On the Valued Constraint Satisfaction Problem

Peter Fulla
Balliol College
University of Oxford

DPhil in Computer Science

Abstract

On the Valued Constraint Satisfaction Problem
Peter Fulla
Balliol College
DPhil in Computer Science
Trinity 2018

The Valued Constraint Satisfaction Problem (VCSP) is a framework which captures many natural decision and optimisation problems. An instance of the VCSP consists of a set of variables, which are to be assigned labels from a finite domain, and a collection of local constraints, each specified by a weighted relation mapping labellings of the variables in the constraint's scope to values. The objective is to minimise the total value from all constraints. The VCSP is commonly parameterised by a language, i.e. a set of weighted relations that are available for use in the constraints. Languages are classified according to the computational complexity of the VCSP as tractable, for which the problem can be solved in polynomial time, and intractable, for which the problem is NP-hard. The recently proved VCSP dichotomy theorem established a classification of all languages into these two categories. Additionally, various structural restrictions can be imposed to limit the set of admissible instances further, thus potentially changing the complexity of the VCSP.

Our first contribution relates to the algebraic approach to the VCSP, which proved instrumental in recent advances in the field. We generalise the Galois connection between weighted relational clones and weighted clones so that it applies to infinite sets as well. Second, we study a structural restriction requiring that the incidence graph be planar. In this setting, we establish a complexity classification of conservative languages (i.e. languages containing all $\{0, 1\}$-valued unary weighted relations) and a necessary tractability condition for Boolean languages (i.e. languages over a two-element domain). Third, we study the surjective variant of the VCSP, in which labellings are required to assign every domain element to at least one variable. We establish a complexity classification of Boolean languages, which encompasses a new tractable class of problems.

Declaration

Parts of this thesis have been published in academic journals and conference proceedings. Namely, the results in Chapter 3 were published in [49, 50], most of Chapter 4 in [51, 52], and most of Chapter 5 in [53, 48].

Contents

Chapter 1

Introduction

The Valued Constraint Satisfaction Problem (VCSP) is a discrete optimisation problem which enables expressing many natural decision and optimisation problems as its special cases [77, 31, 91]. An instance of the VCSP consists of a finite set of variables on which there are imposed constraints. Each variable is to be assigned a label from a finite set called the domain;[1] such an assignment is called a labelling. A constraint is represented by a weighted relation which maps labellings of the variables in the constraint's scope to values (rational numbers or positive infinity). Because the scope covers only a (typically small) subset of all the variables, constraints can be thought of as local. The goal is to find a labelling that minimises the sum of the values from all the constraints.

For example, consider how the maximum cut problem can be naturally expressed in the VCSP framework. Let the domain consist of labels 0 and 1 representing the two parts of a cut. We associate each vertex of the graph with a variable; the label assigned to a variable indicates which of the two parts the corresponding vertex belongs to. In order to determine the total weight of the cut, we impose on every adjacent pair of vertices u, v a binary constraint $\gamma_{\neq}(u, v)$ that, based on the labels of the vertices, evaluates whether the edge is cut or not. Weighted relation $\gamma_{\neq}$ assigns a larger value to uncut edges (see Example 2.10), and hence minimising the total value corresponds to maximising the cut.

The minimum (s, t)-cut problem can be expressed in a similar way. This time, we impose on every adjacent pair of vertices u, v a binary constraint $\gamma_{=}(u, v)$, where weighted relation $\gamma_{=}$ assigns a larger value to cut edges (see Example 2.11). To ensure that vertices s and t belong to different parts, we impose on them unary constraints $\rho_0(s)$ and $\rho_1(t)$, where ρ_a maps label a to

---

[1] The VCSP over infinite domains has been studied as well, see [8].

value 0 and the other label to ∞.

The VCSP captures many classic decision problems as well. Consider the problem of determining whether the chromatic number of a given graph is at most k. Let the domain consist of k distinct labels representing k colours. Again, we associate each vertex of the graph with a variable. For every edge of the graph, we impose a binary constraint on the corresponding pair of variables; the constraint assigns value ∞ if the endpoints have the same colour and value 0 otherwise (see Example 2.9). Hence, proper vertex colourings obtain a total value of 0, while other labellings are assigned ∞. Instances such as this one, in which the minimisation objective reduces to finding a labelling with a finite value, form an important special case of the VCSP called the Constraint Satisfaction Problem (CSP) [31, 91].

The VCSP is an NP-hard problem[2] in its full generality, but restricting the set of admissible instances may result in tractable special cases. The main types of restrictions that have been studied are language restrictions [15, 28, 25, 75, 100, 70], which limit the choice of weighted relations in constraints; structural restrictions [58, 44, 57, 82, 21], which limit how constraints interact; and hybrid restrictions [29], which are a combination of the first two types.

The language-restricted VCSP is parameterised by a set of weighted relations called a (constraint) language that specifies which weighted relations may appear in the constraints of admissible instances. For example, the language consisting of all $\{0, \infty\}$-valued weighted relations restricts the VCSP to the CSP, and language $\{\gamma_{\neq}\}$ encodes the maximum cut problem. These are examples of intractable languages, as the restricted VCSP is still NP-hard. On the other hand, there exist languages that limit the expressive power of the VCSP sufficiently so that it becomes possible to find an optimal solution in polynomial time; these are called tractable languages. For example, any submodular language (such as $\{\gamma_=, \rho_0, \rho_1\}$, which we previously used to express the minimum (s, t)-cut problem) is tractable (see Example 2.29).

Identifying tractable and intractable languages has been at the centre of research on the CSP for many years. Schaefer [92] classified languages over two-element (also called Boolean) domains (see Theorem 2.32), while Hell and Nešetřil [60] classified languages consisting of a single binary symmetric relation. Both results are in the form of a dichotomy, where every language considered is either tractable or intractable. By Ladner's theorem [78], there exist NP-intermediate problems (assuming P $\neq$ NP); however, it seemed plausible that none of them can be encoded by a constraint language. Feder and Vardi [46] formulated this as a conjecture stating that, in general, every

---

[2] The decision version, which asks whether there exists a solution with at most a given value, is NP-complete.

language is either tractable or intractable. Moreover, they established that the CSP is, in a certain sense, the largest natural subclass of NP that may exhibit such a dichotomy. The conjecture was gaining support as further dichotomy classifications in special cases were established: languages over three-element domains [13], languages consisting of a single binary relation corresponding to a digraph without sources and sinks [4], and languages containing all unary relations [17, 2]. This progress was driven by the algebraic approach introduced in [15], which enables characterising languages by their symmetries called polymorphisms [5]. In particular, rather than classifying all languages, it is sufficient to consider only relational clones, i.e. sets of relations closed under certain operations that preserve language tractability. Conversely, the polymorphisms of a language always form a so-called clone. Because of a Galois connection (a type of a one-to-one correspondence) between relational clones and clones, the focus may shift to classifying clones of polymorphisms, for which powerful tools from the field of universal algebra can be employed. Apart from establishing complexity classifications, the algebraic approach was also instrumental in identifying precise boundaries of applicability of efficient algorithms such as local consistency methods [16, 3] (see Theorem 2.18) and a generalisation of Gaussian elimination [18, 33, 63, 6]. Bulatov, Krokhin, and Jeavons [15] strengthened the conjecture of Feder and Vardi by proposing an algebraic condition characterising tractable and intractable languages, and showed that languages violating this condition are intractable. Finally, the tractability part of this so-called algebraic CSP dichotomy conjecture was independently proved by Bulatov [14] and Zhuk [107] (see Theorem 2.17).

Some of the earliest results on the optimisation front concerned the Min-CSP (or Max-CSP[3]), in which the goal is to minimise (maximise) the number of unsatisfied (satisfied) constraints [26, 66, 37, 67]. Equivalently, these problems correspond to the VCSP over $\{0, 1\}$-valued languages. The algebraic approach was extended from the CSP to the (general-valued) VCSP [28, 25]. Analogously to the CSP, this allowed characterising languages by their symmetries called weighted polymorphisms, as a Galois connection between weighted relational clones and weighted clones was established [25]. However, this correspondence is limited to weighted relational clones that can be generated by a finite set. We present a new Galois connection that applies to all weighted relational clones in Chapter 3. The algebraic approach for the VCSP led to a number of complexity classifications for special classes of general-valued languages: languages over two-element (Boolean) domains [28] (see Theorem 2.30), languages containing all $\{0, 1\}$-valued unary weighted

---

[3] When classifying languages with respect to exact solvability, the Min-CSP and Max-CSP are interchangeable problems. However, this is not true for approximate solvability [31].

relations [73], and finite-valued languages [100]. Moreover, a classification of languages with respect to their solvability by a particular efficient algorithms was established for the basic linear programming relaxation [72] (see Theorem 2.25) and the Sherali-Adams relaxation [101] (see Theorem 2.26). As with the CSP, the complexity classifications in all known special cases had the form of a dichotomy between tractable and intractable languages. Kozik and Ochremiak [75] proposed an analogue of the algebraic CSP dichotomy conjecture, and showed that languages that violate their algebraic condition are intractable. Kolmogorov, Krokhin, and Rolínek [70] proved that this condition is sufficient for tractability under the assumption that the underlying CSP is tractable. Because this assumption follows from the algebraic CSP dichotomy conjecture, resolving the conjecture for the CSP immediately established a dichotomy classification for the VCSP as well (see Theorem 2.24).

Orthogonal to restricting the VCSP by a constraint language are various structural restrictions, which limit the set of admissible instances by specifying requirements for the interaction of constraints [58, 57, 82, 21]. This may reduce the computational complexity of the VCSP; in the case of hybrid restrictions (a combination of language and structural restrictions), an intractable language may become tractable under a structural restriction. Although some progress has been made on extending the algebraic approach to the VCSP with hybrid restrictions [71, 97], many of its tools (e.g., a Galois connection) still remain unavailable.

We focus on the planar VCSP, which requires that the incidence graph of instances[4] be planar. In Chapter 4, we show that the complexity of any language containing all $\{0, 1\}$-valued unary weighted relations is not affected by the planarity restriction; in other words, there are no new tractable cases in the planar setting compared to the general setting [73]. As another line of research, we consider Boolean languages. In the case of the CSP (i.e., $\{0, \infty\}$-valued languages), Dvořák and Kupec identified a necessary condition for an intractable language to become tractable under the planarity restriction. This condition relates to a combinatorial structure called even delta-matroids [10, 22, 39], generalising matroids. Kazda, Kolmogorov, and Rolínek [69] proved the sufficiency of the condition by designing an efficient algorithm for a special case of the VCSP to which the original problem can be reduced. Interestingly, their algorithm applies to a wider class of problems, as it does not rely on the planarity of instances; that is required only for the reduction. In Chapter 4,

---

[4] In the incidence graph of an instance, two parts of vertices represent the variables and constraints respectively, and the edges connect variables with the constraints in whose scopes they appear.

we extend the condition of Dvořák and Kupec to the planar VCSP (i.e., general-valued languages) and prove its necessity. Our condition relates to valuated delta-matroids [40], which generalise weighted matroids.

To model problems arising in practice, it is sometimes convenient or even necessary to impose constraints that are not local (i.e., the number of involved variables cannot be bounded by a constant). In such cases, the VCSP alone is not an appropriate framework, and hence its extensions with various kinds of global constraints[5] may be adopted instead. For example, a global cardinality constraint specifies the cardinality for each label from the domain, i.e., the number of variables to be assigned that label. A more general variant specifies a set of allowed cardinalities for each label; in fact, the two variants are polynomial-time equivalent, as there are only polynomially many cardinality vectors for a fixed size of the domain. In contrast to hybrid restrictions, extending the VCSP with a (general) global cardinality constraint may increase the computational complexity of a language. Bulatov and Marx [20] classified all $\{0, \infty\}$-valued languages with respect to this extended framework as either tractable or intractable. Although their classification can be expressed in terms of polymorphisms, the proof relies largely on combinatorial tools, as the algebraic approach has not yet been fully extended to the setting with global constraints.

The surjectivity constraint, an interesting special case of a general global cardinality constraint, requires that each label be assigned to at least one variable. Consider, for example, how to express the minimum cut problem in the VCSP framework. As in the case of the minimum (s, t)-cut problem mentioned at the beginning of this introduction, we use two labels representing the two parts of a cut, associate each vertex with a variable, and impose a binary (i.e., local) constraint on every edge to determine the total weight of a cut. However, such an instance admits a trivial cut with all the vertices in one part as an optimal solution. The surjectivity constraint prevents this issue, and hence it allows expressing the minimum cut problem in a natural way. Complexity classifications of Boolean languages with respect to the surjective VCSP were established in some special cases: Creignou and Hébrard [30] classified $\{0, \infty\}$-valued languages, and Uppman [102] classified $\{0, 1\}$-valued languages. In Chapter 5, we extend both of these results by establishing a dichotomy classification of all Boolean languages. We identify a class of languages that are trivially tractable in the general setting but require a different (non-trivial) approach under the surjectivity constraint. In particular, we establish a reduction to a generalisation of the minimum cut problem, for which we develop a polynomial-time algorithm.

---

[5] See `http://sofdem.github.io/gccat/` for a catalog of global constraints.

Chapter 2

Valued constraint satisfaction problem

In this chapter, we define the terminology used throughout the thesis. Section 2.1 gives a formal definition of the VCSP. Section 2.2 defines basic concepts of the algebraic approach and states the CSP and VCSP dichotomy theorems. Section 2.3 deals specifically with the case of Boolean VCSP.

For more details on the CSP and VCSP, we refer the reader to recent surveys [5, 77].

2.1 Preliminaries

We work in the arithmetic model of computation, i.e., every number is represented in constant space, and basic arithmetic operations take constant time. Let $\overline{\mathbb{Q}} = \mathbb{Q} \cup \{\infty\}$ denote the set of extended rationals. For any $c \in \overline{\mathbb{Q}}$, we define $c \leq \infty$ and $\infty + c = c + \infty = \infty$. If $c \geq 0$, we define $c \cdot \infty = \infty \cdot c = \infty$. We leave the result of multiplying ∞ undefined for $c < 0$.

For any integer $n \geq 1$, let $[n] = \{1, \ldots, n\}$. We denote the standard polynomial-time Turing reduction by $\leq_p$.

Definition 2.1. A *domain* is a finite set D of size at least 2. The elements of a domain are called *labels*.

A *Boolean domain* is a domain D with $|D| = 2$. Conventionally, $D = \{0, 1\}$.

Definition 2.2. Let $r \geq 1$ be an integer. An r-ary *weighted relation* over D is a mapping $\gamma : D^r \to \overline{\mathbb{Q}}$; the *arity* of γ equals $\mathrm{ar}(\gamma) = r$. We denote by $\mathrm{Feas}(\gamma)$ the underlying *feasibility relation* of γ, i.e.

$$\mathrm{Feas}(\gamma) = \{\mathbf{x} \in D^r \mid \gamma(\mathbf{x}) < \infty\} \ . \tag{2.1}$$

We denote by $\operatorname{Opt}(\gamma)$ the relation consisting of the minimal-valued tuples, i.e.

$$\operatorname{Opt}(\gamma) = \{\mathbf{x} \in \operatorname{Feas}(\gamma) \mid \gamma(\mathbf{x}) \leq \gamma(\mathbf{y}) \text{ for every } \mathbf{y} \in D^r\} . \tag{2.2}$$

A weighted relation γ is called *crisp* if $\operatorname{Feas}(\gamma) = \operatorname{Opt}(\gamma)$. In other words, there exists a constant $c \in \mathbb{Q}$ such that $\gamma(\mathbf{x}) = c$ for all $\mathbf{x} \in \operatorname{Feas}(\gamma)$ and $\gamma(\mathbf{x}) = \infty$ for all $\mathbf{x} \in D^r \setminus \operatorname{Feas}(\gamma)$.

Weighted relations that differ only by a constant are considered equivalent, as adding a constant to a weighted relation changes the value of every solution to the VCSP by the same amount. Therefore, a crisp weighted relation γ can be equated with the relation $\operatorname{Feas}(\gamma)$. Conversely, a relation ρ can be seen as a crisp weighted relation γ_c with $\operatorname{Feas}(\gamma_c) = \rho$ and the codomain equal to $\{c, \infty\}$ for some $c \in \mathbb{Q}$. Unless stated otherwise, we choose $c = 0$.

Definition 2.3. We denote by $\rho_=$ the binary equality relation $\{(d, d) \mid d \in D\}$, by $\rho_{\neq}$ the binary disequality relation $D^2 \setminus \rho_=$, and by $\rho_\emptyset$ the unary empty relation $\emptyset$. For any $d \in D$, we denote by ρ_d the unary relation $\{(d)\}$, which is sometimes called a *constant*.

For any relation ρ, we denote by $\operatorname{Soft}(\rho)$ the *soft variant* of ρ defined by $\operatorname{Soft}(\rho)(\mathbf{x}) = 0$ if $\mathbf{x} \in \rho$ and $\operatorname{Soft}(\rho)(\mathbf{x}) = 1$ otherwise.

Definition 2.4. We define a few operations on weighted relations that occur throughout the thesis. Let γ be an r-ary weighted relation.

- *Addition of a constant*: For any $c \in \mathbb{Q}$, $\gamma + c = \gamma'$ such that $\gamma'(\mathbf{x}) = \gamma(\mathbf{x}) + c$.
- *Non-negative scaling*: For any $c \in \mathbb{Q}_{\geq 0}$, $c \cdot \gamma = \gamma'$ such that $\gamma'(\mathbf{x}) = c \cdot \gamma(\mathbf{x})$. Note that $0 \cdot \gamma = \operatorname{Feas}(\gamma)$.
- *Coordinate mapping*: For any arity r' and mapping $f : [r] \to [r']$, $f(\gamma) = \gamma'$ such that $\gamma'(x_1, \ldots, x_{r'}) = \gamma\left(x_{f(1)}, \ldots, x_{f(r)}\right)$.
- *Minimisation*: For any $i \in [r]$, the minimisation of γ at coordinate i results in γ' such that $\gamma'(x_1, \ldots, x_{i-1}, x_{i+1}, \ldots, x_r) = \min_{x_i \in D} \gamma(x_1, \ldots, x_r)$.
- *Pinning*: For any $a \in D$ and $i \in [r]$, the pinning of γ to label a at coordinate i results in γ' such that $\gamma'(x_1, \ldots, x_{i-1}, x_{i+1}, \ldots, x_r) = \gamma(x_1, \ldots, x_{i-1}, a, x_{i+1}, \ldots, x_r)$. A pinning to label a is called an *a-pinning*.
- *Addition*: For any weighted relations γ_1, γ_2 with $\operatorname{ar}(\gamma_1) = \operatorname{ar}(\gamma_2)$, $\gamma_1 + \gamma_2 = \gamma'$ such that $\gamma'(\mathbf{x}) = \gamma_1(\mathbf{x}) + \gamma_2(\mathbf{x})$.

Definition 2.5. A *constraint language* (or simply a *language*) over D is a set Γ of weighted relations over D. We extend operations on weighted relations to languages in the natural way, e.g., $\mathrm{Feas}(\Gamma) = \{\mathrm{Feas}(\gamma) \mid \gamma \in \Gamma\}$.

Definition 2.6. An *instance* $I = (V, D, \phi_I)$ of the VCSP on domain D is given by a finite set of variables $V = \{x_1, \ldots, x_n\}$ and an objective function $\phi_I : D^n \to \overline{\mathbb{Q}}$ expressed as a weighted sum of *constraints* over V, i.e.,

$$\phi_I(x_1, \ldots, x_n) = \sum_{i=1}^{q} w_i \cdot \gamma_i(\mathbf{x}_i)\,, \tag{2.3}$$

where γ_i is a weighted relation, $w_i \in \mathbb{Q}_{\geq 0}$ is the *weight* and $\mathbf{x}_i \in V^{\mathrm{ar}(\gamma_i)}$ the *scope* of the ith constraint. Note that zero weights are allowed.

A *labelling* is an assignment $s : V \to D$. The value of a labelling s equals $\phi_I(s) = \phi_I(s(x_1), \ldots, s(x_n))$. We call s *feasible* if $\phi_I(s) < \infty$, and *optimal* if it is feasible and $\phi_I(s) \leq \phi_I(s')$ for all labellings s'. Given an instance I, the goal is to decide whether a feasible labelling exists and, if so, to find an optimal one (i.e. a labelling that minimises ϕ_I).

Definition 2.7. For any language Γ over D, we denote by VCSP(Γ) a restriction of the VCSP in which only instances with $\gamma_i \in \Gamma$ for all $i \in [q]$ are permitted.

A language Γ is called *tractable* if, for every finite $\Gamma' \subseteq \Gamma$, VCSP(Γ') can be solved in polynomial time. If there exists a finite $\Gamma' \subseteq \Gamma$ such that VCSP(Γ') is NP-hard, language Γ is called *intractable.*

Defining tractability in terms of finite subsets ensures that the tractability of a language is independent of whether the weighted relations are represented explicitly (via tables of values) or implicitly (via oracles). A stronger notion of *global tractability* [15] requires that VCSP(Γ) can be solved in polynomial time. As it turns out, these notions are equivalent in the case of the VCSP [15, 75, 70, 14, 107], i.e., every tractable language is globally tractable, and every globally intractable language is intractable. For the surjective VCSP, however, we also consider the stronger notion, as there exist tractable languages that are not globally tractable (see Chapter 5).

