf{g}} \circ \mathbf{g}'$ where $\mathbf{h} = \mathbb{1}^{s_1\cdots s_k}$. It can be checked that mapping $\mathbf{g} = \mathbf{h} \circ \hat{\mathbf{g}}$ has the desired properties.

Part (c). By assumption, $x = \mathbf{g}'(y)$ for some $\mathbf{g}' \in \mathbb{H}' \in \mathsf{Sinks}(\mathbb{G}, E)$ and $y \in [D^n]^m$. By part (b) there exists $\mathbf{g} \in \mathbb{H}$ satisfying $\mathbf{g} \circ \mathbf{g}' = \mathbf{g}'$. We get that $\mathbf{g}(x) = \mathbf{g}(\mathbf{g}'(y)) = (\mathbf{g} \circ \mathbf{g}')(y) = \mathbf{g}'(y) = x$.

A.2. Proof of Proposition 5.2. By assumption, we have $\hat{x} = \mathbf{g}^*(y)$ for some $\mathbf{g}^* \in \mathbb{G}^*$, $y \in D^m$ and $x = \mathbf{h}(\hat{x})$ for some $\mathbf{h} \in \mathbb{G}$.

Part (a). We have $x = (\mathbf{h} \circ \mathbf{g}^*)(y)$ with $\mathbf{h} \circ \mathbf{g}^* \in \mathbb{G}^*$; this establishes the claim.

Part (b). Let $\mathbb{H} \in \mathsf{Sinks}(\mathbb{G}, E)$ be the strongly connected component to which $\mathbf{g}^*$ belongs. There exists a path in $(\mathbb{H}, E[\mathbb{H}])$ from $\mathbf{h} \circ \mathbf{g}^* \in \mathbb{H}$ to $\mathbf{g}^* \in \mathbb{H}$, i.e., $\mathbf{g}^* = \mathbb{1}^{s_1\cdots s_k} \circ \mathbf{h} \circ \mathbf{g}^*$ for some $s_1, \ldots, s_k \in \mathrm{supp}(\omega) = \mathbb{G}$. Define $\mathbf{g} = \mathbb{1}^{s_1\cdots s_k} \in \mathbb{G}$, then $\mathbf{g}^* = \mathbf{g} \circ \mathbf{h} \circ \mathbf{g}^*$. We have $\hat{x} = \mathbf{g}^*(y) = (\mathbf{g} \circ \mathbf{h} \circ \mathbf{g}^*)(y) = (\mathbf{g} \circ \mathbf{h})(\hat{x}) = \mathbf{g}(x)$, as claimed.

A.3. Proof of Theorem 5.3. First, we make the following observation.

Proposition A.1. *Suppose vector* ρ *is a fractional polymorphism of* Γ *of arity* $m \to m$ *and* $\mathbf{g} \in \mathrm{supp}(\rho)$. *Then the following vector is also a fractional polymorphism of* Γ *of arity* $m \to m$:

$$\rho[\mathbf{g}] \;=\; \rho + \frac{\rho(\mathbf{g})}{2}\left[-\chi_{\mathbf{g}} + \sum_{s\in\omega} \omega(s)\chi_{\mathbf{g}^s}\right]. \tag{21}$$

Proof. Denote the vector in the square brackets as δ. Consider function $f \in \Gamma$ and labeling $x \in [\mathrm{dom}\, f]^m$. Since ρ is a fractional polymorphism of Γ, we have $\mathbf{g}(x) \in [\mathrm{dom}\, f]^m$. We can write

$$\sum_{\mathbf{h}\in\mathrm{supp}(\rho[\mathbf{g}])} \delta(\mathbf{h}) f^m(\mathbf{h}(x)) = -f^m(\mathbf{g}(x)) + \sum_{s\in\mathrm{supp}(\omega)} \omega(s) f^m(\mathbf{g}^s(x)) \leq 0,$$

where the last inequality follows from condition (7a) applied to labelings $\mathbf{g}(x)$. Thus, adding the extra term to ρ in (21) will not violate the fractional polymorphism inequality for any $x \in [\mathrm{dom}\, f]^m$. $\square$

Note that $\mathrm{supp}(\rho[\mathbf{g}]) = \mathrm{supp}(\rho) \cup \{\mathbf{g}^s \mid s \in \mathrm{supp}(\omega)\}$ for $\mathbf{g} \in \mathrm{supp}(\rho)$.