If a language Γ is crisp (i.e., it contains only crisp weighted relations), solving VCSP(Γ) reduces to finding a feasible labelling, as any such solution has the same objective value. This subclass of the VCSP is known as the constraint satisfaction problem (CSP).

Example 2.8. Let Γ be a language over $D = \{0, 1\}$ that consists of ternary relations ψ_{a_1,a_2,a_3} defined by

$$\psi_{a_1,a_2,a_3} = D^3 \setminus \{(a_1, a_2, a_3)\} \tag{2.4}$$

for all $a_1, a_2, a_3 \in D$. Then $\mathrm{VCSP}(\Gamma)$ corresponds to the 3-SAT problem. Similarly, $\mathrm{VCSP}(\mathrm{Soft}(\Gamma))$ encodes the MAX-3-SAT problem.

Example 2.9. Problem $\mathrm{VCSP}(\{\rho_{\neq}\})$ corresponds to graph colouring with $|D|$ colours (we impose a constraint $\rho_{\neq}(x, y)$ for every edge (x, y) of the graph). Therefore, $\{\rho_{\neq}\}$ is tractable for $|D| = 2$ and intractable for $|D| \geq 3$ [54].

Example 2.10. Let $D = \{0, 1\}$ and $\gamma_{\neq} = \mathrm{Soft}(\rho_{\neq})$. Similarly as in the previous example, $\mathrm{VCSP}(\{\gamma_{\neq}\})$ corresponds to the problem of minimising the total weight of the edges such that both their endpoints are assigned the same colour, i.e. the maximum cut problem. Language $\{\gamma_{\neq}\}$ is thus intractable [54].

Example 2.11. Let $D = \{0, 1\}$ and $\gamma_{=} = \mathrm{Soft}(\rho_{=})$. Then $\mathrm{VCSP}(\{\rho_0, \rho_1, \gamma_{=}\})$ corresponds to a version of the minimum (s, t)-cut problem in which two *sets* of terminal vertices are given: A constraint of the form $\rho_a(v)$ specifies that v is a terminal vertex that must be assigned to part a of the cut, and a constraint of the form $w \cdot \gamma_{=}(u, v)$ increases the overall value of the solution by w if edge (u, v) is cut. Language $\{\rho_0, \rho_1, \gamma_{=}\}$ is thus tractable [93].

2.2 Algebraic approach

In this section, we formally define basic concepts forming the algebraic approach to classifying constraint languages introduced in [15, 25].

Definition 2.12. For any $k \geq 1$ and $i \in [k]$, let $\pi_i^{(k)} : D^k \to D$ denote the k-ary operation such that $\pi_i^{(k)}(x_1, \dots, x_k) = x_i$ for all $x_1, \dots, x_k \in D$. Operations $\pi_i^{(k)}$ are called *projections*.

For any $k \geq 2$, a k-ary operation $f : D^k \to D$ is called a *weak near-unanimity operation* (WNU) if

$$f(y, x, \dots, x) = f(x, y, x, \dots, x) = \cdots = f(x, \dots, x, y) \tag{2.5}$$

holds for all $x, y \in D$; and it is called *symmetric* if

$$f(x_1, \dots, x_k) = f\left(x_{p(1)}, \dots, x_{p(k)}\right) \tag{2.6}$$

holds for all $x_1, \dots, x_k \in D$ and all permutations $p : [k] \to [k]$.

For any $a \in D$, we define $c_a : D \to D$ to be the *constant unary operation* such that $c_a(x) = a$ for all $x \in D$.

If a total order on D is specified, we denote by min (max) the binary operation which returns the smaller (larger) of its two arguments with respect to that order. Note that min and max are symmetric (and thus also WNU) operations.

A ternary operation $f : D^3 \to D$ is called a *majority* operation if

$$f(x, x, y) = f(x, y, x) = f(y, x, x) = x \tag{2.7}$$

for all $x, y \in D$, and a *minority* operation if

$$f(x, x, y) = f(x, y, x) = f(y, x, x) = y \tag{2.8}$$

for all $x, y \in D$. Note that majority and minority operations are special cases of ternary WNU operations but are not necessarily symmetric.

We denote by $\mathcal{O}_D$ the set of all operations on D, and by $\Pi_D \subseteq \mathcal{O}_D$ the set of all projections on D. For any $C \subseteq \mathcal{O}_D$, we denote by $C^{(k)}$ the set of k-ary operations in C.

Definition 2.13 ([95])**.** The *superposition* of a k-ary operation $f : D^k \to D$ with ℓ-ary operations $g_i : D^\ell \to D$ for $i \in [k]$ is the ℓ-ary operation $f[g_1, \ldots, g_k] : D^\ell \to D$ defined by

$$f[g_1, \ldots, g_k](x_1, \ldots, x_\ell) = f(g_1(x_1, \ldots, x_\ell), \ldots, g_k(x_1, \ldots, x_\ell)) . \tag{2.9}$$

A set of operations $C \subseteq \mathcal{O}_D$ is called a *clone* if $\Pi_D \subseteq C$ and C is closed under superposition.

Observation 2.14. *Sets Π_D and $\mathcal{O}_D$ are clones.*

For any $r \geq 1$ and a k-ary operation $f : D^k \to D$, we extend f to r-tuples over D by applying it componentwise. Namely, for $\mathbf{x}_1, \ldots, \mathbf{x}_k \in D^r$ where $\mathbf{x}_i = (x_{i,1}, \ldots, x_{i,r})$, we define $f(\mathbf{x}_1, \ldots, \mathbf{x}_k) \in D^r$ by

$$f(\mathbf{x}_1, \ldots, \mathbf{x}_k) = (f(x_{1,1}, \ldots, x_{k,1}), \ldots, f(x_{1,r}, \ldots, x_{k,r})) . \tag{2.10}$$

Polymorphisms, which can be thought of as symmetries of a constraint language, form the basis of the algebraic approach for the CSP.

Definition 2.15 ([15])**.** Let γ be a weighted relation on D. A k-ary operation $f : D^k \to D$ is a *polymorphism* of γ (and γ is *invariant* under f) if, for every $\mathbf{x}_1, \ldots, \mathbf{x}_k \in \mathrm{Feas}(\gamma)$, it holds $f(\mathbf{x}_1, \ldots, \mathbf{x}_k) \in \mathrm{Feas}(\gamma)$.

We say that f is a polymorphism of a language Γ if it is a polymorphism of every $\gamma \in \Gamma$. By $\mathrm{Pol}(\Gamma)$ we denote the set of polymorphisms of Γ.

Note that projections are (trivial) polymorphisms of all constraint languages. Moreover, it can be shown that polymorphisms are closed under composition.

Theorem 2.16 ([88])**.** *For any constraint language* Γ*, set* $\mathrm{Pol}(\Gamma)$ *is a clone.*

Polymorphisms of a crisp language characterise its computational complexity. In particular, the CSP dichotomy theorem can be stated in terms of polymorphisms as follows.

Theorem 2.17 ([14, 107])**.** *Let* Γ *be a finite* $\{0, \infty\}$*-valued language. Then* Γ *is tractable if it is invariant under a WNU operation; otherwise,* Γ *is intractable.*

Similarly, polymorphisms can be used to characterise languages that are tractable using a particular algorithms, e.g., so-called *local consistency methods* [35].

Theorem 2.18 ([16, 3, 74])**.** *Let* Γ *be a finite* $\{0, \infty\}$*-valued language. Then* Γ *is solvable by local consistency methods if and only if it is invariant under a* k*-ary WNU operation for all* $k \geq 3$*.*

For general ($\overline{\mathbb{Q}}$-valued) languages, an extension of the notion of a polymorphism that allows to state a full complexity classification is called a *weighted polymorphism* [25]. First, we define an important special case called a *multimorphism.*

Definition 2.19 ([28])**.** Let γ be a weighted relation on D. A list $\langle f_1, \ldots, f_k \rangle$ of k-ary polymorphisms of γ is a k-ary *multimorphism* of γ (and γ is *improved* by $\langle f_1, \ldots, f_k \rangle$) if, for every $\mathbf{x}_1, \ldots, \mathbf{x}_k \in \mathrm{Feas}(\gamma)$, it holds

$$\sum_{i=1}^{k} \gamma\left(\mathbf{x}_i\right) \geq \sum_{i=1}^{k} \gamma\left(f_i\left(\mathbf{x}_1, \ldots, \mathbf{x}_k\right)\right) . \tag{2.11}$$

We say that $\langle f_1, \ldots, f_k \rangle$ is a multimorphism of a language Γ if it is a multimorphism of every $\gamma \in \Gamma$.

Example 2.20 ([28])**.** For any total order on D, languages admitting multimorphism $\langle \min, \max \rangle$ are tractable. In the case of a Boolean domain, this multimorphism characterises submodular functions (see Example 2.29).

Multimorphisms are in some cases sufficient to capture certain properties of languages, e.g., submodularity or even tractability of Boolean languages (see Theorem 2.30). For a general classification, however, we need to define weightings and weighted polymorphisms.

Definition 2.21 ([25])**.** A k-ary *weighting* of a clone C is a function $\omega : C^{(k)} \to \mathbb{Q}$ such that $\omega(f) < 0$ only if f is a projection and

$$\sum_{f \in C^{(k)}} \omega(f) = 0\,. \tag{2.12}$$

We define the *support* of ω by

$$\operatorname{supp}(\omega) = \Pi_D^{(k)} \cup \left\{ f \in C^{(k)} \;\middle|\; \omega(f) > 0 \right\}\,. \tag{2.13}$$

For any set of weightings Ω, we define $\operatorname{supp}(\Omega) = \Pi_D \cup \bigcup_{\omega \in \Omega} \operatorname{supp}(\omega)$.

When specifying a weighting, we often write it as a weighted sum of operations excluding terms with weight 0. For example, $\omega = 4 \cdot f - \pi_1^{(2)} - 3 \cdot \pi_2^{(2)}$ is the binary weighting such that $\omega(f) = 4$, $\omega(\pi_1^{(2)}) = -1$, $\omega(\pi_2^{(2)}) = -3$, and $\omega(g) = 0$ for $g \in C^{(2)} \setminus \{f, \pi_1^{(2)}, \pi_2^{(2)}\}$.

Definition 2.22 ([25])**.** Let γ be a weighted relation on D, and let C be a clone of operations on D such that $C \subseteq \operatorname{Pol}(\{\gamma\})$. A k-ary weighting ω of clone C is a *weighted polymorphism* of γ (and γ is *improved* by ω) if, for every $\mathbf{x}_1, \dots, \mathbf{x}_k \in \operatorname{Feas}(\gamma)$, it holds

$$0 \geq \sum_{f \in C^{(k)}} \omega(f) \cdot \gamma(f(\mathbf{x}_1, \dots, \mathbf{x}_k))\,. \tag{2.14}$$

We say that ω is a weighted polymorphism of a language Γ if it is a weighted polymorphism of every $\gamma \in \Gamma$. By $\operatorname{wPol}(\Gamma)$, we denote the set of weighted polymorphisms of Γ.

For any set of weightings Ω, we denote by $\operatorname{Imp}(\Omega)$ the set of weighted relations which are improved by all weightings $\omega \in \Omega$.

Note that a k-ary multimorphism $\langle f_1, \dots, f_k \rangle$ corresponds to a k-ary weighted polymorphism

$$f_1 + \dots + f_k - \pi_1^{(k)} - \dots - \pi_k^{(k)}\,. \tag{2.15}$$

Theorem 2.23 ([75, Proposition 12])**.** *For any constraint language Γ, the set* $\operatorname{supp}(\operatorname{wPol}(\Gamma))$ *is a clone.*

The VCSP dichotomy theorem can be stated in terms of weighted polymorphisms as follows.

Theorem 2.24 ([75, 70, 14, 107])**.** *Let Γ be a finite $\overline{\mathbb{Q}}$-valued language. Then Γ is tractable if* $\operatorname{supp}(\operatorname{wPol}(\Gamma))$ *contains a WNU operation; otherwise, Γ is intractable.*

Weighted polymorphisms can be also used to characterise languages that are tractable using a particular algorithm, e.g., the so-called *basic linear programming relaxation* and the *Sherali-Adams relaxation* [94].

Theorem 2.25 ([72])**.** *Let* Γ *be a finite* $\overline{\mathbb{Q}}$*-valued language. Then* Γ *is solvable by the basic linear programming relaxation if and only if* $\operatorname{supp}(\operatorname{wPol}(\Gamma))$ *contains a* k*-ary symmetric operation for all* $k \geq 2$*.*

Theorem 2.26 ([101])**.** *Let* Γ *be a finite* $\overline{\mathbb{Q}}$*-valued language. Then* Γ *is solvable by the Sherali-Adams relaxation if and only if* $\operatorname{supp}(\operatorname{wPol}(\Gamma))$ *contains a* k*-ary WNU operation for all* $k \geq 3$*.*

Example 2.27. Let D be totally ordered, f, g be any binary operations on D, and $\alpha, \beta \in \mathbb{Q}_{\geq 0}$ such that $\alpha + \beta < 2$. We define a binary weighting

$$\omega = \alpha \cdot f + \beta \cdot g + (2 - \alpha - \beta) \cdot \min - \pi_1^{(2)} - \pi_2^{(2)} . \tag{2.16}$$

Let $\Gamma \subseteq \operatorname{Imp}(\{\omega\})$ be a finite set of weighted relations that are improved by ω. Because $\omega \in \operatorname{wPol}(\Gamma)$, it holds $\min \in \operatorname{supp}(\operatorname{wPol}(\Gamma))$. Language Γ is thus tractable by the VCSP complexity classification (Theorem 2.24).

Let $\min^{(k)}$ be the k-ary operation which returns the smallest of its k arguments. As

$$\min^{(k)} = \min^{(k-1)} \left[\min \left[\pi_1^{(k)}, \pi_2^{(k)} \right], \pi_3^{(k)}, \ldots, \pi_k^{(k)} \right] \tag{2.17}$$

and $\operatorname{supp}(\operatorname{wPol}(\Gamma))$ is a clone, it holds $\min^{(k)} \in \operatorname{supp}(\operatorname{wPol}(\Gamma))$ for all $k \geq 2$. By Theorem 2.25, language Γ is solvable by the basic linear programming relaxation.

2.3 Boolean languages

We assume $D = \{0, 1\}$ in this section. For any $r \geq 1$, we denote by $\mathbf{0}^r$ ($\mathbf{1}^r$) the zero (one) r-tuple over D.

Definition 2.28. We denote by $\neg$ the unary negation, i.e., $\neg(0) = 1$ and $\neg(1) = 0$. For a weighted relation $\gamma : D^r \to \overline{\mathbb{Q}}$, we define $\neg(\gamma) : D^r \to \overline{\mathbb{Q}}$ to be the weighted relation such that $\neg(\gamma)(\mathbf{x}) = \gamma(\neg(\mathbf{x}))$.

We order the domain so that $0 < 1$. Hence, operation min (max) corresponds to the logical conjunction (disjunction) when 0 and 1 are interpreted as the logical truth values false and true respectively.

By $\oplus$, we denote the addition modulo 2 operation. In this case, we use the infix notation, i.e., $0 \oplus 0 = 0 = 1 \oplus 1$, $0 \oplus 1 = 1 = 1 \oplus 0$.

By Mn (Mj), we denote the unique ternary minority (majority) operation on D.

One of the tractable classes of Boolean languages are submodular functions.

Example 2.29. A submodular function on a lattice family [93] can be represented by a weighted relation $\gamma : D^{\mathrm{ar}(\gamma)} \to \overline{\mathbb{Q}}$ that satisfies

$$\gamma(\mathbf{x}) + \gamma(\mathbf{y}) \geq \gamma(\min(\mathbf{x}, \mathbf{y})) + \gamma(\max(\mathbf{x}, \mathbf{y})) \tag{2.18}$$

for all $\mathbf{x}, \mathbf{y} \in D^{\mathrm{ar}(\gamma)}$. Equivalently, γ is submodular if and only if it admits $\langle \min, \max \rangle$ as a multimorphism.

We finish by formally stating some of the complexity classifications of Boolean languages that form the basis of our proofs in following sections. Cohen et al. classified all Boolean languages in [28].

Theorem 2.30 ([28, Theorem 7.1])**.** *Let Γ be a finite $\overline{\mathbb{Q}}$-valued language. Then Γ is tractable if it admits any the following multimorphisms: $\langle c_0 \rangle$, $\langle c_1 \rangle$, $\langle \min, \min \rangle$, $\langle \max, \max \rangle$, $\langle \min, \max \rangle$, $\langle \mathrm{Mn}, \mathrm{Mn}, \mathrm{Mn} \rangle$, $\langle \mathrm{Mj}, \mathrm{Mj}, \mathrm{Mj} \rangle$, $\langle \mathrm{Mj}, \mathrm{Mj}, \mathrm{Mn} \rangle$. Otherwise, Γ is intractable.*

In the case of finite-valued languages, the classification can be simplified as follows.

Theorem 2.31 ([28, Corollary 7.11])**.** *Let Γ be a finite $\mathbb{Q}$-valued language. Then Γ is tractable if it admits any of the following multimorphisms: $\langle c_0 \rangle$, $\langle c_1 \rangle$, $\langle \min, \max \rangle$. Otherwise, Γ is intractable.*

The special case of crisp languages is equivalent to the CSP, for which a complexity classification was established by Schaefer [92].

Theorem 2.32 ([92])**.** *Let Γ be a finite $\{0, \infty\}$-valued language. Then Γ is tractable if it is invariant under any of the following operations: c_0, c_1, $\min$, $\max$, Mn, Mj. Otherwise, Γ is intractable.*

Chapter 3

Galois connection for infinite sets

Cohen, Cooper, Creed, Jeavons, and Živný [25] introduced an algebraic theory of weighted clones for classifying the computational complexity of constraint languages, which was further developed in [99, 75]. This theory establishes a Galois connection (a one-to-one correspondence) between *weighted relational clones* (languages closed under expressibility, see Definition 3.2) and *weighted clones* (sets of weightings closed under proper superposition, see Definition 3.6). It extends the algebraic approach to the CSP, which relies on a Galois connection between *relational clones* and *clones* [15]. Such a correspondence makes it possible to utilise deep results from universal algebra, and has ultimately led to the resolution of the (V)CSP dichotomy conjecture [75, 70, 14, 107].

Augmenting a language with weighted relations that are expressible over that language does not affect the complexity of the corresponding VCSP, as all such weighted relations occurring in an instance can be replaced with appropriate expressibility gadgets in polynomial time. Therefore, in order to classify all languages in terms of tractability, it is sufficient to only consider weighted relational clones. A weighted relation is expressible over a finite language Γ if and only if it is improved by every weighted polymorphism of Γ. This means that any finitely-generated weighted relational clone is precisely characterised by its set of weighted polymorphisms. Conversely, a weighting can be obtained by proper superposition from a finite set of weightings Ω if and only if it is a weighted polymorphism of every weighted relation improved by Ω, and hence any finitely-generated weighted clone is precisely characterised by the set of weighted relations it improves. These two theorems constitute the Galois connection from [25].

In this chapter, we consider the possibility of generalising the algebraic approach to infinite languages and infinite sets of weightings. We show that the Galois connection from [25] does not hold in general. In particular, we give an example of an infinite language Γ such that some weighted relations are not expressible over it even though they are improved by every weighted polymorphism of Γ (see Theorem 3.10), and an example of an infinite set of weightings Ω such that some weightings cannot be obtained from it by proper superposition even though they are weighted polymorphisms of every weighted relation improved by Ω (see Theorem 3.11). Then we establish a new Galois connection which applies to infinite sets as well. In order to do so, we modify the definition of weighted (relational) clones; namely, we require them to be topologically closed. This makes it necessary to work in a complete field, and hence we allow real values in place of rationals.

As an independent improvement of the notion of weighted relational clones, we require them to be closed under operation Opt. We show that closing a language under Opt preserves its tractability, and hence it is sufficient to classify only languages satisfying this stronger condition. Changing the definition of weighted relational clones also simplifies the structure of weighted clones; for details, see Section 3.3.

Section 3.1 presents the Galois connection introduced in [25]. In Section 3.2, we examine the extent to which this Galois connection applies to infinite sets, and give an example for which it does not hold. We establish a new Galois connection in Section 3.3.

3.1 Previous Galois connection

3.1.1 Weighted relational clones

Definition 3.1. A constraint language Γ is called a *weighted relational clone* if it contains the binary equality relation $\rho_=$ and the unary empty relation $\rho_\emptyset$,[1] and is closed under addition, minimisation, coordinate mapping, scaling by non-negative rational constants, and addition of rational constants.

We define $\mathrm{wRelClone}(\Gamma)$ to be the smallest weighted relational clone containing Γ.

Note that any weighted relational clone is also closed under operation Feas, as $\mathrm{Feas}(\gamma) = 0 \cdot \gamma$.

An equivalent definition of weighted relational clones (see Theorem 3.4) is based on the notion of expressing weighted relations by gadgets.

---

[1] Although the definition in [25] does not require the inclusion of $\rho_\emptyset$, the proofs there implicitly assume its presence in any weighted relational clone.

Definition 3.2. Let Γ be a constraint language, $I = (V, D, \phi_I)$ an instance of VCSP(Γ), and $L = (v_1, \dots, v_r)$ a list of (not necessarily distinct) variables from V. The *projection* of I onto L, denoted by $\pi_L(I)$, is the r-ary weighted relation on D defined by

$$\pi_L(I)(x_1, \dots, x_r) = \min \{\phi_I(s) \mid s : V \to D \ \wedge \ s(v_i) = x_i \text{ for all } i \in [r]\} \ , \tag{3.1}$$

where the minimum over the empty set equals ∞.

We say that a weighted relation γ is *expressible* over a constraint language Γ if $\gamma = \pi_L(I)$ for some $I \in \text{VCSP}(\Gamma)$ and a list of variables L. We call the pair (I, L) a *gadget* for expressing γ over Γ.

Example 3.3. For any language Γ, we can express the binary equality relation $\rho_=$ over Γ using the following gadget. Let $I \in \text{VCSP}(\Gamma)$ be the instance with a single variable v and no constraints, and let $L = (v, v)$. Then $\pi_L(I) = \rho_=$.

Theorem 3.4 ([25, Proposition 4.5])**.** *A constraint language is a weighted relational clone if and only if it contains the unary empty relation $\rho_\emptyset$ and is closed under expressibility and addition of rational constants.*

The following result implies that, in order to classify constraint languages in terms of their (in)tractability, it is sufficient to consider only weighted relational clones.

Theorem 3.5 ([25, Theorems 2.4 and 4.3])**.** *A constraint language Γ is tractable if and only if* wRelClone(Γ) *is tractable, and Γ is intractable if and only if* wRelClone(Γ) *is intractable.*

3.1.2 Weighted clones

We call a function $\omega : C^{(k)} \to \mathbb{Q}$ that satisfies (2.12) but assigns a negative weight to some operation $f \not\in \Pi_D^{(k)}$ an *improper weighting*. In order to emphasise the distinction we may also call a weighting a *proper weighting*.

A weighting ω of a clone C can be also seen as a weighting of a clone $C' \supseteq C$, where we extend ω to C' by defining $\omega(f) = 0$ for $f \in C' \setminus C$. Given a set Ω of weightings of different clones over D, we consider them as weightings of the smallest clone such that it contains all the clones associated with the weightings in Ω.

Definition 3.6. For any k-ary weighting ω of a clone C and any ℓ-ary operations $g_1, \dots, g_k \in C^{(\ell)}$, the *superposition* of ω and $g_1, \dots, g_k$ is the function $\omega[g_1, \dots, g_k] : C^{(\ell)} \to \mathbb{Q}$ defined by

$$\omega[g_1, \dots, g_k](f') = \sum_{f \in C^{(k)} \ \wedge \ f[g_1, \dots, g_k] = f'} \omega(f) \, . \tag{3.2}$$

By convention, the value of an empty sum equals 0.

A superposition which results in a proper weighting (i.e., negative weights are only assigned to projections) is called a *proper* superposition.

Definition 3.7. A non-empty set of weightings Ω of a clone C is called a *weighted clone* if it is closed under scaling by non-negative rational constants, addition of weightings of equal arity, and proper superposition with operations from C. Clone C is called the *support clone* of Ω.

We define $\mathrm{wClone}(\Omega)$ to be the smallest weighted clone containing Ω.

The main result in [25] establishes a 1-to-1 correspondence between weighted relational clones and weighted clones.

Theorem 3.8 ([25]). *For any finite constraint language* Γ*, it holds*

$$\mathrm{Imp}(\mathrm{wPol}(\Gamma)) = \mathrm{wRelClone}(\Gamma)\,. \tag{3.3}$$

For any finite set of weightings Ω*, it holds*

$$\mathrm{wPol}(\mathrm{Imp}(\Omega)) = \mathrm{wClone}(\Omega)\,. \tag{3.4}$$

Thus, when trying to identify tractable constraint languages, it is sufficient to consider only languages of the form $\mathrm{Imp}(\Omega)$ for some weighted clone Ω.

3.2 Extent of the previous Galois connection

First, we show that Theorem 3.8 can be extended to certain infinite sets. For example, consider an infinite language Γ such that $\mathrm{wRelClone}(\Gamma)$ is finitely generated, i.e., $\mathrm{wRelClone}(\Gamma) = \mathrm{wRelClone}(\Gamma')$ for some finite language Γ'. The following theorem implies that $\mathrm{Imp}(\mathrm{wPol}(\Gamma)) = \mathrm{wRelClone}(\Gamma)$, since it holds $\mathrm{wRelClone}(\Gamma) = \mathrm{Imp}(\Omega)$ for $\Omega = \mathrm{wPol}(\Gamma')$ by Theorem 3.8.

Theorem 3.9. *For any constraint language* Γ*, it holds* $\mathrm{Imp}(\mathrm{wPol}(\Gamma)) = \mathrm{wRelClone}(\Gamma)$ *if and only if* $\mathrm{wRelClone}(\Gamma) = \mathrm{Imp}(\Omega)$ *for some set of weightings* Ω*.*

For any set of weightings Ω*, it holds* $\mathrm{wPol}(\mathrm{Imp}(\Omega)) = \mathrm{wClone}(\Omega)$ *if and only if* $\mathrm{wClone}(\Omega) = \mathrm{wPol}(\Gamma)$ *for some constraint language* Γ*.*

Proof. If a constraint language Γ satisfies $\mathrm{Imp}(\mathrm{wPol}(\Gamma)) = \mathrm{wRelClone}(\Gamma)$, it holds $\mathrm{wRelClone}(\Gamma) = \mathrm{Imp}(\Omega)$ for $\Omega = \mathrm{wPol}(\Gamma)$.

Suppose that $\mathrm{wRelClone}(\Gamma) = \mathrm{Imp}(\Omega)$ for some set of weightings Ω. As $\Gamma \subseteq \mathrm{wRelClone}(\Gamma)$, set Ω improves Γ. Therefore, $\Omega \subseteq \mathrm{wPol}(\Gamma)$ and

$$\mathrm{Imp}(\mathrm{wPol}(\Gamma)) \subseteq \mathrm{Imp}(\Omega) = \mathrm{wRelClone}(\Gamma)\,. \tag{3.5}$$

The converse inclusion $\mathrm{wRelClone}(\Gamma) \subseteq \mathrm{Imp}(\mathrm{wPol}(\Gamma))$ follows from the fact that $\mathrm{Imp}(\mathrm{wPol}(\Gamma))$ is a weighted relational clone [25, Proposition 6.2] that contains Γ.

The second claim of the theorem can be proved analogously. □

In contrast, our next results show that Theorem 3.8 does *not* hold for some infinite constraint languages and infinite sets of weightings.

Theorem 3.10. *There is an infinite constraint language Γ such that*

$$\mathrm{Imp}(\mathrm{wPol}(\Gamma)) \not\subseteq \mathrm{wRelClone}(\Gamma)\,. \tag{3.6}$$

Proof. We set the domain to be $D = \{0, 1, 2\}$ and choose a positive *irrational* number t. Let U be the set of unary weighted relations μ such that, if $\mu(0), \mu(1), \mu(2)$ are all finite, it holds

$$\mu(2) - \mu(0) \geq (1 + t) \cdot (\mu(1) - \mu(0))\,. \tag{3.7}$$

Note that set U contains all unary crisp weighted relations, and it is closed under addition, scaling by non-negative rational constants, and addition of rational constants.

Let Γ equal the set of weighted relations γ that can be written as

$$\gamma(x_1, \dots, x_r) = \sum_{i=1}^{r} \mu_i(x_i) + \sum_{(i,j)\in S} \rho_=(x_i, x_j)\,, \tag{3.8}$$

where $r = \mathrm{ar}(\gamma)$, $\mu_i \in U$ for all $i \in [r]$, and S is an equivalence relation on $[r]$.

Claim 1. *Language Γ is a weighted relational clone.*

Proof. Clearly, crisp weighted relations $\rho_=, \rho_\emptyset \in U$ are included in Γ.

We show that Γ is closed under minimisation. Without loss of generality, let us assume we minimise an r-ary weighted relation γ ($r \geq 2$) over the last variable (x_r). If the equivalence class of r in S is a singleton, we simply add the value of $\min_{x_r \in D} \mu_r(x_r)$ to (say) μ_1. Otherwise, we can pick any $i \neq r$ such that $(i, r) \in S$ and replace weighted relation μ_i with $\mu_i + \mu_r$.

Similarly, Γ is closed under the remaining operations (i.e., addition, coordinate mapping, scaling by non-negative rational constants, and addition of rational constants). ■

Claim 2. *Let $k \geq 1$, $\omega \in \mathrm{wPol}^{(k)}(\Gamma)$, and $\mathbf{x} \in D^k$. For every $a \in D$, it holds*

$$\sum_{f(\mathbf{x})=a} \omega(f) = 0\,. \tag{3.9}$$

Proof. We denote $s_a = \sum_{f(\mathbf{x})=a} \omega(f)$. Note that $s_0 + s_1 + s_2 = 0$.

For any rational $u < t$ and $v > t$, we define unary weighted relations $\mu_u^-, \mu_v^+ \in U \subseteq \Gamma$ by

$$\mu_u^-(x) = \begin{cases} 0 & \text{for } x = 0 \\ -1 & \text{for } x = 1 \\ -1-u & \text{for } x = 2 \end{cases}, \qquad \mu_v^+(x) = \begin{cases} 0 & \text{for } x = 0 \\ 1 & \text{for } x = 1 \\ 1+v & \text{for } x = 2 \end{cases}. \tag{3.10}$$

Because weighting ω improves μ_v^+, it holds

$$0 \geq \sum_{f \in \mathrm{Pol}^{(k)}(\Gamma)} \omega(f) \cdot \mu_v^+(f(\mathbf{x})) = s_1 + s_2 \cdot (1+v)\,, \tag{3.11}$$

and hence $s_0 \geq v \cdot s_2$. Similarly, it holds $s_0 \leq u \cdot s_2$, as ω improves μ_u^-. Both u and v can be chosen arbitrarily close to t, so it must hold $s_0 = t \cdot s_2$. However, s_0 and s_2 are rational while t is not. Therefore, it holds $s_0 = s_1 = s_2 = 0$. ■

It follows from Claim 2 that wPol(Γ) improves any unary finite-valued weighted relation. Consider a unary weighted relation μ defined by $\mu(0) = 0$ and $\mu(1) = \mu(2) = 1$. It holds $\mu \in \mathrm{Imp}(\mathrm{wPol}(\Gamma))$. However, μ violates (3.7), and therefore $\mu \notin \Gamma = \mathrm{wRelClone}(\Gamma)$. □

Theorem 3.11. *There is an infinite set of weightings Ω such that*

$$\mathrm{wPol}(\mathrm{Imp}(\Omega)) \not\subseteq \mathrm{wClone}(\Omega)\,. \tag{3.12}$$

Proof. We set the domain to be $D = \{0, 1, 2\}$ and choose a positive *irrational* number t. We define a set of weightings Ω of the support clone $\mathcal{O}_D$. For every arity $k \geq 1$, let $\Omega^{(k)}$ consist of weightings ω such that, for all $\mathbf{x} \in D^k$,

$$t \cdot \sum_{f(\mathbf{x})=2} \omega(f) \leq \sum_{f(\mathbf{x})=0} \omega(f)\,. \tag{3.13}$$

Claim 1. *Set Ω is a weighted clone.*

Proof. Clearly, Ω is closed under addition of weightings and non-negative scaling.

Consider any $\omega \in \Omega^{(k)}$ and ℓ-ary operations $g_1, \ldots, g_k$ such that $\omega[g_1, \ldots, g_k]$ is a proper weighting. For any $\mathbf{x} \in D^\ell$ and $a \in D$, it holds

$$\sum_{f(\mathbf{x})=a} \omega[g_1, \ldots, g_k](f) = \sum_{f[g_1,\ldots,g_k](\mathbf{x})=a} \omega(f) = \sum_{f(\mathbf{y})=a} \omega(f)\,, \tag{3.14}$$

where $\mathbf{y} = (g_1(\mathbf{x}), \ldots, g_k(\mathbf{x}))$. As ω satisfies (3.13) for tuple $\mathbf{y}$, the superposition $\omega[g_1, \ldots, g_k]$ satisfies it for tuple $\mathbf{x}$. Therefore, $\omega[g_1, \ldots, g_k] \in \Omega$. ■

For any $a_0, a_1, a_2 \in D$, let $f_{a_0 a_1 a_2}$ be a unary operation defined by $f_{a_0 a_1 a_2}(x) = a_x$ for all $x \in D$.

Claim 2. *For any* $\gamma \in \operatorname{Imp}(\Omega)$ *and* $\mathbf{x} \in \operatorname{Feas}(\gamma)$*, it holds* $\gamma(\mathbf{x}) = \gamma(f_{022}(\mathbf{x}))$*.*

Proof. For any rational $v > t$, we define a unary weighting $\mu_v^{(1)}$ by

$$\mu_v^{(1)} = -(1+v) \cdot \pi_1^{(1)} + v \cdot f_{002} + f_{022}\,. \tag{3.15}$$

Because $\mu_v^{(1)} \in \Omega$, it holds

$$v \cdot \gamma(f_{002}(\mathbf{x})) + \gamma(f_{022}(\mathbf{x})) \leq (1+v) \cdot \gamma(\mathbf{x}) \tag{3.16}$$

$$\gamma(f_{022}(\mathbf{x})) - \gamma(\mathbf{x}) \leq v \cdot (\gamma(\mathbf{x}) - \gamma(f_{002}(\mathbf{x})))\,. \tag{3.17}$$

Let g be a binary operation defined by

$$g(x,y) = \begin{cases} 1 & \text{for } x = 0 \ \wedge \ y = 2 \\ 2 & \text{for } x = 2 \ \wedge \ y = 2\,. \\ 0 & \text{otherwise} \end{cases} \tag{3.18}$$

For any *positive* rational $u < t$, we define a binary weighting $\mu_u^{(2)}$ by

$$\mu_u^{(2)} = -u \cdot \pi_1^{(2)} - \pi_2^{(2)} + (1+u) \cdot g\,. \tag{3.19}$$

Because $\mu_u^{(2)} \in \Omega$ and $g(f_{002}(a), f_{022}(a)) = a$ for all $a \in D$, it holds

$$(1+u) \cdot \gamma(g(f_{002}(\mathbf{x}), f_{022}(\mathbf{x}))) \leq u \cdot \gamma(f_{002}(\mathbf{x})) + \gamma(f_{022}(\mathbf{x})) \tag{3.20}$$

$$u \cdot (\gamma(\mathbf{x}) - \gamma(f_{002}(\mathbf{x}))) \leq \gamma(f_{022}(\mathbf{x})) - \gamma(\mathbf{x})\,. \tag{3.21}$$

We can choose both u and v arbitrarily close to t, and hence (3.17), (3.21) imply

$$\gamma(f_{022}(\mathbf{x})) - \gamma(\mathbf{x}) = t \cdot (\gamma(\mathbf{x}) - \gamma(f_{002}(\mathbf{x})))\,. \tag{3.22}$$

However, weights assigned by γ are rational while t is not. Therefore, it holds $\gamma(\mathbf{x}) = \gamma(f_{002}(\mathbf{x})) = \gamma(f_{022}(\mathbf{x}))$. ■

Consider a unary weighting $\omega = -\pi_1^{(1)} + f_{022}$. By Claim 2, it holds $\omega \in \operatorname{wPol}(\operatorname{Imp}(\Omega))$. However, ω violates (3.13) for $\mathbf{x} = (1)$, and therefore $\omega \notin \Omega = \operatorname{wClone}(\Omega)$. □

3.3 New Galois connection

In order to establish a Galois connection for infinite sets, we need to work in a complete field, and require weighted relational clones and weighted clones to be topologically closed (see the counterexamples in Theorems 3.10 and 3.11). Hence, we allow real values in place of rationals.

In particular, this change affects the values assigned by weighted relations, weightings, and the weights of constraints in VCSP instances. To distinguish the redefined concepts from their rational-valued counterparts, we use a subscript $\mathbb{R}$. We denote by $\overline{\mathbb{R}} = \mathbb{R} \cup \{\infty\}$ the set of extended real numbers.

We also need to relax the notion of tractability of a constraint language so that it is preserved by the topological closure operation, as our new definition of weighted relational clones requires them to be topologically closed. Instead of considering only optimal solutions, we accept any solution with a value exceeding the optimum by at most an additive constant ϵ (for arbitrarily small $\epsilon > 0$; see Definition 3.20).

Inspired by weighted pp-definitions [98], we alter the definition of weighted relational clones further and require them to be closed under operation Opt. This is justified by Lemma 3.19, in which we prove that the inclusion of Opt preserves tractability; in fact, it preserves the stronger (exact) tractability as well. One of the benefits of this change is a simplified structure of weighted clones: According to the original definition, a weighted clone is determined by a support clone C and a set of weightings of C. By our new definition, a k-ary weighting assigns weights to all k-ary operations, and a weighted clone is simply a set of weightings. Moreover, any non-projection polymorphism of a weighted relational clone Γ is now guaranteed to be assigned a positive weight by some weighted polymorphism of Γ (see Corollary 3.37).

The new Galois connection is stated as follows.

Theorem 3.12 (cf. Theorem 3.8)**.** *For any constraint language* Γ*, it holds* $\mathrm{Imp}_{\mathbb{R}}(\mathrm{wPol}_{\mathbb{R}}(\Gamma)) = \mathrm{wRelClone}_{\mathbb{R}}(\Gamma)$*. For any set of weightings* Ω*, it holds* $\mathrm{wPol}_{\mathbb{R}}(\mathrm{Imp}_{\mathbb{R}}(\Omega)) = \mathrm{wClone}_{\mathbb{R}}(\Omega)$*.*

In Section 3.3.1, we redefine weighted relational clones and prove that they preserve tractability in the relaxed sense. In Section 3.3.2, we redefine weighted clones. In Section 3.3.3, we prove Theorem 3.12, stated as Theorems 3.36 and 3.38.

3.3.1 Weighted relational clones

Definition 3.13 (cf. Definition 2.2)**.** An r-ary *weighted relation* over D is a mapping $\gamma : D^r \to \overline{\mathbb{R}}$.

For any fixed arity r and any $F \subseteq D^r$, let

$$H_{r,F} = \left\{ \gamma : D^r \to \overline{\mathbb{R}} \;\middle|\; \mathrm{Feas}(\gamma) = F \right\} . \tag{3.23}$$

Set $H_{r,F}$ with an inner product defined by

$$\langle \alpha, \beta \rangle = \sum_{\mathbf{x} \in F} \alpha(\mathbf{x}) \cdot \beta(\mathbf{x}) \tag{3.24}$$

is a real Hilbert space. The set of all weighted relations is a disjoint union of such Hilbert spaces for all r and F, and therefore a topological space with the disjoint union topology induced by inner products on the underlying Hilbert spaces. When we say a set of weighted relations is open/closed, we will be referring to this topology.

Definition 3.14 (cf. Definition 3.1)**.** A constraint language Γ is called a *weighted relational clone* if it contains the binary equality relation $\rho_=$ and the unary empty relation $\rho_\emptyset$, is closed under addition, minimisation, coordinate mapping, scaling by non-negative real constants, addition of real constants, and operation Opt, and is topologically closed.

We define $\mathrm{wRelClone}_{\mathbb{R}}(\Gamma)$ to be the smallest weighted relational clone containing Γ.

Any expressibility gadget can be simulated using the binary equality relation $\rho_=$, addition of weighted relations, minimisation, coordinate mapping, and scaling by non-negative real constants. Conversely, all of these can be simulated using appropriate gadgets. Therefore, we obtain the following characterisation of weighted relational clones.

Theorem 3.15 (cf. Theorem 3.4)**.** *A constraint language is a weighted relational clone if and only if it contains the unary empty relation $\rho_\emptyset$, is closed under expressibility, addition of real constants, and operation* Opt*, and is topologically closed.*

Observation 3.16. *All solutions to an instance with a constraint γ such that* $\mathrm{Feas}(\gamma) = \emptyset$ *are infeasible.*

Lemma 3.17. *Let γ be a weighted relation expressible over a constraint language Γ. Then*

$$\mathrm{VCSP}_{\mathbb{R}}(\Gamma \cup \{\gamma\}) \leq_p \mathrm{VCSP}_{\mathbb{R}}(\Gamma) . \tag{3.25}$$

Proof. Let (J, L) be a gadget for expressing γ over Γ, where $L = (v_1, \ldots, v_r)$. Given an instance $I \in \mathrm{VCSP}_{\mathbb{R}}(\Gamma \cup \{\gamma\})$, we replace every constraint of the form $\gamma(x_1, \ldots, x_r)$ in I with a copy of J. The weights of the introduced constraints are scaled by the weight of the constraint $\gamma(x_1, \ldots, x_r)$ in I. We also add constraints $\rho_=(v_i, x_i)$ for all $i \in [r]$. This reduces the problem of solving I to an instance of $\mathrm{VCSP}_{\mathbb{R}}(\Gamma \cup \{\rho_=\})$. To eliminate any constraint of the form $\rho_=(x, y)$, we replace variables x and y with a single new one. ☐

Observation 3.18. *Adding a real constant to a weighted relation changes the value of every solution by the same amount, and hence does not affect the set of optimal solutions.*

Lemma 3.19. *Let Γ be a constraint language and $\gamma \in \Gamma$. Then*

$$\mathrm{VCSP}_{\mathbb{R}}(\Gamma \cup \{\mathrm{Opt}(\gamma)\}) \leq_p \mathrm{VCSP}_{\mathbb{R}}(\Gamma)\,. \tag{3.26}$$

Proof. We may without loss of generality assume that $\mathrm{Feas}(\gamma) \neq \emptyset$ (by Observation 3.16). Also, we may assume that the minimum values assigned by γ and $\mathrm{Opt}(\gamma)$ equal 0, and that all weighted relations in Γ assign non-negative values (by Observation 3.18). If γ is crisp, it holds $\mathrm{Opt}(\gamma) = \mathrm{Feas}(\gamma) = 0 \cdot \gamma$, and hence we may simply replace every constraint of the form $w \cdot \mathrm{Opt}(\gamma)(\mathbf{x})$ with a constraint $0 \cdot \gamma(\mathbf{x})$. Otherwise, γ assigns a positive value to some tuple; let m denote the smallest such value.