We claim that Γ admits a fractional polymorphism $\widehat{\rho}$ with $\mathrm{supp}(\widehat{\rho}) = \mathbb{G}$. Indeed, we can start with vector $\rho = \chi_{\mathbb{1}}$ and then repeatedly modify it as $\rho \leftarrow \rho[\mathbf{g}]$ for mappings $\mathbf{g} \in \mathrm{supp}(\rho)$ that haven't appeared before; after $|\mathbb{G}| - 1$ steps we get a vector $\widehat{\rho}$ with the claimed property.

Let Ω be the set of fractional polymorphisms ρ of Γ with $\mathrm{supp}(\rho) \subseteq \mathbb{G}$ that satisfy $\rho(\mathbf{g}) \geq \widehat{\rho}(\mathbf{g})$ for all $\mathbf{g} \in \widehat{\mathbb{G}}$. Set Ω is nonempty since it contains $\widehat{\rho}$. Let ρ be a vector in Ω that maximizes $\rho(\widehat{\mathbb{G}}) = \sum_{\mathbf{g}\in\widehat{\mathbb{G}}} \rho(\mathbf{g})$. (This maximum is attained since Ω is a compact subset of $\mathbb{R}^{|\mathbb{G}|}$). We claim that $\mathrm{supp}(\rho) = \widehat{\mathbb{G}}$. Indeed, the inclusion $\widehat{\mathbb{G}} \subseteq \mathrm{supp}(\rho)$ is by construction. Suppose there exists $\mathbf{g} \in \mathrm{supp}(\rho) - \widehat{\mathbb{G}}$. By the condition of Theorem 5.3 there exists a path $\mathbf{g}_0, \ldots, \mathbf{g}_k$ in $(\mathbb{G}, E)$ from $\mathbf{g}_0 = \mathbf{g}$ such that $\mathbf{g}_0, \ldots, \mathbf{g}_{k-1} \in \mathbb{G} - \widehat{\mathbb{G}}$ and $\mathbf{g}_k \in \widehat{\mathbb{G}}$. It can be checked that vector $\rho' = \rho[\mathbf{g}_0] \cdots [\mathbf{g}_{k-1}]$ satisfies $\rho' \in \Omega$, $\rho'(\mathbf{g}) \geq \rho(\mathbf{g})$ for $\mathbf{g} \in \widehat{\mathbb{G}}$, and $\rho'(\mathbf{g}_k) > \rho(\mathbf{g}_k)$. This contradicts the choice of ρ.

A.4. Proof of Theorem 5.4(a). Consider component $\mathbb{H} \in \mathsf{Sinks}(\mathbb{G}, E)$, and denote $\mathbb{H}^* = \arg\min\{f^m(\mathbf{g}(x)) \mid \mathbf{g} \in \mathbb{H}\}$. We claim that $\mathbb{H}^* = \mathbb{H}$. Indeed, consider $\mathbf{g} \in \mathbb{H}^*$. Applying inequality (7a) to labelings $\mathbf{g}(x) \in [\mathrm{dom}\, f]^m$ gives

$$\sum_{s\in\mathrm{supp}(\omega)} \omega(s) f^m(\mathbf{g}^s(x)) \leq f^m(\mathbf{g}(x)) \qquad \forall x \in [\mathrm{dom}\, f]^m. \tag{22}$$

For each $s \in \mathrm{supp}(\omega)$ we have $\mathbf{g}^s \in \mathbb{H}$ and thus $f^m(\mathbf{g}^s(x)) \geq f^m(\mathbf{g}(x))$. This means that $f^m(\mathbf{g}^s(x)) = f^m(\mathbf{g}(x))$. We showed that if $\mathbf{g} \in \mathbb{H}^*$ and $(\mathbf{g}, \mathbf{h}) \in E$ then $\mathbf{h} \in \mathbb{H}^*$. Since $\mathbb{H}$ is a strongly connected component of $(\mathbb{G}, E)$, we conclude that $\mathbb{H} = \mathbb{H}^*$.