Given an instance $I \in \mathrm{VCSP}_{\mathbb{R}}(\Gamma \cup \{\mathrm{Opt}(\gamma)\})$, let M be an upper bound on the value of any feasible solution to I. We replace every constraint of the form $w \cdot \mathrm{Opt}(\gamma)(\mathbf{x})$ in I with a constraint $(M/m + 1) \cdot \gamma(\mathbf{x})$ to obtain an instance $I' \in \mathrm{VCSP}_{\mathbb{R}}(\Gamma)$.

Any feasible solution to instance I gets assigned the same value by I'. Any infeasible solution to instance I is either infeasible for I' as well, or it incurs an infinite value from a constraint of the form $w \cdot \mathrm{Opt}(\gamma)(\mathbf{x})$ in I and thus a value of at least $(M/m + 1) \cdot m > M$ from a constraint $\gamma(\mathbf{x})$ in I'. Therefore, an optimal solution to I' is optimal for I as well. ☐

In contrast to the rest of the closure operations defining weighted relational clones, we can only show that topological closure preserves tractability in the following relaxed sense.

Definition 3.20. Let $\epsilon > 0$. We say that a labelling $s : V \to D$ of a VCSP instance $I = (V, D, \phi_I)$ is *ϵ-optimal* if it is feasible and $\phi_I(s) \leq \phi_I(s') + \epsilon$ for all labellings s'. An algorithm *ϵ-solves* a VCSP instance if it correctly decides whether a feasible labelling exists and, if so, it finds an ϵ-optimal one.

A language Γ is called *ϵ-tractable* if, for every finite $\Gamma' \subseteq \Gamma$, $\mathrm{VCSP}_{\mathbb{R}}(\Gamma')$ can be ϵ-solved in polynomial time. If there exists a finite $\Gamma' \subseteq \Gamma$ such that it is NP-hard to ϵ-solve $\mathrm{VCSP}_{\mathbb{R}}(\Gamma')$, language Γ is called *ϵ-intractable*.

Lemma 3.21. *Let* Γ *be a constraint language,* $\overline{\Gamma}$ *its topological closure, and* $\epsilon > 0$*. Then* ϵ*-solving* $\mathrm{VCSP}_{\mathbb{R}}(\overline{\Gamma})$ *reduces in polynomial time to solving* $\mathrm{VCSP}_{\mathbb{R}}(\Gamma)$*.*

Proof. Given an instance $I \in \mathrm{VCSP}_{\mathbb{R}}(\overline{\Gamma})$, we construct an instance $I' \in \mathrm{VCSP}_{\mathbb{R}}(\Gamma)$ on the same set of variables V as follows.

Let M be a positive upper bound on the sum of weights of constraints in I. For any $\gamma \in \overline{\Gamma}$, there is a weighted relation $\gamma' \in \Gamma$ with $\mathrm{ar}(\gamma') = \mathrm{ar}(\gamma)$ and $\mathrm{Feas}(\gamma') = \mathrm{Feas}(\gamma)$ such that the distance between γ and γ' is at most $\epsilon/2M$. We obtain the sought instance I' by replacing every constraint of the form $\gamma(\mathbf{x})$ in I with its counterpart $\gamma'(\mathbf{x})$.

For any labelling $t : V \to D$, it holds

$$I(t) - \frac{\epsilon}{2} \leq I'(t) \leq I(t) + \frac{\epsilon}{2}\,. \tag{3.27}$$

Let s and s' be optimal solutions to I and I' respectively. Since

$$I(s') - \frac{\epsilon}{2} \leq I'(s') \leq I'(s) \leq I(s) + \frac{\epsilon}{2}\,, \tag{3.28}$$

s' is an ϵ-optimal solution to I. □

The following result justifies the definition of weighted relational clones.

Theorem 3.22 (cf. Theorem 3.5)**.** *Let* Γ *be a constraint language and* $\epsilon > 0$*. If* Γ *is tractable, then* $\mathrm{wRelClone}_{\mathbb{R}}(\Gamma)$ *is* ϵ*-tractable. If* $\mathrm{wRelClone}_{\mathbb{R}}(\Gamma)$ *is* ϵ*-intractable, then* Γ *is intractable.*

Proof. Recall the characterisation of weighted relational clones from Theorem 3.15. The claim follows from Observation 3.16, Lemma 3.17, Observation 3.18, Lemma 3.19, and Lemma 3.21. □

3.3.2 Weighted clones

Definition 3.23 (cf. Definition 2.21)**.** A k-ary *weighting* is a function $\omega : \mathcal{O}_D^{(k)} \to \mathbb{R}$ such that $\omega(f) < 0$ only if f is a projection and

$$\sum_{f \in \mathcal{O}_D^{(k)}} \omega(f) = 0\,. \tag{3.29}$$

We define the *support* of ω by

$$\mathrm{supp}(\omega) = \Pi_D^{(k)} \cup \left\{ f \in \mathcal{O}_D^{(k)} \;\middle|\; \omega(f) > 0 \right\}\,. \tag{3.30}$$

For any set of weightings Ω, we define $\operatorname{supp}(\Omega) = \Pi_D \cup \bigcup_{\omega\in\Omega} \operatorname{supp}(\omega)$.

For any fixed arity k, let $H_k = \mathcal{O}_D^{(k)} \to \mathbb{R}$. Set H_k with an inner product defined by

$$\langle \alpha, \beta \rangle = \sum_{f\in\mathcal{O}_D^{(k)}} \alpha(f)\cdot\beta(f) \tag{3.31}$$

is a real Hilbert space. The set of all weightings lies in the disjoint union of such Hilbert spaces for all k, which is a topological space with the disjoint union topology induced by inner products on the underlying Hilbert spaces. When we say a set of weightings is open/closed, we will be referring to this topology.

Observation 3.24. *Any point of closure of a set of weightings is itself a weighting.*

Definition 3.25 (cf. Definition 2.22)**.** Let γ be a weighted relation on D. A k-ary weighting ω is a *weighted polymorphism* of γ (and γ is *improved* by ω) if $\operatorname{supp}(\omega) \subseteq \operatorname{Pol}(\{\gamma\})$ and, for every $\mathbf{x}_1, \ldots, \mathbf{x}_k \in \operatorname{Feas}(\gamma)$, it holds

$$\sum_{f\in\operatorname{supp}(\omega)} \omega(f)\cdot\gamma(f(\mathbf{x}_1,\ldots,\mathbf{x}_k)) \leq 0\,. \tag{3.32}$$

We say that ω is a weighted polymorphism of a language Γ if it is a weighted polymorphism of every $\gamma\in\Gamma$. By $\operatorname{wPol}_{\mathbb{R}}(\Gamma)$ we denote the set of weighted polymorphisms of Γ.

For any set of weightings Ω, we denote by $\operatorname{Imp}_{\mathbb{R}}(\Omega)$ the set of weighted relations which are improved by all weightings $\omega\in\Omega$.

Definition 3.26 (cf. Definition 3.7)**.** A non-empty set of weightings Ω is called a *weighted clone* if it is closed under scaling by non-negative real constants, addition of weightings of equal arity, and proper superposition with operations from $\operatorname{supp}(\Omega)$, and is topologically closed.

We define $\operatorname{wClone}_{\mathbb{R}}(\Omega)$ to be the smallest weighted clone containing Ω.

3.3.3 Proofs

We start with a few auxiliary lemmas. Then we relate the concept of weighted polymorphisms and polar cones (see Definition 3.30), which enables us to apply known properties of polar cones in our proofs. Finally, we establish a new Galois connection in Theorems 3.36 and 3.38.

It is often convenient to build a desired proper weighting by taking a sum of (possibly) improper superpositions. The following lemma, which is an analogue of [25, Lemma 6.4], shows that weighted clones are closed under such constructions.

Lemma 3.27. *Let Ω be a weighted clone, $\omega_1, \dots, \omega_n \in \Omega$, and $c_1, \dots, c_n \in \mathbb{R}_{\geq 0}$. Let $k \geq 1$, and $g_{i,j} \in \operatorname{supp}(\Omega)$ be a k-ary operation for all $i \in [n]$, $j \in [\operatorname{ar}(\omega_i)]$. If the k-ary weighting μ defined by*

$$\mu = \sum_{i=1}^{n} c_i \cdot \omega_i \left[g_{i,1}, \dots, g_{i,\operatorname{ar}(\omega_i)}\right] \tag{3.33}$$

is proper, then $\mu \in \Omega$.

Proof. We show that weighting μ can be constructed using proper superpositions only.

Let us denote $t = \sum_{i \in [n]} \operatorname{ar}(\omega_i)$. For any $m \in [n]$, let $s_m = \sum_{1 \leq i < m} \operatorname{ar}(\omega_i)$. A superposition with projections is always proper (as all negative weights are transferred to projections), and therefore the t-ary weighting μ' defined by

$$\mu' = \sum_{i=1}^{n} c_i \cdot \omega_i \left[\pi^{(t)}_{s_i+1}, \dots, \pi^{(t)}_{s_i+\operatorname{ar}(\omega_i)}\right] \tag{3.34}$$

belongs to Ω. Since $\mu = \mu'[g_{1,1}, \dots, g_{1,\operatorname{ar}(\omega_1)}, g_{2,1}, \dots, g_{n,\operatorname{ar}(\omega_n)}]$, we get $\mu \in \Omega$. ☐

The following lemma has also been observed in [75, 101].

Lemma 3.28. *For any weighted clone Ω, $\operatorname{supp}(\Omega)$ is a clone.*

Proof. We will denote $\operatorname{supp}(\Omega)$ by C. Since it contains all projections, we only need to show that it is closed under superposition.

Let $f \in C^{(k)}$ and $g_1, \dots, g_k \in C^{(\ell)}$. If $f[g_1, \dots, g_k]$ is a projection or is equal to g_i for some i, then it clearly belongs to C. Otherwise, f is not a projection and therefore there is a k-ary weighting $\omega \in \Omega$ for which $\omega(f) > 0$. Weighting $\omega[g_1, \dots, g_k]$ certainly assigns a positive weight to $f[g_1, \dots, g_k]$ (we are using the fact that only operations $g_1, \dots, g_k$ may receive negative weight from projections in ω). However, it is possibly improper, as it may assign a negative weight to some g_i that is not a projection.

We denote by G the set of such operations $g \in \{g_1, \dots, g_k\}$ that are not projections and $\omega[g_1, \dots, g_k](g) < 0$. For any $g \in G$, there is an ℓ-ary weighting $\mu_g \in \Omega$ for which $\mu_g(g) > 0$. Then the ℓ-ary weighting defined by

$$\omega[g_1, \dots, g_k] + \sum_{g \in G} \frac{-\omega[g_1, \dots, g_k](g)}{\mu_g(g)} \cdot \mu_g \tag{3.35}$$

is proper, belongs to Ω (by Lemma 3.27), and assigns a positive weight to $f[g_1, \dots, g_k]$. ☐

Lemma 3.29. *For any weighted clone* Ω*, it holds* $\operatorname{supp}(\Omega) = \operatorname{Pol}(\operatorname{Imp}_{\mathbb{R}}(\Omega))$.

Proof. Projections are polymorphisms of every weighted relation, and any operation f with $\omega(f) > 0$ for some $\omega \in \Omega$ is a polymorphism of $\operatorname{Imp}_{\mathbb{R}}(\Omega)$ by the definition of weighted polymorphisms. Therefore, $\operatorname{supp}(\Omega) \subseteq \operatorname{Pol}(\operatorname{Imp}_{\mathbb{R}}(\Omega))$.

Let $\operatorname{Inv}(\operatorname{supp}(\Omega))$ be the set of crisp weighted relations that are invariant under all operations from $\operatorname{supp}(\Omega)$ (i.e., operations from $\operatorname{supp}(\Omega)$ are their polymorphisms). Because such weighted relations are improved by Ω, it holds $\operatorname{Inv}(\operatorname{supp}(\Omega)) \subseteq \operatorname{Imp}_{\mathbb{R}}(\Omega)$. Therefore,

$$\operatorname{Pol}(\operatorname{Imp}_{\mathbb{R}}(\Omega)) \subseteq \operatorname{Pol}(\operatorname{Inv}(\operatorname{supp}(\Omega))) = \operatorname{supp}(\Omega)\,, \tag{3.36}$$

where the last equality follows from the Galois connection between relational clones and clones of operations [9, 56] and Lemma 3.28. ☐

Inequality (3.32) defining weighted polymorphisms translates to an inequality $\langle \alpha, \beta \rangle \le 0$ for appropriately chosen vectors α and β in a Hilbert space, where α depends on weighting ω and β depends on weighted relation γ and tuples $\mathbf{x}_1, \dots, \mathbf{x}_k$ (see Lemma 3.33). Therefore, set $\operatorname{wPol}_{\mathbb{R}}(\Gamma)$ for a constraint language Γ relates to a polar cone (defined below) of an appropriately chosen set of vectors.

Definition 3.30. Let H be a Hilbert space. A set $K \subseteq H$ is a *convex cone* if it is closed under scaling by non-negative real constants and addition. The *polar cone* (also called the *internal polar cone*) of a set $K \subseteq H$ is denoted by K° and defined by

$$K^\circ = \{\alpha \in H \mid \langle \alpha, \beta \rangle \le 0 \text{ for all } \beta \in K\}\,. \tag{3.37}$$

Theorem 3.31 ([12, 61])**.** *The polar cone of any* $K \subseteq H$ *is a topologically closed convex cone. Set* $K^{\circ\circ} = (K^\circ)^\circ$ *equals the smallest topologically closed convex cone containing* K*.*

Definition 3.32. Let $X = (\mathbf{x}^1, \dots, \mathbf{x}^k) \in (D^m)^k$ be a sequence of m-tuples over D, where $\mathbf{x}^i = (x^i_1, \dots, x^i_m)$ for all $i \in [k]$. By X^T, we denote the transpose of X, i.e. the sequence of k-tuples $(\mathbf{y}^1, \dots, \mathbf{y}^m) \in (D^k)^m$ such that $\mathbf{y}^i = (x^1_i, \dots, x^k_i)$ for all $i \in [m]$.

For any k-ary operation f, we denote by $f(X)$ the m-tuple obtained by applying f componentwise to $\mathbf{x}^1, \dots, \mathbf{x}^k$, i.e.,

$$f(X) = f\left(\mathbf{x}^1, \dots, \mathbf{x}^k\right) = \left(f\left(\mathbf{y}^1\right), \dots, f\left(\mathbf{y}^m\right)\right)\,. \tag{3.38}$$

Lemma 3.33. *Let* Γ *be a constraint language,* $k \geq 1$*, and* H'_k *a Hilbert space of functions* $\mathrm{Pol}^{(k)}(\Gamma) \to \mathbb{R}$ *with an inner product analogous to* (3.31)*. For any* $\gamma \in \Gamma$ *and* $X \in (\mathrm{Feas}(\gamma))^k$*, we define* $\gamma[X] \in H'_k$ *by* $\gamma[X](f) = \gamma(f(X))$ *for all* $f \in \mathrm{Pol}^{(k)}(\Gamma)$*. Let*

$$K = \left\{ \gamma[X] \;\middle|\; \gamma \in \Gamma \ \wedge \ X \in (\mathrm{Feas}(\gamma))^k \right\} . \tag{3.39}$$

A k*-ary weighting* ω *with* $\mathrm{supp}(\omega) \subseteq \mathrm{Pol}(\Gamma)$ *is a weighted polymorphism of* Γ *if and only if* $\omega|_{\mathrm{Pol}^{(k)}(\Gamma)} \in K^\circ$*.*

Proof. For any $\gamma \in \Gamma$ and $X = (\mathbf{x}_1, \ldots, \mathbf{x}_k) \in (\mathrm{Feas}(\gamma))^k$, (3.32) is equivalent to

$$\left\langle \omega|_{\mathrm{Pol}^{(k)}(\{\gamma\})}, \gamma[X] \right\rangle \leq 0 \,. \tag{3.40}$$

□

Lemma 3.34. *For any constraint language* Γ*, set* $\mathrm{wPol}_{\mathbb{R}}(\Gamma)$ *is a weighted clone.*

Proof. Let $k \geq 1$ be a fixed arity. Any k-ary weighted polymorphism ω of Γ satisfies $\mathrm{supp}(\omega) \subseteq \mathrm{Pol}^{(k)}(\Gamma)$, and thus can be equated with its restriction to $\mathrm{Pol}^{(k)}(\Gamma)$. By Lemma 3.33 and Theorem 3.31, set $\mathrm{wPol}^{(k)}_{\mathbb{R}}(\Gamma)$ is closed under scaling by non-negative real constants, closed under addition, and topologically closed.

It remains to show that $\mathrm{wPol}_{\mathbb{R}}(\Gamma)$ is closed under superposition. Let $\omega \in \mathrm{wPol}_{\mathbb{R}}(\Gamma)$ be a k-ary weighting and $g_1, \ldots, g_k \in \mathrm{supp}(\mathrm{wPol}_{\mathbb{R}}(\Gamma)) \subseteq \mathrm{Pol}(\Gamma)$ be ℓ-ary operations. For any $\gamma \in \Gamma$ and $X \in (\mathrm{Feas}(\gamma))^\ell$, let $Y = (g_1(X), \ldots, g_k(X))$. It holds $Y \in (\mathrm{Feas}(\gamma))^k$ and

$$\begin{aligned}
& \sum_{f \in \mathrm{supp}(\omega[g_1, \ldots, g_k])} \omega[g_1, \ldots, g_k](f) \cdot \gamma(f(X)) && (3.41)\\
&= \sum_{f \in \mathrm{supp}(\omega)} \omega(f) \cdot \gamma(f[g_1, \ldots, g_k](X)) && (3.42)\\
&= \sum_{f \in \mathrm{supp}(\omega)} \omega(f) \cdot \gamma(f(Y)) && (3.43)\\
&\leq 0 \,. && (3.44)
\end{aligned}$$

Therefore, weighting $\omega[g_1, \ldots, g_k]$ is a weighted polymorphism of Γ. □

Lemma 3.35. *For any set of weightings* Ω*,* $\mathrm{Imp}_{\mathbb{R}}(\Omega)$ *is a weighted relational clone.*

Proof. Both $\rho_=$ and $\rho_\emptyset$ are improved by any weighting and hence belong to $\mathrm{Imp}_{\mathbb{R}}(\Omega)$. Addition, coordinate mapping, non-negative scaling, and addition of a constant preserve (3.32), and therefore $\mathrm{Imp}_{\mathbb{R}}(\Omega)$ is closed under these operations.

We need to prove that $\mathrm{Imp}_{\mathbb{R}}(\Omega)$ is closed under minimisation. Let $\gamma \in \mathrm{Imp}_{\mathbb{R}}(\Omega)$ be an r-ary weighted relation and consider an $(r-1)$-ary weighted relation γ' obtained by minimising γ at some coordinate. Let $\omega \in \Omega$ be a k-ary weighting and $X' = (\mathbf{x}'_1, \ldots, \mathbf{x}'_k) \in (\mathrm{Feas}(\gamma'))^k$. For any $i \in [k]$, we can extend $(r-1)$-tuple $\mathbf{x}'_i$ to an r-tuple $\mathbf{x}_i \in \mathrm{Feas}(\gamma)$ so that $\gamma'(\mathbf{x}'_i) = \gamma(\mathbf{x}_i)$; we denote the list of these extended r-tuples by $X = (\mathbf{x}_1, \ldots, \mathbf{x}_k)$. Note that $f(X)$ is an extension of $f(X')$ for any k-ary operation $f \in \mathrm{supp}(\omega)$, and therefore $\gamma'(f(X')) \leq \gamma(f(X))$. Moreover, $\gamma'(f(X')) = \gamma(f(X))$ whenever f is a projection. As γ satisfies (3.32) and only projections may be assigned a negative weight, we have

$$\sum_{f \in \mathrm{supp}(\omega)} \omega(f) \cdot \gamma'(f(X')) \leq \sum_{f \in \mathrm{supp}(\omega)} \omega(f) \cdot \gamma(f(X)) \leq 0\,, \tag{3.45}$$

and thus $\gamma' \in \mathrm{Imp}_{\mathbb{R}}(\Omega)$.

Now we prove that $\mathrm{Imp}_{\mathbb{R}}(\Omega)$ is closed under operation Opt. Let $\gamma \in \mathrm{Imp}_{\mathbb{R}}(\Omega)$ and $\rho = \mathrm{Opt}(\gamma)$. We will assume that $\mathrm{Feas}(\gamma)$ is non-empty (otherwise $\gamma = \rho$) and denote by c the minimum weight assigned by γ. Let $\omega \in \Omega$ be a k-ary weighting. As ρ is a relation, we only need to show that all operations in the support of ω are polymorphisms of ρ. Let $X \in (\mathrm{Feas}(\rho))^k \subseteq (\mathrm{Feas}(\gamma))^k$. For any operation f in the support of ω, it holds $\gamma(f(X)) \geq c$. Moreover, $\gamma(f(X)) = c$ whenever f is a projection. We have

$$0 \geq \sum_{f \in \mathrm{supp}(\omega)} \omega(f) \cdot \gamma(f(X)) \geq \sum_{f \in \mathrm{supp}(\omega)} \omega(f) \cdot c = 0\,, \tag{3.46}$$

which for all $f \in \mathrm{supp}(\omega)$ implies $\gamma(f(X)) = c$ and hence $f(X) \in \mathrm{Feas}(\rho)$. Therefore, $\rho \in \mathrm{Imp}_{\mathbb{R}}(\Omega)$.

Finally, we show that $\mathrm{Imp}_{\mathbb{R}}(\Omega)$ is topologically closed. Let r be a fixed arity and $F \subseteq D^r$; we claim that the set Γ of r-ary weighted relations γ with $\mathrm{Feas}(\gamma) = F$ which are *not improved* by Ω is an open set. Take any $\gamma \in \Gamma$. There must be a non-zero weighting $\omega \in \Omega$ (let us denote its arity by k) and $X \in F^k$ such that $\langle \omega, \gamma[X] \rangle = d$ for some positive d, i.e., ω violates (3.32) for γ and X. Then, for every r-ary weighted relation γ' with $\mathrm{Feas}(\gamma') = F$ and distance from γ less than

$$\frac{d}{\displaystyle\sum_{f \in \mathrm{supp}(\omega)} |\omega(f)|}\,, \tag{3.47}$$

it holds $\langle \omega, \gamma'[X] \rangle > 0$. Therefore, γ has a neighbourhood contained in Γ. $\square$

Theorem 3.36. *For any constraint language* Γ, *it holds*

$$\mathrm{Imp}_{\mathbb{R}}(\mathrm{wPol}_{\mathbb{R}}(\Gamma)) = \mathrm{wRelClone}_{\mathbb{R}}(\Gamma)\,. \tag{3.48}$$

Proof. By the fact that $\Gamma \subseteq \mathrm{Imp}_{\mathbb{R}}(\mathrm{wPol}_{\mathbb{R}}(\Gamma))$ and Lemma 3.35, it holds

$$\mathrm{wRelClone}_{\mathbb{R}}(\Gamma) \subseteq \mathrm{wRelClone}_{\mathbb{R}}(\mathrm{Imp}_{\mathbb{R}}(\mathrm{wPol}_{\mathbb{R}}(\Gamma))) = \mathrm{Imp}_{\mathbb{R}}(\mathrm{wPol}_{\mathbb{R}}(\Gamma))\,. \tag{3.49}$$

Now we prove the converse inclusion $\mathrm{Imp}_{\mathbb{R}}(\mathrm{wPol}_{\mathbb{R}}(\Gamma)) \subseteq \mathrm{wRelClone}_{\mathbb{R}}(\Gamma)$. Let $\rho \in \mathrm{Imp}_{\mathbb{R}}(\mathrm{wPol}_{\mathbb{R}}(\Gamma))$ be a weighted relation and denote $|\,\mathrm{Feas}(\rho)|$ by k. If $k = 0$, ρ is expressible from $\rho_{\emptyset}$ and hence $\rho \in \mathrm{wRelClone}_{\mathbb{R}}(\Gamma)$. Otherwise, we will focus solely on the k-ary weighted polymorphisms of Γ. Let us denote the Hilbert space of functions $\mathrm{Pol}^{(k)}(\Gamma) \to \mathbb{R}$ by H; the k-ary weighted polymorphisms of Γ can be then seen as vectors from H. Denoting $m = |D|^k$, a k-ary operation on D is uniquely determined by the m-tuple of labels it assigns to its m possible inputs. Later we will define a correspondence between a subset of m-ary weighted relations and H.

The outline of the proof is as follows. We transform Γ into a set $M \subseteq \mathrm{wRelClone}_{\mathbb{R}}(\Gamma)$ of m-ary weighted relations and consider their corresponding vectors in H. The polar cone of these vectors equals $\mathrm{wPol}^{(k)}_{\mathbb{R}}(\Gamma)$, and its polar cone, in turn, consists of m-ary weighted relations improved by $\mathrm{wPol}^{(k)}_{\mathbb{R}}(\Gamma)$. We know that the polar cone of the polar cone of a set is the closure of the smallest convex cone containing this set; therefore, any m-ary weighted relation improved by $\mathrm{wPol}^{(k)}_{\mathbb{R}}(\Gamma)$ belongs to $\mathrm{wRelClone}_{\mathbb{R}}(\Gamma)$. This also includes a particular m-ary weighted relation that we use to express ρ, so we get $\rho \in \mathrm{wRelClone}_{\mathbb{R}}(\Gamma)$.

We begin by formally defining the correspondence between certain m-ary weighted relations and vectors from H. Let us denote by $(\mathbf{z}_1, \dots, \mathbf{z}_m) = Z^T$ the sequence of all k-tuples over D in an arbitrary fixed order; any k-ary operation f is then determined by the m-tuple $(f(\mathbf{z}_1), \dots, f(\mathbf{z}_m)) = f(Z)$. Let us define a set $F \subseteq D^m$ by

$$F = \left\{ f(Z) \;\middle|\; f \in \mathrm{Pol}^{(k)}(\Gamma) \right\}\,. \tag{3.50}$$

For any m-ary weighted relation γ with $\mathrm{Feas}(\gamma) = F$, the corresponding vector in H is $\gamma[Z]$. Conversely, for any vector in H there is a corresponding m-ary weighted relation with finite weights precisely on F.

Now we transform Γ into a set of m-ary weighted relations that captures enough information to reconstruct the set of k-ary weighted polymorphisms of Γ.

Claim 1. *For any n-ary weighted relation $\gamma \in \Gamma$ and $X \in (\mathrm{Feas}(\gamma))^k$, there is an m-ary weighted relation $\mu_{\gamma,X} \in \mathrm{wRelClone}_{\mathbb{R}}(\Gamma)$ such that* $\mathrm{Feas}(\mu_{\gamma,X}) = F$ *and $\mu_{\gamma,X}(f(Z)) = \gamma(f(X))$ for all $f \in \mathrm{Pol}^{(k)}(\Gamma)$.*

Proof. We denote the k-tuples of X^T by $(\mathbf{x}_1, \ldots, \mathbf{x}_n)$. First, we construct an m-ary weighted relation $\mu'_{\gamma,X}$ as

$$\mu'_{\gamma,X}(y_{\mathbf{z}_1}, \ldots, y_{\mathbf{z}_m}) = \gamma(y_{\mathbf{x}_1}, \ldots, y_{\mathbf{x}_n}), \tag{3.51}$$

where $y_{\mathbf{z}_i}$ are variables indexed by k-tuples over D. It holds

$$\mu'_{\gamma,X}(f(Z)) = \mu'_{\gamma,X}(f(\mathbf{z}_1), \ldots, f(\mathbf{z}_m)) = \gamma(f(\mathbf{x}_1), \ldots, f(\mathbf{x}_n)) = \gamma(f(X)). \tag{3.52}$$

However, we are not done yet, as $\mu'_{\gamma,X}$ assigns a finite weight to all m-tuples $f(Z)$ such that $f(X) \in \mathrm{Feas}(\gamma)$, even if $f \notin \mathrm{Pol}^{(k)}(\Gamma)$. We can easily fix this: Let f be a k-ary operation that is not a polymorphism of Γ; then there is a weighted relation $\gamma_f \in \Gamma$ and $X_f \in (\mathrm{Feas}(\gamma_f))^k$ such that $f(X_f) \notin \mathrm{Feas}(\gamma_f)$. Adding $\mathrm{Feas}(\mu'_{\gamma_f,X_f})$ to $\mu'_{\gamma,X}$ ensures that the weighted relation will assign infinity to m-tuple $f(Z)$ without changing other weights. This can be done for all (finitely many) such operations f, so we obtain a weighted relation $\mu_{\gamma,X}$ with $\mathrm{Feas}(\mu_{\gamma,X}) = F$. ■

Claim 2. *There are m-ary weighted relations $\mu_\iota, \mu_{-\iota} \in \mathrm{wRelClone}_{\mathbb{R}}(\Gamma)$ such that* $\mathrm{Feas}(\mu_\iota) = \mathrm{Feas}(\mu_{-\iota}) = F$ *and $\mu_\iota(f(Z)) = 1$, $\mu_{-\iota}(f(Z)) = -1$ for all $f \in \mathrm{Pol}^{(k)}(\Gamma)$.*

Proof. Again, we start with $\mu'_\iota(y_{\mathbf{z}_1}, \ldots, y_{\mathbf{z}_m}) = 1$, $\mu'_{-\iota}(y_{\mathbf{z}_1}, \ldots, y_{\mathbf{z}_m}) = -1$ and then add $\mathrm{Feas}(\mu'_{\gamma_f,X_f})$ for all k-ary operations $f \notin \mathrm{Pol}^{(k)}(\Gamma)$ to ensure that the resulting weighted relations $\mu_\iota, \mu_{-\iota}$ assign finite weights only to m-tuples from F. ■

Let $\iota \in H$ be the vector assigning every operation value 1. For any operation $f \in \mathrm{Pol}^{(k)}(\Gamma)$ that is not a projection, let $\varepsilon_f \in H$ be the vector such that $\varepsilon_f(f) = 1$ and $\varepsilon_f(g) = 0$ for all $g \neq f$. We define a set of m-ary weighted relations $M \subseteq \mathrm{wRelClone}_{\mathbb{R}}(\Gamma)$, the set of corresponding vectors $V \subseteq H$, and an auxiliary set of vectors $W \subseteq H$ as follows:

$$M = \left\{\mu_{\gamma,X} \;\middle|\; \gamma \in \Gamma \;\wedge\; X \in (\mathrm{Feas}(\gamma))^k\right\} \cup \{\mu_\iota, \mu_{-\iota}\} \tag{3.53}$$

$$V = \left\{\gamma[X] \;\middle|\; \gamma \in \Gamma \;\wedge\; X \in (\mathrm{Feas}(\gamma))^k\right\} \cup \{\iota, -\iota\} \tag{3.54}$$

$$W = V \cup \left\{-\varepsilon_f \;\middle|\; f \in \mathrm{Pol}^{(k)}(\Gamma) \setminus \Pi_D^{(k)}\right\}. \tag{3.55}$$

Claim 3. *The polar cone W° consists of k-ary weighted polymorphisms of Γ.*

Proof. Let $\omega \in W^\circ$ be a vector. As $\langle \omega, \iota \rangle \leq 0$ and $\langle \omega, -\iota \rangle \leq 0$, we have $\langle \omega, \iota \rangle = 0$, i.e., the sum of weights of ω equals 0. For any non-projection f, it holds $\langle \omega, -\varepsilon_f \rangle \leq 0$, i.e., $\omega(f)$ is non-negative. Finally, for any $\gamma \in \Gamma$ and $X \in (\mathrm{Feas}(\gamma))^k$, it holds $\langle \omega, \gamma[X] \rangle \leq 0$; hence ω is a weighted polymorphism of Γ. ■

Let us now return to weighted relation ρ and denote the sequence of elements of $\mathrm{Feas}(\rho)$ in an arbitrary fixed order by $R \in (\mathrm{Feas}(\rho))^k$. As ρ is improved by $\mathrm{wPol}_\mathbb{R}(\Gamma)$, any vector $\omega \in W^\circ$ satisfies (3.32) for ρ and any $X \in (\mathrm{Feas}(\rho))^k$, in particular for $X = R$. Hence, we would like to claim that $\langle \omega, \rho[R] \rangle \leq 0$ for all $\omega \in W^\circ$ and thus $\rho[R] \in W^{\circ\circ}$. However, $\rho[R]$ might be ill-defined: Although $\rho \in \mathrm{Imp}_\mathbb{R}(\mathrm{wPol}_\mathbb{R}(\Gamma))$, not necessarily all operations $f \in \mathrm{Pol}^{(k)}(\Gamma)$ are polymorphisms of ρ, and therefore possibly $\rho(f(R)) = \infty$. Let us denote the set of these problematic operations by

$$Q = \left\{ f \in \mathrm{Pol}^{(k)}(\Gamma) \;\middle|\; f(R) \notin \mathrm{Feas}(\rho) \right\} . \tag{3.56}$$

On the other hand, every operation in the support of $\mathrm{wPol}^{(k)}_\mathbb{R}(\Gamma)$ is a polymorphism of ρ. This implies that operations in Q must be assigned a zero weight by all $\omega \in W^\circ$. As $\rho[R]$ might not exist, let us define instead a substitute vector $\beta \in H$ such that $\beta(f) = \rho(f(R))$ for all $f \in \mathrm{Pol}^{(k)}(\Gamma) \setminus Q$, with arbitrary values assigned to operations in Q. By the previous argument, $\beta \in W^{\circ\circ}$. Additionally, let $\beta_0 \in H$ be a vector such that $\beta_0(f) > 0$ if $f \in Q$, and $\beta_0(f) = 0$ otherwise. For any $\omega \in W^\circ$ it holds $\langle \omega, \beta_0 \rangle = 0$, so β_0 also belongs to $W^{\circ\circ}$.

Any vector in $W^{\circ\circ}$ can be obtained from some vector in $V^{\circ\circ}$ by adding non-negative multiples of $-\varepsilon_f$ for $f \in \mathrm{Pol}^{(k)}(\Gamma) \setminus \Pi^{(k)}_D$. Therefore, there is a vector $\alpha \in V^{\circ\circ}$ such that $\alpha(f) \geq \beta(f) = \rho(f(R))$ for all $f \notin Q$, and $\alpha(f) = \beta(f) = \rho(f(R))$ when f is a projection. Also, there is a non-negative vector $\alpha_0 \in V^{\circ\circ}$ such that $\alpha_0(f) \geq \beta_0(f) > 0$ if $f \in Q$, and $\alpha_0(f) = \beta_0(f) = 0$ if f is a projection.

Vectors in V correspond to weighted relations in $M \subseteq \mathrm{wRelClone}_\mathbb{R}(\Gamma)$. Set $V^{\circ\circ}$ is the closure of the smallest convex cone containing V, and therefore weighted relations corresponding to vectors in $V^{\circ\circ}$ also belong to $\mathrm{wRelClone}_\mathbb{R}(\Gamma)$ (as it is closed under addition and non-negative scaling, and is topologically closed). Hence, there are m-ary weighted relations $\psi, \psi_0 \in \mathrm{wRelClone}_\mathbb{R}(\Gamma)$ with $\mathrm{Feas}(\psi) = \mathrm{Feas}(\psi_0) = F$ such that $\psi[Z] = \alpha$ and $\psi_0[Z] = \alpha_0$. We are going to express ρ from them.

Let us denote the arity of ρ by n and the k-tuples of R^T by $(\mathbf{r}_1, \ldots, \mathbf{r}_n)$. Consider the following gadget. Let I be a $\mathrm{VCSP}_\mathbb{R}$ instance with variables $y_{\mathbf{z}_1}, \ldots, y_{\mathbf{z}_m}$ and a single constraint $\psi(y_{\mathbf{z}_1}, \ldots, y_{\mathbf{z}_m})$, and let $L = (y_{\mathbf{r}_1}, \ldots, y_{\mathbf{r}_n})$.

Then $\pi_L(I)$ is an n-ary weighted relation expressible over $\mathrm{wRelClone}_{\mathbb{R}}(\Gamma)$; we will denote it by ρ'. For any n-tuple $\mathbf{x} \in D^n$, we have

$$\begin{aligned}
\rho'(\mathbf{x}) &= \min_{\{(y_{\mathbf{z}_1},\dots,y_{\mathbf{z}_m})\in D^m \;\mid\; (y_{\mathbf{r}_1},\dots,y_{\mathbf{r}_n})=\mathbf{x}\}} \psi(y_{\mathbf{z}_1},\dots,y_{\mathbf{z}_m}) && (3.57)\\
&= \min_{\{f:D^k\to D \;\mid\; (f(\mathbf{r}_1),\dots,f(\mathbf{r}_n))=\mathbf{x}\}} \psi(f(\mathbf{z}_1),\dots,f(\mathbf{z}_m)) && (3.58)\\
&= \min_{f(R)=\mathbf{x}} \psi(f(Z)) = \min_{f(R)=\mathbf{x}} \alpha(f)\,. && (3.59)
\end{aligned}$$

Analogously, by replacing ψ with ψ_0 in the gadget we obtain an n-ary weighted relation ρ'_0 for which $\rho'_0(\mathbf{x}) = \min_{f(R)=\mathbf{x}} \alpha_0(f)$.

For any $\mathbf{x} \in \mathrm{Feas}(\rho)$ and k-ary operation f such that $f(R) = \mathbf{x}$, it holds $\alpha(f) \geq \rho(f(R)) = \rho(\mathbf{x})$. As R is a list of all elements of $\mathrm{Feas}(\rho)$, there is a projection f such that $f(R) = \mathbf{x}$; for it we have $\alpha(f) = \rho(f(R)) = \rho(\mathbf{x})$. Therefore, $\rho'(\mathbf{x}) = \rho(\mathbf{x})$. Similarly we get $\rho'_0(\mathbf{x}) = 0$ for any $\mathbf{x} \in \mathrm{Feas}(\rho)$.

We are almost done; the last issue is that $\rho'(\mathbf{x})$ may be finite also for some $\mathbf{x} \notin \mathrm{Feas}(\rho)$. But $f(R) \notin \mathrm{Feas}(\rho)$ implies $f \in Q$, and in that case $\alpha_0(f)$ is positive. Therefore, $\mathrm{Opt}(\rho'_0)$ is finite only on $\mathrm{Feas}(\rho)$, and $\rho' + \mathrm{Opt}(\rho'_0) = \rho$. ☐

In the context of Min-Sol-Hom and Min-Cost-Hom [103], the following corollary has been observed by Hannes Uppman.[2]

Corollary 3.37. *Let* Γ *be a weighted relational clone. Then* $\mathrm{supp}(\mathrm{wPol}_{\mathbb{R}}(\Gamma)) = \mathrm{Pol}(\Gamma)$.

Proof. As $\mathrm{wPol}_{\mathbb{R}}(\Gamma)$ is a weighted clone (Lemma 3.34), we get

$$\mathrm{supp}(\mathrm{wPol}_{\mathbb{R}}(\Gamma)) = \mathrm{Pol}(\mathrm{Imp}_{\mathbb{R}}(\mathrm{wPol}_{\mathbb{R}}(\Gamma))) = \mathrm{Pol}(\mathrm{wRelClone}_{\mathbb{R}}(\Gamma)) = \mathrm{Pol}(\Gamma) \tag{3.60}$$

by Lemma 3.29 and Theorem 3.36. ☐

Theorem 3.38. *For any set of weightings* Ω*, it holds*

$$\mathrm{wPol}_{\mathbb{R}}(\mathrm{Imp}_{\mathbb{R}}(\Omega)) = \mathrm{wClone}_{\mathbb{R}}(\Omega)\,. \tag{3.61}$$

Proof. By the fact that $\Omega \subseteq \mathrm{wPol}_{\mathbb{R}}(\mathrm{Imp}_{\mathbb{R}}(\Omega))$ and Lemma 3.34, it holds

$$\mathrm{wClone}_{\mathbb{R}}(\Omega) \subseteq \mathrm{wClone}_{\mathbb{R}}(\mathrm{wPol}_{\mathbb{R}}(\mathrm{Imp}_{\mathbb{R}}(\Omega))) = \mathrm{wPol}_{\mathbb{R}}(\mathrm{Imp}_{\mathbb{R}}(\Omega))\,. \tag{3.62}$$

We prove that, for any $k \geq 1$ and any k-ary weighting $\mu \in \mathrm{wPol}_{\mathbb{R}}(\mathrm{Imp}_{\mathbb{R}}(\Omega))$, it holds $\mu \in \mathrm{wClone}_{\mathbb{R}}(\Omega)$. First, let us establish the clone of operations we will be working with. Let C be the smallest clone containing $\mathrm{supp}(\Omega)$. The

---

[2] Private communication, 2014.

support of $\mathrm{wClone}_{\mathbb{R}}(\Omega)$ is itself a clone (by Lemma 3.28) so we also have $C = \mathrm{supp}(\mathrm{wClone}_{\mathbb{R}}(\Omega))$. As in the proof of Theorem 3.36, we will represent k-ary weightings by vectors of the Hilbert space $H = C^{(k)} \to \mathbb{R}$, and identify those vectors with certain m-ary weighted relations (where $m = |D|^k$).

The outline of the proof is as follows. We transform Ω into a set W of k-ary weightings. Although some of these weightings may be improper, any proper weighting obtained as their non-negative linear combination belongs to $\mathrm{wClone}_{\mathbb{R}}(\Omega)$. The polar cone W° consists of m-ary weighted relations improved by Ω, and its polar cone $W^{\circ\circ}$ hence contains μ. As the polar cone of the polar cone of a set is the closure of the smallest convex cone containing this set, we get $\mu \in \mathrm{wClone}_{\mathbb{R}}(\Omega)$.

Recall the correspondence between a subset of m-ary weighted relations and H from the proof of Theorem 3.36. This time, we are working with clone C, so we define F by

$$F = \left\{ f(Z) \;\middle|\; f \in C^{(k)} \right\} . \tag{3.63}$$

Let Γ be the set of all m-ary weighted relations γ with $\mathrm{Feas}(\gamma) = F$. Similarly as before, there is a bijection between Γ and H: the corresponding vector to a weighted relation $\gamma \in \Gamma$ is $\gamma[Z]$.