We showed that $f^m(\mathbf{g}(x))$ is the same for all $\mathbf{g} \in \mathbb{H}$. By Proposition 5.1(c) there exists $\mathbf{h} \in \mathbb{H}$ with $\mathbf{h}(x) = x$, and therefore $f^m(\mathbf{g}(x)) = f^m(\mathbf{h}(x)) = f^m(x)$ for all $\mathbf{g} \in \mathbb{H}$. Since this holds for any $\mathbb{H} \in \mathsf{Sinks}(\mathbb{G}, E)$, the claim follows.

A.5. Proof of Theorem 5.4(b). We mainly follow an argument from [49] (although without using the language of Markov chains, relying on the Farkas lemma instead, as in [35]).

Let $(\mathbb{G}^*, E')$ be the subgraph of $(\mathbb{G}, E)$ induced by $\mathbb{G}^*$. For an edge $(\mathbf{g}, \mathbf{h}) \in E'$, define positive weight $w(\mathbf{g}, \mathbf{h}) = \sum_{s\in\mathrm{supp}(\omega):\mathbf{g}^s=\mathbf{h}} \omega(s)$. Note that we have $\sum_{\mathbf{h}:(\mathbf{g},\mathbf{h})\in E'} w(\mathbf{g}, \mathbf{h}) = 1$ for all $\mathbf{g} \in \mathbb{G}^*$.

We claim that there exists vector $\lambda \in \mathbb{R}^{\mathbb{G}^*}_{\geq 0}$ that satisfies

$$\sum_{\mathbf{g}:(\mathbf{g},\mathbf{h})\in E'} w(\mathbf{g}, \mathbf{h})\lambda_{\mathbf{g}} - \lambda_{\mathbf{h}} = 0 \qquad \forall \mathbf{h} \in \mathbb{G}^*, \tag{23a}$$

$$\sum_{\mathbf{g}\in\mathbb{G}^*} \lambda_{\mathbf{g}} = 1. \tag{23b}$$

Indeed, suppose system (23) does not have a solution. By Farkas's lemma (see Lemma 6.4), there exists a vector $y \in \mathbb{R}^{\mathbb{G}^*}$ and a scalar $z \in \mathbb{R}$ such that

$$z - y_{\mathbf{g}} + \sum_{\mathbf{h}:(\mathbf{g},\mathbf{h})\in E'} w(\mathbf{g},\mathbf{h}) y_{\mathbf{h}} \geq 0 \qquad \forall \mathbf{g} \in \mathbb{G}^*, \tag{24a}$$

$$z < 0. \tag{24b}$$

Consider $\mathbf{g} \in \mathbb{G}^*$ with the maximum value of $y_{\mathbf{g}}$. We have

$$0 \leq z - y_{\mathbf{g}} + \sum_{\mathbf{h}:(\mathbf{g},\mathbf{h})\in E'} w(\mathbf{g},\mathbf{h}) y_{\mathbf{h}} \leq z - y_{\mathbf{g}} + \sum_{\mathbf{h}:(\mathbf{g},\mathbf{h})\in E'} w(\mathbf{g},\mathbf{h}) y_{\mathbf{g}} = z - y_{\mathbf{g}} + y_{\mathbf{g}} = z.$$

This contradicts (24b), and thus proves that vector $\lambda \geq 0$ satisfying (23) exists. Next, we will show that this vector satisfies the property of Theorem 5.4(b).

Let us rewrite condition (7b) as follows:

$$\sum_{s\in\operatorname{supp}(\omega)} \omega(s) f(x^{\mathbf{g}^s i}) \leq \frac{1}{m-1} \sum_{j\in[m]-\{i\}} f(x^{\mathbf{g}j}) \qquad \forall \mathbf{g} \in \mathbb{G}^*, i \in [m]. \tag{25}$$

Multiplying this inequality by $\lambda_{\mathbf{g}}$ and summing over $\mathbf{g} \in \mathbb{G}^*$ (for a fixed $i \in [m]$) gives

$$\sum_{\mathbf{g}\in\mathbb{G}^*} \sum_{\mathbf{h}:(\mathbf{g},\mathbf{h})\in E'} w(\mathbf{g},\mathbf{h}) \lambda_{\mathbf{g}} f(x^{\mathbf{h}i}) \leq \frac{1}{m-1} \sum_{\mathbf{g}\in\mathbb{G}^*} \lambda_{\mathbf{g}} \sum_{j\in[m]-\{i\}} f(x^{\mathbf{g}j}) \qquad \forall i \in [m]. \tag{26}$$