Claim 1. *For any $\gamma \in \Gamma$, it holds $\mathrm{Pol}(\{\gamma\})^{(k)} = C^{(k)}$.*

Proof. Let $f \in C^{(k)}$. For any $X \in F^k$, there are k-ary operations $g_1, \ldots, g_k \in C^{(k)}$ such that $X = (g_1(Z), \ldots, g_k(Z))$. So we have $f(X) = f[g_1, \ldots, g_k](Z) \in F$ because $f[g_1, \ldots, g_k] \in C^{(k)}$.

Conversely, let $f \in \mathcal{O}_D^{(k)} \setminus C^{(k)}$. Certainly $Z = (\pi_1^{(k)}(Z), \ldots, \pi_k^{(k)}(Z)) \in F^k$, but $f(Z) \notin F$. Therefore, f is not a polymorphism of γ. ■

Let us define a set $W \subseteq H$ by

$$W = \left\{ \omega[g_1, \ldots, g_\ell] \in H \;\middle|\; \ell \geq 1 \;\wedge\; \omega \in \Omega^{(\ell)} \;\wedge\; g_1, \ldots, g_\ell \in C^{(k)} \right\} . \tag{3.64}$$

Claim 2. *For any $\gamma \in \Gamma$ such that $\gamma[Z] \in W^\circ$, it holds $\Omega \subseteq \mathrm{wPol}_{\mathbb{R}}(\{\gamma\})$.*

Proof. Let $\omega \in \Omega$ be an ℓ-ary weighting, and $X \in F^\ell$. Then there are k-ary operations $g_1, \ldots, g_\ell \in C^{(k)}$ for which $X = (g_1(Z), \ldots, g_\ell(Z))$, and we have

$$\sum_{f \in \mathrm{supp}(\omega)} \omega(f) \cdot \gamma(f(X)) = \sum_{f \in C^{(\ell)}} \omega(f) \cdot \gamma(f[g_1, \ldots, g_\ell](Z)) \tag{3.65}$$

$$= \sum_{f \in C^{(k)}} \omega[g_1, \ldots, g_\ell](f) \cdot \gamma(f(Z)) \tag{3.66}$$

$$= \langle \omega[g_1, \ldots, g_\ell], \gamma[Z] \rangle \leq 0 . \tag{3.67}$$

■

Weighting μ is a weighted polymorphism of $\mathrm{Imp}_{\mathbb{R}}(\Omega)$, so it improves any weighted relation γ corresponding to a vector $\gamma[Z] \in W^{\circ}$. Firstly, this implies $\mathrm{supp}(\mu) \subseteq C^{(k)}$; we can therefore view μ as a vector of H. Secondly, μ satisfies (3.32) for γ and any $X \in F^k$. In particular, $Z \in F^k$, so we get $\langle \mu, \gamma[Z] \rangle \leq 0$ and thus $\mu \in W^{\circ\circ}$.

Set $W^{\circ\circ}$ is the closure of the smallest convex cone containing W. By Lemma 3.27, any proper weighting obtained as a non-negative linear combination of weightings from W belongs to $\mathrm{wClone}_{\mathbb{R}}(\Omega)$. Therefore, $\mu \in \mathrm{wClone}_{\mathbb{R}}(\Omega)$. $\square$

Chapter 4

Planar VCSP

The maximum cut problem can be expressed in the VCSP framework on a Boolean domain using a single weighted relation $\gamma_{\neq}$ (see Example 2.10). Because the problem is NP-hard [54], language $\{\gamma_{\neq}\}$ is intractable. However, the maximum cut problem can be solved in polynomial time on planar graphs by a reduction to the maximum matching problem [59]. In this chapter, we study the VCSP under a structural restriction of planarity, which reduces the computational complexity of some constraint languages (including $\{\gamma_{\neq}\}$).

The structural restriction concerns the incidence graph of an instance. This is a bipartite graph, in which the two parts of vertices represent the variables and the constraints of the instance respectively, and edges connect constraints with the variables in their scopes. We require that the incidence graph be planar and, moreover, that it can be drawn in a way that respects the order of arguments of the constraints; this stronger notion of planarity leads to a finer complexity classification of languages (see Definition 4.1). We denote the VCSP restricted to such instances over a language Γ by $\mathrm{VCSP_p}(\Gamma)$.

Another example of an intractable Boolean language that becomes tractable under the planarity restriction (we say it is *p-tractable*) corresponds to a variant of the 3-Sat problem called the NAE-3-Sat (Not-All-Equal 3-Satisfiability) problem, in which a clause is satisfied if and only if its three variables are not assigned the same label (see Example 4.2). Again, the p-tractability was established by a reduction to the maximum matching problem [83]. On the other hand, there exist intractable languages that are also p-intractable, i.e., the associated VCSP is NP-hard even on planar instances. For example, the minimum vertex cover problem, the 3-Sat problem itself and also its variant called the 1-in-3 Positive 3-Sat problem (see Example 4.3) are all p-intractable [55, 80, 84]. For an example on a larger domain, consider graph colouring with k colours, which corresponds to language $\{\rho_{\neq}\}$ on a domain of size k (see Example 2.9). In the case of $k = 3$, the problem is NP-hard even on planar graphs [54]. However, it becomes p-tractable if $k \geq 4$; namely, the four colour theorem guarantees that a 4-colouring exists for any planar graph, and a solution can be found in polynomial time [90].

Considering that a full complexity classification of languages on larger domains would involve difficult special cases such as the graph colouring problem, it is natural to focus on Boolean languages first. Dvořák and Kupec [41] studied the planar Boolean CSP (i.e., the case of $\{0, \infty\}$-valued languages) and identified a necessary condition for an intractable language to be p-tractable. The condition is stated in two parts: First, the language must be self-complementary, i.e., swapping the two labels at all coordinates simultaneously must preserve the feasibility of any assignment. Equivalently, the language must be invariant under operation $\neg$ (the unary negation). As Dvořák and Kupec showed, the planar CSP over a self-complementary language Γ can be reduced to an instance over a related language $\mathrm{diff}(\Gamma)$[1] in which every variable participates in at most two constraints. This structural restriction has been studied under the name *fanout limitation* [45]; here we adopt instead the term *edge* CSP proposed in [69]. The second part of the necessary condition requires that $\mathrm{diff}(\Gamma)$ consist of even delta-matroids, which are a type of relation generalising matroids [10, 22, 39]. The condition was proved to be sufficient for p-tractability by Kazda, Kolmogorov, and Rolínek [69], who designed a polynomial-time algorithm for the edge CSP over even delta-matroids, and thus established a complexity classification of Boolean $\{0, \infty\}$-valued languages with respect to p-tractability. It is interesting to note that their algorithm resembles in many aspects the classic blossom algorithm for finding a maximum matching [43] and matroid partitioning algorithm [42].

In Section 4.2, we generalise the necessary condition of Dvořák and Kupec to all ($\overline{\mathbb{Q}}$-valued) Boolean languages. In particular, we show that an intractable language must satisfy the following two requirements in order to be p-tractable. First, the language must be self-complementary in a stronger (valued) sense: Swapping the two labels at all coordinates simultaneously must preserve the value of any assignment, or, equivalently, the language must admit multimorphism $\langle \neg \rangle$. Second, $\mathrm{diff}(\Gamma)$ must consist of valuated delta-matroids, which are a type of weighted relation generalising weighted matroids [40]. We conjecture that the edge VCSP over valuated delta-matroids can be efficiently solved by a generalisation of the algorithm from [69], and hence our necessary condition is also sufficient for p-tractability.

Our second result concerns conservative languages (i.e. languages con-

---

[1] Informally, operation diff eliminates redundancy in self-complementary languages by mapping tuples $\mathbf{x}$ and $\neg(\mathbf{x})$ to a single one.

taining all $\{0,1\}$-valued unary weighted relations) on any finite domain. Kolmogorov and Živný [73] obtained a complexity classification of all conservative languages, which characterises tractable cases by the existence of a certain pair of multimorphisms (see Theorem 4.43). In Section 4.3, we establish a complexity classification of all conservative languages with respect to p-tractability that matches the classification with respect to tractability. In other words, any intractable conservative language is also p-intractable. We follow the proof in [73], but we simplify and adapt it to the planar setting. For a discussion of the differences between the two proofs, see Section 4.3.

4.1 Preliminaries

First, we formally define the planarity restriction and give a few examples of intractable languages that are p-tractable or p-intractable. Second, we introduce a language closure that preserves p-tractability and is used in our hardness proofs.

4.1.1 Planar VCSP

Definition 4.1. The *incidence graph* of a VCSP instance is an undirected bipartite multigraph, where the two parts of vertices represent the variables and constraints of the instance respectively, and the edge multiset is defined as follows: Each constraint $\gamma_i(x_{i,1},\dots,x_{i,\mathrm{ar}(\gamma_i)})$ is incident to $\mathrm{ar}(\gamma_i)$ edges connecting it to variables $x_{i,1},\dots,x_{i,\mathrm{ar}(\gamma_i)}$.

A *planar* VCSP *instance* is an instance with a planar incidence graph[2] that can be drawn in a way that respects the order of arguments of the constraints.[3] Namely, the edges incident to constraint $\gamma_i(x_{i,1},\dots,x_{i,\mathrm{ar}(\gamma_i)})$ should lead, in clockwise order, to variables $x_{i,1},\dots,x_{i,\mathrm{ar}(\gamma_i)}$. For a language Γ, we denote by $\mathrm{VCSP_p}(\Gamma)$ the VCSP restricted to planar instances over language Γ.

A language Γ is called *p-tractable* if, for every finite $\Gamma' \subseteq \Gamma$, $\mathrm{VCSP_p}(\Gamma')$ can be solved in polynomial time. If there exists a finite $\Gamma' \subseteq \Gamma$ such that $\mathrm{VCSP_p}(\Gamma')$ is NP-hard, language Γ is called *p-intractable*.

---

[2] Another standard representation of a VCSP instance is the *primal graph* (also called the *Gaifman graph*), in which the vertices correspond to the variables and the edges connect all pairs of variables that occur together in the scope of a constraint [35]. We find the incidence graph a more natural choice for the planarity restriction, as restricting the Gaifman graph would disallow, among others, constraints of arity more than four.

[3] This stronger planarity restriction was also assumed in [41]. It leads to a finer classification, as the variant without respecting the order corresponds to languages that are closed under permutation of arguments.

As the following example shows, there exist intractable languages that become tractable under the planarity restriction.

Example 4.2. Let $D = \{0,1\}$. The NAE-3-Sat (Not-All-Equal 3-Satisfiability) problem can be expressed by a ternary relation

$$\rho_{\text{NAE}} = \left\{(x,y,z) \in \{0,1\}^3 \;\middle|\; \{x,y,z\} = \{0,1\}\right\} . \tag{4.1}$$

Language $\{\rho_{\text{NAE}}\}$ is intractable [54], but it is p-tractable [83].

Let $\gamma_{\neq} = \text{Soft}(\rho_{\neq})$ denote the soft variant of the binary disequality relation $\rho_{\neq}$. Language $\{\gamma_{\neq}\}$ encodes the maximum cut problem (see Example 2.10); it is intractable [54] but p-tractable [59].

On the other hand, some languages remain intractable even on planar instances.

Example 4.3. Let $D = \{0,1\}$. The 1-in-3 Positive 3-Sat problem can be expressed by a ternary relation

$$\rho_{\text{1-in-3}} = \{(0,0,1),(0,1,0),(1,0,0)\} . \tag{4.2}$$

Language $\{\rho_{\text{1-in-3}}\}$ is p-intractable [84] and thus also intractable.

Let $\gamma_0 = \text{Soft}(\rho_0)$ denote the soft variant of the unary relation $\rho_0 = \{(0)\}$, and let $\rho_\vee = \{(0,1),(1,0),(1,1)\}$. Then language $\{\gamma_0, \rho_\vee\}$ encodes the weighted minimum vertex cover problem, where the vertices included in a cover are labelled with 1 (thus incurring a non-negative value from constraints $\gamma_0(v)$) and every edge $\{u,v\}$ is represented by a constraint $\rho_\vee(u,v)$. It is a p-intractable language [55], and hence it is also intractable.

We denote by $\rho_\times$ the quaternary relation $\{(a,b,a,b) \mid a,b \in D\}$. It is easy to see that relation $\rho_\times$ can be used to transform any VCSP instance to a planar one by replacing crossings.

Lemma 4.4 ([41, Observation 1])**.** *For any language Γ, it holds*

$$\text{VCSP}(\Gamma) \leq_p \text{VCSP}_\text{p}(\Gamma \cup \{\rho_\times\}) . \tag{4.3}$$

Consider a planar incidence graph. Because it is bipartite, we may represent it by a plane graph in which vertices correspond to variables and faces to constraints [41]. This plane graph is allowed to have parallel edges and self-loops (possibly several at a single vertex).

For a connected plane graph G, we denote by $F(G)$ the set of its faces. For any face $f \in F(G)$, let $b(f)$ denote a closed walk in G bounding f, enumerated in clockwise order around f.

Definition 4.5. Let $I = (V, D, \phi_I)$ be a planar VCSP instance with q constraints. A *plane instance* (I, G, τ) extends I with a connected plane graph G over vertices V and an injective mapping $\tau : [q] \to F(G)$ such that, for every constraint $\gamma_i(x_{i,1}, \dots, x_{i,\mathrm{ar}(\gamma_i)})$, it holds $b(\tau(i)) = x_{i,1} \dots x_{i,\mathrm{ar}(\gamma_i)}$.

Example 4.6. Let I be a VCSP instance with $V = \{x_1, x_2, x_3, x_4\}$ and

$$\phi_I(x_1, x_2, x_3, x_4) = 2{\cdot}\gamma_1(x_1) + 0{\cdot}\gamma_2(x_2, x_3, x_1) + \gamma_3(x_3, x_2) + \frac{5}{3}{\cdot}\gamma_4(x_3, x_4)\,. \quad (4.4)$$

The incidence graph of I (see Figure 4.1(a)) is planar, and hence I is a planar instance. Its extension to a plane VCSP instance is depicted in Figure 4.1(b).

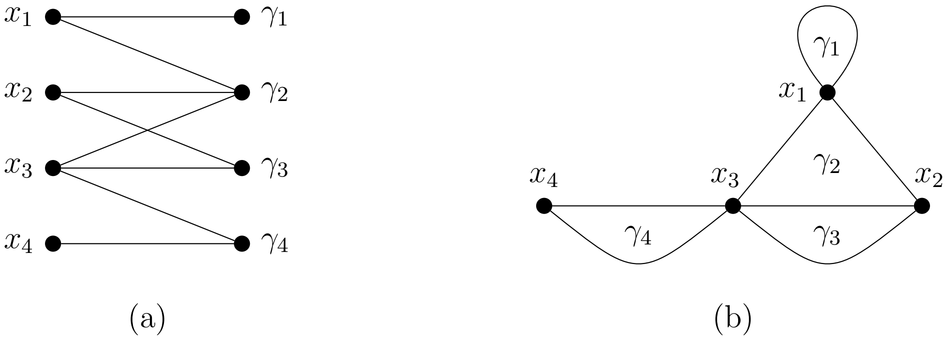

Figure 4.1: Graphs from Example 4.6.

As mentioned in [41], a plane representation of a *planar instance* can be found in polynomial time [62]. We may hence assume that we are given a *plane instance* (i.e., graph G and mapping τ).

4.1.2 Planar weighted relational clones

We define planar weighted relational clones (see Definition 4.13), which are closures of constraint languages that do not change the tractability of the corresponding planar VCSP. Compared with regular weighted relational clones (see Theorems 3.4 and 3.15), the notion of expressibility must be restricted so that it preserves p-tractability (see Theorem 4.15).

Definition 4.7 (cf. Definition 3.2)**.** Let Γ be a constraint language, (I, G, τ) a plane VCSP instance such that $I = (V, D, \phi_I) \in \mathrm{VCSP_p}(\Gamma)$ and τ does not map any constraint to the outer face $f_o \in F(G)$ of G, and let $L = (v_1, \dots, v_r)$ be a list of (not necessarily distinct) variables from V such that

$b(f_o) = v_r v_{r-1} \dots v_1$. The *projection* of I onto L, denoted by $\pi_L(I)$, is the r-ary weighted relation on D defined by

$$\pi_L(I)(x_1, \dots, x_r) = \min \{\phi_I(s) \mid s : V \to D \;\wedge\; s(v_i) = x_i \text{ for all } i \in [r]\} \,, \tag{4.5}$$

where the minimum over the empty set equals ∞.

We say that a weighted relation γ is *p-expressible* over a constraint language Γ if there exist (I, G, τ) and L such that $\gamma = \pi_L(I)$.

Example 4.8 (cf. Example 3.3)**.** Relation $\rho_=$ is p-expressible over any language by a plane instance consisting of a single variable x with two self-loops, and $L = (x, x)$.

Example 4.9. Let (I, G, τ) be a plane VCSP instance depicted in Figure 4.2 with $D = \{0, 1\}$, $V = \{x_1, x_2, x_3, z\}$, and

$$\phi_I(x_1, x_2, x_3, z) = \gamma_{\neq}(x_1, z) + \gamma_{\neq}(x_2, z) + \gamma_{\neq}(x_3, z) \,, \tag{4.6}$$

where $\gamma_{\neq} = \mathrm{Soft}(\rho_{\neq})$ is the soft variant of the binary disequality relation $\rho_{\neq}$. Then $\pi_{(x_1,x_2,x_3)}(I) = \gamma$ is a ternary weighted relation such that $\gamma(x, y, z) = 0$ if $x = y = z$ and $\gamma(x, y, z) = 1$ otherwise. Hence, γ is p-expressible over $\{\gamma_{\neq}\}$.

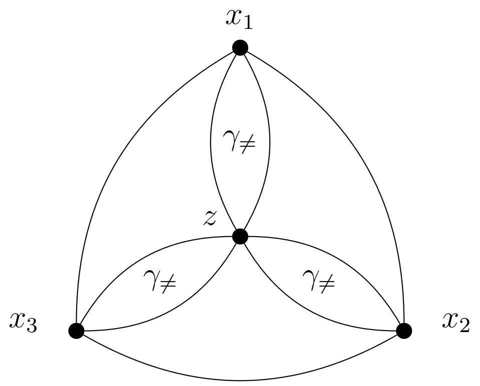

Figure 4.2: Instance from Example 4.9.

Example 4.10. To see that p-expressibility is a proper restriction of regular expressibility from Definition 3.2, note that, for any domain D, relation $\rho_\times$ is expressible from the binary equality relation $\rho_=$, as

$$\rho_\times(x_1, x_2, x_3, x_4) = \rho_=(x_1, x_3) + \rho_=(x_2, x_4) \,. \tag{4.7}$$

However, $\rho_\times$ is not *p-expressible* from $\rho_=$. This can be proved unconditionally but here we give a simpler argument assuming P $\neq$ NP:

By Theorem 4.15 below, p-expressibility preserves p-tractability. Relation $\rho_=$ can be included in any language without affecting its complexity (see Example 4.9). On the other hand, relation $\rho_\times$ enables bypassing the planarity restriction; by Lemma 4.4, languages containing $\rho_\times$ have the same complexity in the planar setting as in general. Consequently, if $\rho_\times$ is p-expressible from $\rho_=$, then every p-tractable language is also tractable; for instance, the NAE-3-SAT problem (see Example 4.2) can be solved in polynomial time even on non-planar instances.

Now we show that minimisation (see Definition 2.4) as well as a few operations defined below are p-expressible.

Definition 4.11. Let γ and μ be an r-ary and a unary weighted relation respectively. The *1-addition* of μ to γ at coordinate $i \in [r]$ results in an r-ary weighted relation γ' such that, for every $\mathbf{x} = (x_1, \ldots, x_r) \in D^r$, it holds $\gamma'(\mathbf{x}) = \gamma(\mathbf{x}) + \mu(x_i)$.

Let γ and μ be an r-ary and a binary weighted relation respectively. The *2-addition* of μ to γ at coordinate $i \in [r]$ results in an r-ary weighted relation γ' such that, for every $\mathbf{x} = (x_1, \ldots, x_r) \in D^r$, it holds $\gamma'(\mathbf{x}) = \gamma(\mathbf{x}) + \mu(x_i, x_{i+1})$, where $x_{r+1} = x_1$.

Let γ be an r-ary weighted relation. The *domain restriction* of γ to $D' \subseteq D$ at coordinate $i \in [r]$ results in an r-ary weighted relation γ' such that, for every $\mathbf{x} = (x_1, \ldots, x_r) \in D^r$, it holds $\gamma'(\mathbf{x}) = \gamma(\mathbf{x})$ if $x_i \in D'$ and $\gamma'(\mathbf{x}) = \infty$ otherwise.

Let $\gamma, \gamma_1, \gamma_2$ be binary weighted relations. We say that γ is a *join* of γ_1 and γ_2 if it can be written as

$$\gamma(x, y) = \min_{z \in D} \left(\gamma_1(u_1, v_1) + \gamma_2(u_2, v_2)\right) , \tag{4.8}$$

where $\{u_1, v_1\} = \{x, z\}$, $\{u_2, v_2\} = \{y, z\}$.