Rearranging terms gives

$$\sum_{\mathbf{h}\in\mathbb{G}^*} \left[\sum_{\mathbf{g}:(\mathbf{g},\mathbf{h})\in E'} w(\mathbf{g},\mathbf{h}) \lambda_{\mathbf{g}} \right] f(x^{\mathbf{h}i}) \leq \frac{1}{m-1} \sum_{j\in[m]-\{i\}} \sum_{\mathbf{g}\in\mathbb{G}^*} \lambda_{\mathbf{g}} f(x^{\mathbf{g}j}) \qquad \forall i \in [m]. \tag{27}$$

By (23a) the expression in the square brackets equals $\lambda_{\mathbf{h}}$, and therefore (27) can be rewritten as

$$f_i^\lambda(x) \leq \frac{1}{m-1} \sum_{j\in[m]-\{i\}} f_j^\lambda(x) \qquad \forall i \in [m]. \tag{28}$$

Consider index $i \in [m]$ with the maximum value of $f_i^\lambda(x)$. We have $f_i^\lambda(x) \geq f_j^\lambda(x)$ for all $j \in [m] - \{i\}$, which together with (28) gives $f_i^\lambda(x) = f_j^\lambda(x)$ for all $j \in [m] - \{i\}$, as claimed.

REFERENCES

[1] L. Barto, *The dichotomy for conservative constraint satisfaction problems revisited*, in Proceedings of the 26th IEEE Symposium on Logic in Computer Science (LICS'11), IEEE Computer Society, Washington, 2011, pp. 301–310, doi:10.1109/LICS.2011.25.

[2] L. Barto, *The collapse of the bounded width hierarchy*, J. Log. Comput., 26 (2016), pp. 923–943, doi:10.1093/logcom/exu070.

[3] L. Barto and M. Kozik, *Absorbing subalgebras, cyclic terms and the constraint satisfaction problem*, Log. Methods Comput. Sci., 8 (2012), pp. 1–26.

[4] L. Barto and M. Kozik, *Constraint satisfaction problems solvable by local consistency methods*, J. ACM, 61 (2014), p. Article 3.

[5] L. Barto and M. Kozik, *Robustly solvable constraint satisfaction problems*, SIAM J. Comput., 45 (2016), pp. 1646–1669, doi:10.1137/130915479.

[6] L. Barto, M. Kozik, and T. Niven, *The CSP dichotomy holds for digraphs with no sources and no sinks (a positive answer to a conjecture of Bang-Jensen and Hell)*, SIAM J. Comput., 38 (2009), pp. 1782–1802, doi:10.1137/070708093.

[7] Y. Boykov, O. Veksler, and R. Zabih, *Markov random fields with efficient approximations*, in Computer Vision and Pattern Recognition, IEEE Computer Society, Washington, 1998, pp. 648–655.

[8] J. Brown-Cohen and P. Raghavendra, *Correlation Decay and Tractability of CSPs*, in 43rd International Colloquium on Automata, Languages, and Programming (ICALP 2016), Leibniz International Proceedings in Informatics (LIPIcs) 55, Schloss Dagstuhl, Wadern, Germany, pp. 79:1–79:13, doi:10.4230/LIPIcs.ICALP.2016.79.

[9] A. Bulatov, *A dichotomy theorem for constraint satisfaction problems on a 3-element set*, J. ACM, 53 (2006), pp. 66–120, doi:10.1145/1120582.1120584.

[10] A. Bulatov, A. Krokhin, and P. Jeavons, *Classifying the complexity of constraints using finite algebras*, SIAM J. Comput., 34 (2005), pp. 720–742, doi:10.1137/S0097539700376676.

[11] A. A. Bulatov, *Complexity of conservative constraint satisfaction problems*, ACM Trans. Comput. Log, 12 (2011), article 24, doi:10.1145/1970398.1970400.

[12] S. O. Chan, J. R. Lee, P. Raghavendra, and D. Steurer, *Approximate constraint satisfaction requires large LP relaxations*, J. ACM, 63 (2016), pp. 34:1–34:22, doi:10.1145/2811255.

[13] D. Cohen, M. Cooper, P. Creed, P. Jeavons, and S. Živný, *An algebraic theory of complexity for discrete optimisation*, SIAM J. Comput., 42 (2013), pp. 1915–1939.