For any $D' \subseteq D$, we denote by $\rho_{D'}$ the unary relation D', i.e., $\rho_{D'}(x) = 0$ if $x \in D'$ and $\rho_{D'}(x) = \infty$ otherwise. Recall from Definition 2.3 that $\rho_{\{a\}} = \rho_a$ for all $a \in D$.

Lemma 4.12. *Minimisation, 1-addition, 2-addition, and join are p-expressible operations over any language. Domain restriction to $D' \subseteq D$ is p-expressible over any language containing $\rho_{D'}$. Operation a-pinning is p-expressible over any language containing ρ_a.*

Proof. The minimisation of an r-ary weighted relation at coordinate i is p-expressible by a cycle $x_1, \dots, x_r$ with an additional edge $\{x_{i-1}, x_{i+1}\}$ in the outer face, which hides vertex x_i.

The 2-addition to an r-ary weighted relation at coordinate i is p-expressible by a cycle $x_1, \dots, x_r$ with a pair of parallel edges $\{x_i, x_{i+1}\}$. In the case of 1-addition, we also add a self-loop at vertex x_i in the face bounded by the pair of edges $\{x_i, x_{i+1}\}$.

A join $\gamma(x, y)$ is p-expressible by three pairs of parallel edges $\{x, z\}$, $\{y, z\}$, $\{x, y\}$ drawn so that the pair of edges $\{x, y\}$ bound the outer face, thus hiding vertex z (similarly as in Figure 4.2).

A domain restriction can be achieved by a 1-addition of $\rho_{D'}$. An a-pinning can be achieved by a domain restriction to $\{a\}$ and subsequent minimisation. □

Definition 4.13 (cf. Theorems 3.4 and 3.15)**.** A constraint language Γ is called a *planar weighted relational clone* if it contains the unary empty relation $\rho_\emptyset$, is closed under p-expressibility, addition of rational constants, and operation Opt.

We define $\mathrm{wRelClone}_\mathrm{p}(\Gamma)$ to be the smallest planar weighted relational clones containing Γ.

Remark 4.14. Note that a planar weighted relational clone also contains the binary equality relation $\rho_=$ and is closed under minimisation, 1-addition, 2-addition, join, scaling by non-negative rational constants, and operation Feas (as $\mathrm{Feas}(\gamma) = 0 \cdot \gamma$).

Theorem 4.15 (cf. Theorems 3.5 and 3.22)**.** *A constraint language* Γ *is p-tractable if and only if* $\mathrm{wRelClone}_\mathrm{p}(\Gamma)$ *is p-tractable, and* Γ *is p-intractable if and only if* $\mathrm{wRelClone}_\mathrm{p}(\Gamma)$ *is p-intractable.*

Proof. Observation 3.16, Observation 3.18, and Lemma 3.19 apply to the planar setting as well; we only need to prove an analogue of Lemma 3.17 for p-expressibility.

Let γ be a weighted relation that is p-expressible using gadget (J, L), where $L = (v_1, \dots, v_r)$, (J, G_J, τ_J) is a plane instance over language Γ, and $f_o \in F(G_J)$ is the outer face of G_J. Given a plane instance (I, G_I, τ_I) over language $\Gamma \cup \{\gamma\}$, we replace every constraint of the form $\gamma(x_1, \dots, x_r)$ in I with a copy of J and add constraints $\rho_=(v_i, x_i)$ exactly as in the proof of Lemma 3.17. This reduces the problem of solving I to an instance I' of $\mathrm{VCSP}(\Gamma \cup \{\rho_=\})$; we claim that the obtained instance I' is in fact planar.

Let $f \in F(G_I)$ be the face assigned by τ_I to a constraint of the form $\gamma(x_1, \dots, x_r)$. Without loss of generality, we may assume that f is *not* the outer face of G_I. As it holds $b(f) = x_1 \dots x_r$ and $b(f_o) = v_r \dots v_1$, plane graph G_J and a pair of parallel edges $\{v_i, x_i\}$ for every $i \in [r]$ can be embedded in face f. Each of these pairs of parallel edges $\{v_i, x_i\}$ bounds a new face, to which the added constraint $\rho_{=}(v_i, x_i)$ can be mapped. Therefore, instance I' is planar.

To eliminate any constraint of the form $\rho_{=}(x, y)$, we replace variables x and y with a new variable z. It is easy to see that this operation preserves planarity of the instance: Constraint $\rho_{=}(x, y)$ is mapped to a face bounded by two $\{x, y\}$ edges. This face can be shrunk to a new vertex z with two additional self-loops in place of the two $\{x, y\}$ edges. □

In order to prove the hardness results for conservative languages in Section 4.3, we do not need the full expressive power of p-expressibility; the following (more restricted) notion of closure is sufficient.

Definition 4.16. A constraint language Γ is called *closed* if it contains all unary weighted relations and the binary equality relation $\rho_{=}$, and is closed under minimisation, 1-addition, join, and operations Feas and Opt.

We define Γ^* to be the smallest closed language containing Γ.

Note that a closed set is also closed under domain restriction and pinning, as these operations can be achieved by 1-addition and minimisation.

Lemma 4.17. *For any conservative language Γ, it holds $\Gamma^* \subseteq \mathrm{wRelClone_p}(\Gamma)$.*

Proof. Let $U \subseteq \Gamma$ denote the set of all $\{0, 1\}$-valued unary weighted relations. Any unary weighted relation can be obtained from U by 1-addition, non-negative scaling, addition of constants, and operator Opt. Hence, set $\mathrm{wRelClone_p}(\Gamma)$ contains all unary weighted relations. The remaining properties follow from Remark 4.14. □

Corollary 4.18. *A conservative constraint language Γ is p-tractable if and only if Γ^* is p-tractable, and Γ is p-intractable if and only if Γ^* is p-intractable.*

Finally, we establish an auxiliary lemma that is used for proving results about both Boolean and conservative languages. Before its statement, we need to define 2-decomposable relations and introduce some notation.

Definition 4.19. Let ρ be an r-ary relation. For any $i, j \in [r]$, we denote by $\mathrm{Pr}_{i,j}(\rho)$ the projection of ρ on coordinates i and j, i.e. the binary relation defined as

$$(a_i, a_j) \in \mathrm{Pr}_{i,j}(\rho) \iff (\exists \mathbf{x} \in \rho)\ x_i = a_i \wedge x_j = a_j\,. \tag{4.9}$$

Relation ρ is *2-decomposable* if

$$\mathbf{x} \in \rho \iff \bigwedge_{1 \le i,j \le r} (x_i, x_j) \in \mathrm{Pr}_{i,j}(\rho)\,. \tag{4.10}$$

Note that all unary and binary relations are 2-decomposable.

For any r-tuple $\mathbf{z}$, we denote its ith component by z_i. Let $I \subseteq [r]$ be a subset of coordinates, we denote by $\mathbf{z}_I$ the projection of $\mathbf{z}$ onto I. For any partition $I \cup J$ of coordinates $[r]$, we then write $\cdot$ for the inverse operation, i.e., $\mathbf{z}_I \cdot \mathbf{z}_J = \mathbf{z}$.

Lemma 4.20. *Let γ be an r-ary weighted relation and $I \cup J$ a partition of its coordinates $[r]$. If $\mathbf{x}, \mathbf{y} \in \mathrm{Feas}(\gamma)$ and*

$$\gamma(\mathbf{x}) + \gamma(\mathbf{y}) < \gamma(\mathbf{x}_I \cdot \mathbf{y}_J) + \gamma(\mathbf{y}_I \cdot \mathbf{x}_J)\,, \tag{4.11}$$

then there exist coordinates $i \in I, j \in J$ and a binary weighted relation $\gamma_{i,j} \in \{\gamma\}^*$ such that $(x_i, x_j), (y_i, y_j) \in \mathrm{Feas}(\gamma_{i,j})$ and

$$\gamma_{i,j}(x_i, x_j) + \gamma_{i,j}(y_i, y_j) < \gamma_{i,j}(x_i, y_j) + \gamma_{i,j}(y_i, x_j)\,. \tag{4.12}$$

Moreover, if every relation in $\{\gamma\}^*$ is 2-decomposable, then $\mathbf{x}_I \cdot \mathbf{y}_J \in \mathrm{Feas}(\gamma)$ implies $(x_i, y_j) \in \mathrm{Feas}(\gamma_{i,j})$ and $\mathbf{y}_I \cdot \mathbf{x}_J \in \mathrm{Feas}(\gamma)$ implies $(y_i, x_j) \in \mathrm{Feas}(\gamma_{i,j})$.

Proof. We prove the lemma by induction on the arity of γ. If $|I| = 0$, $|J| = 0$, or $|I| = |J| = 1$, the claim holds trivially. Otherwise we may without loss of generality assume that $|J| \ge 2$. Let $k \in J$ be an arbitrary coordinate and define $J' = J \setminus \{k\}$. We extend our notation $\cdot$ to $I \cup J' \cup \{k\}$ as a finer partition of $[r]$, and write for instance $\mathbf{x}$ as $\mathbf{x}_I \cdot \mathbf{x}_{J'} \cdot x_k$.

We first consider the case when $\mathbf{x}_I \cdot \mathbf{y}_{J'} \cdot x_k, \mathbf{y}_I \cdot \mathbf{x}_{J'} \cdot y_k \notin \mathrm{Feas}(\gamma)$. We restrict the domain at coordinate k to $\{x_k, y_k\}$ and minimise over it to obtain an $(r-1)$-ary weighted relation γ' with a coordinate partition $I \cup J'$. It holds $\gamma'(\mathbf{x}_I \cdot \mathbf{x}_{J'}) \le \gamma(\mathbf{x}), \gamma'(\mathbf{y}_I \cdot \mathbf{y}_{J'}) \le \gamma(\mathbf{y}), \gamma'(\mathbf{x}_I \cdot \mathbf{y}_{J'}) = \gamma(\mathbf{x}_I \cdot \mathbf{y}_J), \gamma'(\mathbf{y}_I \cdot \mathbf{x}_{J'}) = \gamma(\mathbf{y}_I \cdot \mathbf{x}_J)$, and the claim follows directly from the induction hypothesis for γ'.

We may now assume without loss of generality that $\mathbf{y}_I \cdot \mathbf{x}_{J'} \cdot y_k \in \mathrm{Feas}(\gamma)$. If it holds that

$$\gamma(\mathbf{x}_I \cdot \mathbf{x}_{J'} \cdot x_k) + \gamma(\mathbf{y}_I \cdot \mathbf{x}_{J'} \cdot y_k) < \gamma(\mathbf{x}_I \cdot \mathbf{x}_{J'} \cdot y_k) + \gamma(\mathbf{y}_I \cdot \mathbf{x}_{J'} \cdot x_k)\,, \tag{4.13}$$

we pin γ at every coordinate $j' \in J'$ to its respective label $x_{j'}$ to obtain a weighted relation γ' with a coordinate partition $I \cup \{k\}$. The claim then follows from the induction hypothesis for γ'. Note that $\mathbf{x}_I \cdot \mathbf{y}_J \in \mathrm{Feas}(\gamma)$ implies $(x_i, y_k) \in \mathrm{Pr}_{i,k}(\mathrm{Feas}(\gamma))$ for all $i \in I$; together with $(x_{j'}, y_k) \in$

$\mathrm{Pr}_{j',k}(\mathrm{Feas}(\gamma))$, $(x_i, x_{j'}) \in \mathrm{Pr}_{i,j'}(\mathrm{Feas}(\gamma))$ for all $i \in I, j' \in J'$ (as $\mathbf{y}_I \cdot \mathbf{x}_{J'} \cdot y_k, \mathbf{x} \in \mathrm{Feas}(\gamma)$) this implies $\mathbf{x}_I \cdot \mathbf{x}_{J'} \cdot y_k \in \mathrm{Feas}(\gamma)$ if $\mathrm{Feas}(\gamma)$ is 2-decomposable.

If (4.13) does not hold, we have $\mathbf{x}_I \cdot \mathbf{x}_{J'} \cdot y_k \in \mathrm{Feas}(\gamma)$, and therefore

$$\gamma(\mathbf{x}_I \cdot \mathbf{x}_{J'} \cdot y_k) + \gamma(\mathbf{y}_I \cdot \mathbf{y}_{J'} \cdot y_k) < \gamma(\mathbf{x}_I \cdot \mathbf{y}_{J'} \cdot y_k) + \gamma(\mathbf{y}_I \cdot \mathbf{x}_{J'} \cdot y_k)\,, \quad (4.14)$$

otherwise the sum of negated (4.13) and (4.14) would contradict (4.11). We resolve this case analogously to the previous one, this time pinning γ at coordinate k to y_k. □

4.2 Boolean languages

In this section, we prove a necessary condition for p-tractability of intractable Boolean languages (see Theorems 4.36 and 4.41), and show that the planar VCSP over languages satisfying the condition can be reduced to a special case in which every variable appears in at most two constraints (see Theorem 4.23).

First we define necessary concepts and notation. We assume $D = \{0, 1\}$ throughout the section.

Definition 4.21. A weighted relation (analogously, a language) is called *self-complementary* if it admits multimorphism $\langle \neg \rangle$.

Note that a weighted relation γ is self-complementary if and only if $\gamma(\mathbf{x}) = \gamma(\neg(\mathbf{x}))$ for all $\mathbf{x} \in \mathrm{Feas}(\gamma)$.

As the first part of our main result, we establish in Theorem 4.36 that self-complementarity is a necessary condition for p-tractability of intractable languages.

Definition 4.22. For any $\mathbf{x} = (x_1, \dots, x_r) \in D^r$, we define $\mathrm{diff}(\mathbf{x}) = (x'_1, \dots, x'_r) \in D^r$ by $x'_i = x_i \oplus x_{i+1}$, where $x_{r+1} = x_1$. Note that $x'_1 + \cdots + x'_r$ is even, and that $\mathrm{diff}(\mathbf{x}) = \mathrm{diff}(\mathbf{y})$ if and only if $\mathbf{y} \in \{\mathbf{x}, \neg(\mathbf{x})\}$.

For any self-complementary r-ary weighted relation γ, we define $\mathrm{diff}(\gamma)$ to be the r-ary weighted relation such that $\mathrm{diff}(\gamma)(\mathrm{diff}(\mathbf{x})) = \gamma(\mathbf{x})$ for all $\mathbf{x} \in D^r$, and $\mathrm{diff}(\gamma)(\mathbf{x}') = \infty$ for all $\mathbf{x}' = (x'_1, \dots, x'_r) \in D^r$ with an odd value of $x'_1 + \cdots + x'_r$.

The *fanout limitation* is a structural restriction studied in the context of the CSP [45] which limits the maximum number of constraints in which a variable may participate. Dalmau and Ford [34] showed that allowing at least three occurrences of every variable is sufficient to retain the NP-hardness of any intractable Boolean language. Hence, the only interesting variant is restricting the number of occurrences to at most two. This is called the *edge*

CSP in [69]; the name draws an analogy between variables and edges of a graph whose vertices represent constraints.[4]

We say that a VCSP instance is an *edge instance* if every variable occurs at most twice in the scope of the constraints (i.e., every variable has degree at most two in the incidence graph). We denote by $\mathrm{VCSP_e}(\Gamma)$ the VCSP restricted to edge instances over language Γ.

As showed in [41], a planar instance over a self-complementary $\{0, \infty\}$-valued language Γ can be reduced to an edge instance over language $\mathrm{diff}(\Gamma) \cup \{A_1, A_2, A_3\}$, where A_r denotes the r-ary relation consisting of all tuples $(x_1, \ldots, x_r) \in D^r$ such that $x_1 + \cdots + x_r$ is even. We extend this reduction to $\overline{\mathbb{Q}}$-valued languages. In fact, the resulting edge instance is also planar, but this does not seem to be instrumental in establishing tractability results later (namely, Theorem 4.27 does not require planarity).

Theorem 4.23. *For any self-complementary language* Γ, *it holds*

$$\mathrm{VCSP_p}(\Gamma) \leq_p \mathrm{VCSP_e}(\mathrm{diff}(\Gamma) \cup \{A_1, A_2, A_3\})\,. \tag{4.15}$$

Proof. Let (I, G, τ) be a plane instance with $I = (V, D, \phi_I) \in \mathrm{VCSP_p}(\Gamma)$ and $G = (V, E)$. We denote the image of τ by $T \subseteq F(G)$. Without loss of generality, we may assume that every face in $F(G) \setminus T$ is bounded by at most three edges (otherwise it can be split into two smaller faces by adding a new edge).

We construct an instance $I' = (E, D, \phi_{I'}) \in \mathrm{VCSP_e}(\mathrm{diff}(\Gamma) \cup \{A_1, A_2, A_3\})$ as follows. The variables of I' correspond to the edges of G. For every constraint $\gamma(x_1, \ldots, x_{\mathrm{ar}(\gamma)})$ in I, we impose a constraint $\mathrm{diff}(\gamma)(e_1, \ldots, e_{\mathrm{ar}(\gamma)})$ in I' with the same weight, where $e_i = \{x_i, x_{i+1}\}$ and $x_{\mathrm{ar}(\gamma)+1} = x_1$.[5] For every face $f \in F(G) \setminus T$, we impose a constraint $A_r(e_1, \ldots, e_r)$ in I', where $e_1, \ldots, e_r$ are the edges bounding f. Since every edge of G participates in exactly two constraints, I' is an edge instance.

Now we show that the values of optimal solutions to I and I' are equal. For any labelling $s : V \to D$, we define $s' : E \to D$ by $s'(\{u, v\}) = s(u) \oplus s(v)$. By the definition of operation diff, it holds

$$\gamma(s(x_1), \ldots, s(x_{\mathrm{ar}(\gamma)})) = \mathrm{diff}(\gamma)(s'(e_1), \ldots, s'(e_{\mathrm{ar}(\gamma)})) \tag{4.16}$$

for every constraint $\gamma(x_1, \ldots, x_{\mathrm{ar}(\gamma)})$ in I, and hence $\phi_{I'}(s') = \phi_I(s)$. Conversely, consider any labelling $s' : E \to D$. If there exists a face $f \in F(G)$ bounded by edges $e_1, \ldots, e_r$ such that $s'(e_1) + \cdots + s'(e_r)$ is odd, then s' is

---

[4] In fact, variables are required to occur in *exactly* two constraints in [69]. Both variants are polynomial-time equivalent by a simple reduction.

[5] Since τ maps every constraint of I to a face of G, it holds $e_i \in E$ for all $i \in [\mathrm{ar}(\gamma)]$.

infeasible, as it incurs an infinite value from a constraint $\mathrm{diff}(\gamma)$ (if $f \in T$) or A_r (if $f \in F(G) \setminus T$). Otherwise, $s'(e_1) + \cdots + s'(e_r)$ is even for *every cycle* $e_1, \ldots, e_r$ in G. Therefore, there exists a labelling $s : V \to D$ such that $s(u) \oplus s(v) = s'(\{u, v\})$ holds for all $\{u, v\} \in E$.[6] By the same argument as above, it holds $\phi_I(s) = \phi_{I'}(s')$. ☐

Valuated delta-matroids were originally identified as functions whose *maximum* can be found by a certain greedy algorithm [40]. These can be also described as *M-concave* functions on a special case of constant-parity jump systems [11, 86]. We define valuated delta-matroids as *M-convex* functions or, equivalently, as functions whose *minimum* can be found by an analogous greedy algorithm.

For any r-tuples $\mathbf{x} = (x_1, \ldots, x_r), \mathbf{y} = (y_1, \ldots, y_r) \in D^r$, we denote by $\Delta(\mathbf{x}, \mathbf{y})$ the set of coordinates where $\mathbf{x}$ and $\mathbf{y}$ differ, i.e., $\Delta(\mathbf{x}, \mathbf{y}) = \{i \in [r] \mid x_i \neq y_i\}$. For $\mathbf{x} \in D^r$ and $U \subseteq [r]$, we denote by $\mathbf{x} \oplus U$ the tuple $\mathbf{y} \in D^r$ such that $\Delta(\mathbf{x}, \mathbf{y}) = U$.

Definition 4.24. A relation ρ is an *even delta-matroid* if, for any $\mathbf{x}, \mathbf{y} \in \rho$ and $i \in \Delta(\mathbf{x}, \mathbf{y})$, there exists $j \in \Delta(\mathbf{x}, \mathbf{y}) \setminus \{i\}$ such that $\mathbf{x} \oplus \{i, j\} \in \rho$.

This property is called the *symmetric exchange condition*. It implies the following *strong exchange condition*.

Theorem 4.25 ([106]). *Let ρ be an even delta-matroid. For any $\mathbf{x}, \mathbf{y} \in \rho$ and $i \in \Delta(\mathbf{x}, \mathbf{y})$, there exists $j \in \Delta(\mathbf{x}, \mathbf{y}) \setminus \{i\}$ such that $\mathbf{x} \oplus \{i, j\}, \mathbf{y} \oplus \{i, j\} \in \rho$.*

Definition 4.26. A weighted relation γ is a *valuated delta-matroid* if, for any $\mathbf{x}, \mathbf{y} \in \mathrm{Feas}(\gamma)$ and $i \in \Delta(\mathbf{x}, \mathbf{y})$, there exists $j \in \Delta(\mathbf{x}, \mathbf{y}) \setminus \{i\}$ such that

$$\gamma(\mathbf{x}) + \gamma(\mathbf{y}) \geq \gamma(\mathbf{x} \oplus \{i, j\}) + \gamma(\mathbf{y} \oplus \{i, j\}) \,. \tag{4.17}$$

Note that any even delta-matroid is a valuated delta-matroid (by Theorem 4.25). Conversely, for any valuated delta-matroid γ, both $\mathrm{Feas}(\gamma)$ and $\mathrm{Opt}(\gamma)$ are even delta-matroids.