[14] D. Cohen, M. Cooper, and P. Jeavons, *An algebraic characterisation of complexity for valued constraints*, in Principles and Practice of Constraint Programming - CP, 2006, Lect. Notes Comput. Sci. 4204, Springer, Berlin, 2006, pp. 107–121.

[15] D. Cohen, M. Cooper, P. Jeavons, and A. Krokhin, *Supermodular functions and the complexity of Max CSP*, Discrete, 149 (2005), pp. 53–72.

[16] D. Cohen and P. Jeavons, *The complexity of constraint languages*, in Handbook of Constraint Programming, F. Rossi, P. van Beek, and T. Walsh, eds., Elsevier, New York 2006, Chap. 8.

[17] D. A. Cohen, M. C. Cooper, P. G. Jeavons, and A. A. Krokhin, *The complexity of soft constraint satisfaction*, Artif. Intell., 170 (2006), pp. 983–1016, doi:10.1016/j.artint.2006.04.002.

[18] Y. Crama and P. L. Hammer, *Boolean Functions - Theory, Algorithms, and Applications*, Cambridge University Press, Cambridge, 2011.

[19] N. Creignou, S. Khanna, and M. Sudan, *Complexity Classifications of Boolean Constraint Satisfaction Problems*, SIAM Monogr. Discrete Math. Appl. 7, SIAM, Philadelphia, 2001.

[20] N. Creignou, P. Kolaitis, and H. Vollmer, eds., *Complexity of Constraints*, Lect. Notes Comput. Sci. 5250, Springer, Berlin, 2008.

[21] V. Deineko, P. Jonsson, M. Klasson, and A. Krokhin, *The approximability of Max CSP with fixed-value constraints*, J. ACM, 55 (2008), article 16.

[22] A. Ene, J. Vondrák, and Y. Wu, *Local distribution and the symmetry gap: Approximability of multiway partitioning problems*, in ACM-SIAM Symposium on Discrete Algorithms, SODA 2013, pp. 306–325. Update at arXiv1503.03905.

[23] T. Feder and M. Y. Vardi, *The computational structure of monotone monadic SNP and constraint satisfaction: A study through datalog and group theory*, SIAM J. Comput., 28 (1998), pp. 57–104, doi:10.1137/S0097539794266766.

[24] P. Fulla and S. Živný, *A Galois connection for weighted (relational) clones of infinite size*, ACM Trans. Comput. Theory, 8 (2016), pp. 9:1–9:21, doi:10.1145/2898438.

[25] G. Gottlob, G. Greco, and F. Scarcello, *Tractable optimization problems through hypergraph-based structural restrictions*, in Proceedings of ICALP'09, 2009, pp. 16–30.

[26] J. Håstad, *Every 2-CSP allows nontrivial approximation*, Comput. Complexity, 17 (2008), pp. 549–566.

[27] P. Idziak, P. Markovic, R. McKenzie, M. Valeriote, and R. Willard, *Tractability and learnability arising from algebras with few subpowers*, SIAM J. Comput., 39 (2010), pp. 3023–3037.

[28] P. G. Jeavons, *On the algebraic structure of combinatorial problems*, Theor. Comput. Sci., 200 (1998), pp. 185–204, doi:10.1016/S0304-3975(97)00230-2.

[29] P. G. Jeavons, D. A. Cohen, and M. Gyssens, *Closure properties of constraints*, J. ACM, 44 (1997), pp. 527–548, doi:10.1145/263867.263489.

[30] P. Jonsson, M. Klasson, and A. Krokhin, *The approximability of three-valued Max CSP*, SIAM J. Comput., 35 (2006), pp. 1329–1349.

[31] P. Jonsson, F. Kuivinen, and J. Thapper, *Min CSP on four elements: Moving beyond submodularity*, in Principles and Practice of Constraint Programming - CP 2011, Lect. Notes Comput. Sci. 6876, Springer, Berlin, 2011, pp. 438–453.

[32] P. Jonsson and G. Nordh, *Introduction to the Maximum Solution problem*, in Complexity of Constraints, Lect. Notes Comput. Sci. 5250, Springer, Berlin, 2008, pp. 255–282.
[33] P. Kolaitis and M. Vardi, *Conjunctive-query containment and constraint satisfaction*, J. Comput. Syst. Sci., 61 (2000), pp. 302–332.
[34] V. Kolmogorov, M. Rolínek, and R. Takhanov, *Effectiveness of Structural Restrictions for Hybrid CSPs*, in Proceedings of the 26th International Symposium on Algorithms and Computation (ISAAC), Springer, Berlin, 2015, pp. 566–577.
[35] V. Kolmogorov, J. Thapper, and S. Živný, *The power of linear programming for general-valued CSPs*, SIAM J. Comput., 44 (2015), pp. 1–36.
[36] V. Kolmogorov and S. Živný, *The complexity of conservative valued CSPs*, J. ACM, 60 (2013), article 10.
[37] M. Kozik and J. Ochremiak, *Algebraic properties of valued constraint satisfaction problem*, in Automata, Languages, and Programming - 42nd International Colloquium, ICALP 2015, Kyoto, Japan, July 6–10, 2015, Proceedings, Part I, Springer, Berlin, 2015, pp. 846–858.
[38] M. Kozik and J. Ochremiak, *Algebraic properties of valued constraint satisfaction problem*, arXiv1403.0476, 2015.
[39] A. Krokhin and S. Živný, *The complexity of valued CSPs*, in The Constraint Satisfaction Problem: Complexity and Approximability, A. Krokhin and S. Zivny, eds., Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, Dagstuhl, Germany, 2017, pp. 233–266.
[40] S. L. Lauritzen, *Graphical Models*, Oxford University Press, Oxford, 1996.
[41] D. Marx, *Tractable hypergraph properties for constraint satisfaction and conjunctive queries*, J. ACM, 60 (2013), article 42.
[42] U. Montanari, *Networks of Constraints: Fundamental properties and applications to picture processing*, Inf. Sci., 7 (1974), pp. 95–132, doi:10.1016/0020-0255(74)90008-5.
[43] P. Raghavendra, *Optimal algorithms and inapproximability results for every CSP?*, in Proceedings of the 40th ACM Symposium on Theory of Computing (STOC'08), ACM, New York, 2008, pp. 245–254.
[44] F. Rossi, P. van Beek, and T. Walsh, eds., *Handbook of Constraint Programming*, Elsevier, New York, 2006.
[45] T. J. Schaefer, *The Complexity of Satisfiability Problems*, in Proceedings of the 10th Annual ACM Symposium on Theory of Computing (STOC'78), ACM, New York, 1978, pp. 216–226, doi:10.1145/800133.804350.
[46] A. Schrijver, *Theory of Linear and Integer Programming*, Wiley, New York, 1986.
[47] R. Takhanov, *A dichotomy theorem for the general minimum cost homomorphism problem*, in 27th International Symposium on Theoretical Aspects of Computer Science, STACS'10, 2010, pp. 657–668.
[48] J. Thapper and S. Živný, *Sherali-Adams relaxations for valued CSPs*, in Automata, Languages, and Programming - 42nd International Colloquium, ICALP 2015, Kyoto, Japan, July 6-10, 2015, Proceedings, Part I, 2015, Springer, Berlin, pp. 1058–1069.
[49] J. Thapper and S. Živný, *The complexity of finite-valued CSPs*, in Proceedings of the 45th ACM Symposium on the Theory of Computing (STOC'13), ACM, New York, 2013, pp. 695–704, doi:10.1145/2488608.2488697.
[50] J. Thapper and S. Živný, *The power of Sherali-Adams relaxations for general-valued CSPs*, SIAM J. Comput., to appear.
[51] H. Uppman, *The Complexity of Three-Element Min-Sol and Conservative Min-Cost-Hom*, in Proceedings of the 40th International Colloquium on Automata, Languages, and Programming (ICALP'13), Lect. Notes Comput. Sci. 7965, Springer, Berlin, 2013, pp. 804–815.
[52] H. Uppman, *Computational complexity of the extended minimum cost homomorphism problem on three-element domains*, in 31st International Symposium on Theoretical Aspects of Computer Science (STACS 2014), Schloss Dagstuhl, Wadern, Germany, 2014, pp. 651–662.
[53] M. Wainwright and M. Jordan, *Graphical models, exponential families, and variational inferences*, Found. Trends Mach. Learn., 1 (2008), pp. 1–305.