As the second part of our main result, we establish in Theorem 4.41 that, for any intractable language Γ that is p-tractable (and thus, by Theorem 4.36, self-complementary), language $\mathrm{diff}(\Gamma)$ must consist of valuated delta-matroids. Kazda et al. [69] proved that this condition is sufficient for p-tractability in the case of $\{0, \infty\}$-valued languages (by Theorem 4.23 and the fact that A_r is an even delta-matroid for every r).

---

[6] In fact, exactly two such labellings exist: Starting from an arbitrary vertex u, we may choose $s(u) = 0$ or $s(u) = 1$. The label of its neighbour v is then given by $s(v) = s'(\{u, v\}) \oplus s(u)$, and so on. This determines the labels of all vertices, since G is connected.

Theorem 4.27 ([69])**.** *For any* $\{0, \infty\}$*-valued language* Γ *consisting of even delta-matroids, problem* $\mathrm{VCSP_e}(\Gamma)$ *is polynomial-time solvable.*

We conjecture that this can be generalised to all ($\overline{\mathbb{Q}}$-valued) languages.

Conjecture 4.28. *For any language* Γ *consisting of valuated delta-matroids, problem* $\mathrm{VCSP_e}(\Gamma)$ *is polynomial-time solvable.*

In the rest of this section, we prove the necessary condition, stated as Theorems 4.36 and 4.41.

First, suppose that Γ is an intractable language, and hence it does not admit any of the multimorphisms from Theorem 2.30. Moreover, suppose that Γ is not self-complementary. We show in a series of lemmas that certain relations are p-expressible over Γ. In particular, Lemma 4.35 establishes that $\rho_{\text{1-in-3}}$ (see Example 4.3) is p-expressible, and hence Γ is p-intractable.

For any arity r and $i \in [r]$, let $\mathbf{e}_i^r$ be the r-tuple $\mathbf{0}^r \oplus \{i\}$. We denote by γ_0, γ_1, $\gamma_=$, $\gamma_{\neq}$ the soft variant of relations ρ_0, ρ_1, $\rho_=$, $\rho_{\neq}$ respectively.

Definition 4.29. Let γ be an r-ary weighted relation and $i \in [r]$. We call the 2-addition of $\rho_=$ ($\rho_{\neq}$) to γ at coordinate i the *=-restriction* (*≠-restriction*) of γ at coordinate i.

The *twist* of γ at coordinate i results in an r-ary weighted relation γ' such that, for every $\mathbf{x} \in D^r$, it holds $\gamma'(\mathbf{x}) = \gamma(\mathbf{x} \oplus \{i\})$.

In other words, a twist switches roles of labels 0 and 1 at a single coordinate.

Example 4.30. The twist of a binary relation $\{(0,0),(0,1),(1,0)\}$ at the first coordinate results in a binary relation $\{(1,0),(1,1),(0,0)\}$.

Lemma 4.31. *Operation =-restriction is p-expressible over any language containing* $\rho_=$*. Twist and ≠-restriction are p-expressible over any language containing* $\rho_{\neq}$*.*

Proof. As =-restriction and ≠-restriction are special cases of 2-addition, they are p-expressible by Lemma 4.12.

The twist of an r-ary weighted relation at coordinate i is p-expressible by a cycle $x_1, \dots, x_r$ connected to a vertex x_i' in the outer face by edges $\{x_i', x_{i-1}\}$, $\{x_i', x_{i+1}\}$ and a pair of parallel edges $\{x_i', x_i\}$. We impose a constraint $\rho_{\neq}(x_i', x_i)$ and map it to the face bounded by the pair of edges $\{x_i', x_i\}$. □

Lemma 4.32. *Let* Γ *be a language that admits neither of the multimorphisms* $\langle c_0 \rangle$, $\langle c_1 \rangle$*. Then* $\rho_0, \rho_1 \in \mathrm{wRelClone_p}(\Gamma)$ *or* $\rho_{\neq} \in \mathrm{wRelClone_p}(\Gamma)$*.*

Proof. If Γ does not admit $\langle c_0 \rangle$, it contains a weighted relation assigning to the zero tuple a value larger than the optimum. Applying Opt, we have that $\mathrm{wRelClone}_{\mathrm{p}}(\Gamma)$ contains a *relation* that is not invariant under c_0. We denote by ρ such a relation of minimum arity and by r its arity. Relation ρ is non-empty, but $\mathbf{0}^r \notin \rho$. If $r = 1$, then $\rho = \rho_1 \in \mathrm{wRelClone}_{\mathrm{p}}(\Gamma)$.

Otherwise, $\mathbf{e}_i^r \in \rho$ for all i, because the minimisation of ρ over coordinate i produces a non-empty relation invariant under c_0 (by the choice of ρ) and hence containing $\mathbf{0}^{r-1}$. If $r \geq 3$, the $=$-restriction of ρ at coordinate 2 followed by the minimisation results in an $(r-1)$-ary relation ρ' with $\mathbf{e}_1^{r-1} \in \rho'$ and $\mathbf{0}^{r-1} \notin \rho'$, which contradicts the choice of ρ. Therefore, $r = 2$. If $(1,1) \in \rho$, we would again get a contradiction by applying the $=$-restriction and minimisation at coordinate 1. Hence we have $\rho = \rho_{\neq} \in \mathrm{wRelClone}_{\mathrm{p}}(\Gamma)$.

By the analogous argument for multimorphism $\langle c_1 \rangle$, we get that $\rho_0 \in \mathrm{wRelClone}_{\mathrm{p}}(\Gamma)$ or $\rho_{\neq} \in \mathrm{wRelClone}_{\mathrm{p}}(\Gamma)$. ☐

Lemma 4.33. *Let Γ be a language that admits neither of the multimorphisms $\langle \min, \min \rangle$, $\langle \max, \max \rangle$, $\langle \min, \max \rangle$. If $\rho_0, \rho_1 \in \mathrm{wRelClone}_{\mathrm{p}}(\Gamma)$, then $\rho_{\neq} \in \mathrm{wRelClone}_{\mathrm{p}}(\Gamma)$.*

Proof. If $\min \notin \mathrm{Pol}(\mathrm{wRelClone}_{\mathrm{p}}(\Gamma))$, we choose a minimum-arity relation $\rho'_{\vee} \in \mathrm{wRelClone}_{\mathrm{p}}(\Gamma)$ that is not invariant under min; its arity r is at least 2. Let $\mathbf{x}, \mathbf{y} \in \rho'_{\vee}$ be r-tuples such that $\min(\mathbf{x}, \mathbf{y}) \notin \rho'_{\vee}$. Tuples $\mathbf{x}, \mathbf{y}$ differ at every coordinate, otherwise we would obtain a contradiction with the choice of $\rho'_{\vee}$ by taking a pinning instead. Therefore, $\min(\mathbf{x}, \mathbf{y}) = \mathbf{0}^r \notin \rho'_{\vee}$ and, by the same argument as in Lemma 4.32, we have $\mathbf{e}_i^r \in \rho'_{\vee}$ for all i. But then $r = 2$, otherwise we could take as $\mathbf{x}, \mathbf{y}$ tuples $\mathbf{e}_2^r, \mathbf{e}_3^r$ which agree at the first coordinate, and obtain a smaller counterexample by pinning. Hence we have $\rho_{\neq} \subseteq \rho'_{\vee} \subseteq \rho_{\neq} \cup \{(1,1)\}$.

If $\min \in \mathrm{Pol}(\mathrm{wRelClone}_{\mathrm{p}}(\Gamma))$, then we choose a minimum-arity weighted relation $\gamma \in \mathrm{wRelClone}_{\mathrm{p}}(\Gamma)$ that does not admit multimorphism $\langle \min, \min \rangle$ and denote its arity by r. Let $\mathbf{x}, \mathbf{y} \in \mathrm{Feas}(\gamma)$ be r-tuples such that $\gamma(\mathbf{x}) + \gamma(\mathbf{y}) < 2 \cdot \gamma(\min(\mathbf{x}, \mathbf{y}))$. Without loss of generality, we have $\gamma(\mathbf{x}) < \gamma(\min(\mathbf{x}, \mathbf{y}))$ and may assume that $\mathbf{y} = \min(\mathbf{x}, \mathbf{y})$. Again, $\mathbf{x}$ and $\mathbf{y}$ must differ at every coordinate, which implies $\mathbf{x} = \mathbf{1}^r, \mathbf{y} = \mathbf{0}^r$. If $r \geq 2$, we would obtain a contradiction by applying the $=$-restriction and minimisation at coordinate 1. Hence, $r = 1$ and by scaling and adding a constant to γ we get $\gamma_1 \in \mathrm{wRelClone}_{\mathrm{p}}(\Gamma)$.

Analogously, if $\max \notin \mathrm{Pol}(\mathrm{wRelClone}_{\mathrm{p}}(\Gamma))$, we get $\rho'_{\uparrow} \in \mathrm{wRelClone}_{\mathrm{p}}(\Gamma)$ where $\rho'_{\uparrow}$ is a binary relation such that $\rho_{\neq} \subseteq \rho'_{\uparrow} \subseteq \rho_{\neq} \cup \{(0,0)\}$. Otherwise,

$\gamma_0 \in \mathrm{wRelClone_p}(\Gamma)$. It holds

$$\begin{aligned} \rho_{\neq}(x,y) &= \rho'_{\vee}(x,y) + \rho'_{\uparrow}(x,y) && (4.18)\\ &= \mathrm{Opt}\left(\rho'_{\vee}(x,y) + \gamma_0(x) + \gamma_0(y)\right) && (4.19)\\ &= \mathrm{Opt}\left(\rho'_{\uparrow}(x,y) + \gamma_1(x) + \gamma_1(y)\right)\ , && (4.20) \end{aligned}$$

so $\rho_{\neq}$ can be constructed with a planar gadget if at least one of min, max is not a polymorphism of $\mathrm{wRelClone_p}(\Gamma)$.

Finally, consider the case when $\min, \max \in \mathrm{Pol}(\mathrm{wRelClone_p}(\Gamma))$ and hence $\gamma_0, \gamma_1 \in \mathrm{wRelClone_p}(\Gamma)$. Set $\mathrm{wRelClone_p}(\Gamma)$ is then a conservative language, so we have $\mathrm{wRelClone_p}(\Gamma)^* = \mathrm{wRelClone_p}(\Gamma)$. We choose a minimum-arity weighted relation $\gamma \in \mathrm{wRelClone_p}(\Gamma)$ that does not admit multimorphism $\langle \min, \max\rangle$ and denote its arity by r. Let $\mathbf{x}, \mathbf{y} \in \mathrm{Feas}(\gamma)$ be tuples such that $\gamma(\mathbf{x}) + \gamma(\mathbf{y}) < \gamma(\min(\mathbf{x},\mathbf{y})) + \gamma(\max(\mathbf{x},\mathbf{y}))$. Note that $\min(\mathbf{x},\mathbf{y}), \max(\mathbf{x},\mathbf{y}) \in \mathrm{Feas}(\gamma)$. By the choice of γ, tuples $\mathbf{x}, \mathbf{y}$ must differ at every coordinate, and hence $\mathbf{y} = \neg(\mathbf{x})$, $\min(\mathbf{x},\mathbf{y}) = \mathbf{0}^r$, $\max(\mathbf{x},\mathbf{y}) = \mathbf{1}^r$. We partition coordinates $[r]$ into $I = \{i \mid x_i = 0\}$ and $J = \{j \mid x_j = 1\}$. By Lemma 4.20, $\{\gamma\}^* \subseteq \mathrm{wRelClone_p}(\Gamma)$ contains a *binary* weighted relation that does not admit multimorphism $\langle \min, \max\rangle$, and hence $r = 2$. It holds $\gamma(0,1) + \gamma(1,0) < \gamma(0,0) + \gamma(1,1)$, where all the values are finite. We may assume that $\gamma(0,0) + \gamma(1,1) - \gamma(0,1) - \gamma(1,0) = 2$ and $\gamma(0,0) = 1$ (this can be achieved by scaling and adding a constant). We define unary weighted relations $\mu_1, \mu_2 \in \mathrm{wRelClone_p}(\Gamma)$ as $\mu_1(0) = \mu_2(0) = 0$, $\mu_1(1) = -\gamma(1,0)$, $\mu_2(1) = -\gamma(0,1)$. By adding μ_1 and μ_2 to γ at the first and second coordinate respectively we get $\gamma_{\neq}$, and therefore $\rho_{\neq} = \mathrm{Opt}(\gamma_{\neq}) \in \mathrm{wRelClone_p}(\Gamma)$. □

Lemma 4.34. *Let Γ be a language that does not admit multimorphism $\langle\neg\rangle$. If $\rho_{\neq} \in \mathrm{wRelClone_p}(\Gamma)$, then $\rho_0, \rho_1 \in \mathrm{wRelClone_p}(\Gamma)$.*

Proof. We choose a minimum-arity weighted relation $\gamma \in \mathrm{wRelClone_p}(\Gamma)$ that does not admit multimorphism $\langle\neg\rangle$ and denote its arity by r. Let $\mathbf{x} \in \mathrm{Feas}(\gamma)$ be an r-tuple such that $\gamma(\mathbf{x}) \neq \gamma(\neg(\mathbf{x}))$. It must hold $r = 1$, otherwise we would get a smaller counterexample by applying the $=$-restriction or $\neq$-restriction at the first coordinate (depending on whether $x_1 = x_2$ or $x_1 \neq x_2$) followed by minimisation. Hence, we have $\mathrm{Opt}(\gamma) \in \{\rho_0, \rho_1\}$, and the other relation can be obtained by a twist. □

Recall from Example 4.3 that $\rho_{\text{1-in-3}} = \{(0,0,1), (0,1,0), (1,0,0)\}$.

Lemma 4.35. *Let Γ be a language that admits neither of the multimorphisms $\langle \mathrm{Mn}, \mathrm{Mn}, \mathrm{Mn}\rangle$, $\langle \mathrm{Mj}, \mathrm{Mj}, \mathrm{Mj}\rangle$, $\langle \mathrm{Mj}, \mathrm{Mj}, \mathrm{Mn}\rangle$. If $\rho_0, \rho_1, \rho_{\neq} \in \mathrm{wRelClone_p}(\Gamma)$, then $\rho_{\text{1-in-3}} \in \mathrm{wRelClone_p}(\Gamma)$.*

Proof. If $\mathrm{Mn} \notin \mathrm{Pol}(\mathrm{wRelClone_p}(\Gamma))$, we choose a minimum-arity relation $\rho \in \mathrm{wRelClone_p}(\Gamma)$ that is not invariant under Mn. Its arity r must be at least 2; let us first assume $r \geq 3$. For any triple of r-tuples from ρ that agree at some coordinate, the r-tuple obtained by applying Mn to them also belongs to ρ (otherwise we would get a contradiction with the choice of ρ by taking a pinning instead). Let $\mathbf{x}, \mathbf{y}, \mathbf{z} \in \rho$ be r-tuples such that $\mathrm{Mn}(\mathbf{x}, \mathbf{y}, \mathbf{z}) \notin \rho$. Without loss of generality, we may assume that $\mathrm{Mn}(\mathbf{x}, \mathbf{y}, \mathbf{z}) = \mathbf{0}^r$ (this can be achieved with twists). By the same argument as in Lemma 4.32, we have $\mathbf{e}_i^r \in \rho$ for all i. Let $\mathbf{w} \in \rho$ be a tuple with the minimum even number of ones (such a tuple exists as at least one of $\mathbf{x}, \mathbf{y}, \mathbf{z}$ contains an even number of ones). If $\mathbf{w} \neq \mathbf{1}^r$, there are distinct coordinates i, j, k with $w_i = w_j = 1, w_k = 0$. Because $\mathbf{w}, \mathbf{e}_i^r, \mathbf{e}_j^r$ agree at coordinate k, tuple $\mathrm{Mn}(\mathbf{w}, \mathbf{e}_i^r, \mathbf{e}_j^r)$ belongs to ρ. However, it has two fewer ones than $\mathbf{w}$, which is a contradiction. Hence, $\mathbf{w} = \mathbf{1}^r$ and $r \geq 4$. But then $\mathrm{Mn}(\mathbf{1}^r, \mathbf{e}_3^r, \mathbf{e}_4^r) \notin \rho$ (as it contains an even number of ones), and we obtain a smaller counterexample by taking the $=$-restriction of ρ at the first coordinate followed by minimisation. Therefore, $r = 2$ and $|\rho| = 3$. Using twists, we can get from ρ relation $\rho_\uparrow = \{(0,0), (0,1), (1,0)\} \in \mathrm{wRelClone_p}(\Gamma)$.

If $\mathrm{Mj} \notin \mathrm{Pol}(\mathrm{wRelClone_p}(\Gamma))$, we choose a minimum-arity relation $\rho'_{\text{1-in-3}} \in \mathrm{wRelClone_p}(\Gamma)$ that is not invariant under Mj. Its arity r must be at least 3 since every unary and binary relation admits Mj as a polymorphism. By the same argument as for Mn, we may assume $\mathbf{0}^r \notin \rho'_{\text{1-in-3}}$, and it can be shown that $\mathbf{e}_i^r \in \rho'_{\text{1-in-3}}$ for all i. If $r \geq 4$, tuples $\mathbf{e}_1^r, \mathbf{e}_2^r, \mathbf{e}_3^r$ and $\mathrm{Mj}(\mathbf{e}_1^r, \mathbf{e}_2^r, \mathbf{e}_3^r) = \mathbf{0}^r$ agree at coordinate 4; we then obtain a smaller counterexample by pinning. Therefore, $r = 3$.

If neither of Mn, Mj is a polymorphism of $\mathrm{wRelClone_p}(\Gamma)$, we have

$$\rho_{\text{1-in-3}}(x, y, z) = \rho'_{\text{1-in-3}}(x, y, z) + \rho_\uparrow(x, y) + \rho_\uparrow(y, z) + \rho_\uparrow(z, x)\,, \tag{4.21}$$

which can be implemented in a planar way, and hence $\rho_{\text{1-in-3}} \in \mathrm{wRelClone_p}(\Gamma)$. Otherwise, Γ is not a crisp language, because that would make it admit multimorphism $\langle \mathrm{Mn}, \mathrm{Mn}, \mathrm{Mn} \rangle$ or $\langle \mathrm{Mj}, \mathrm{Mj}, \mathrm{Mj} \rangle$. Let $\mu \in \mathrm{wRelClone_p}(\Gamma)$ be a minimum-arity non-crisp weighted relation and $\mathbf{x}, \mathbf{y} \in \mathrm{Feas}(\mu)$ tuples such that $\mu(\mathbf{x}) \neq \mu(\mathbf{y})$. Tuples $\mathbf{x}, \mathbf{y}$ differ at every coordinate (otherwise we could obtain a smaller counterexample by pinning), and hence $\mathbf{y} = \neg(\mathbf{x})$. Moreover, μ is unary, otherwise we could apply the $=$-restriction or $\neq$-restriction at the first coordinate (depending on whether $x_1 = x_2$ or $x_1 \neq x_2$) followed by minimisation to obtain a smaller counterexample. If $\mu(0) < \mu(1)$, we get $\gamma_0 \in \mathrm{wRelClone_p}(\Gamma)$ by scaling μ and adding a constant, and $\gamma_1 \in \mathrm{wRelClone_p}(\Gamma)$

by twisting γ_0; the case $\mu(0) > \mu(1)$ is symmetric. It holds

$$\rho_{\text{1-in-3}}(x,y,z) = \operatorname{Opt}\left(\rho_{\uparrow}(x,y) + \rho_{\uparrow}(y,z) + \rho_{\uparrow}(z,x) + \gamma_1(x) + \gamma_1(y) + \gamma_1(z)\right) \tag{4.22}$$

$$= \operatorname{Opt}\left(\rho'_{\text{1-in-3}}(x,y,z) + \gamma_0(x) + \gamma_0(y) + \gamma_0(z)\right) . \tag{4.23}$$

Both can be implemented planarly, and therefore $\rho_{\text{1-in-3}} \in \mathrm{wRelClone_p}(\Gamma)$ if exactly one of Mn, Mj is a polymorphism of $\mathrm{wRelClone_p}(\Gamma)$.

Finally, we consider the case when both $\mathrm{Mn}, \mathrm{Mj} \in \mathrm{Pol}(\mathrm{wRelClone_p}(\Gamma))$. Let $\gamma \in \mathrm{wRelClone_p}(\Gamma)$ be an r-ary weighted relation of the minimum arity for which (2.11) does not hold as *equality* for multimorphism $\langle \mathrm{Mj}, \mathrm{Mj}, \mathrm{Mn}\rangle$. Let $\mathbf{x}, \mathbf{y}, \mathbf{z} \in \mathrm{Feas}(\gamma)$ be r-tuples that violate the equality. They do not agree at any coordinate (otherwise we could obtain a smaller counterexample by pinning), and hence $\mathrm{Mj}(\mathbf{x}, \mathbf{y}, \mathbf{z})$ and $\mathrm{Mn}(\mathbf{x}, \mathbf{y}, \mathbf{z})$ differ everywhere. Without loss of generality, we may assume that $\mathrm{Mj}(\mathbf{x}, \mathbf{y}, \mathbf{z}) = \mathbf{0}^r$ and $\mathrm{Mn}(\mathbf{x}, \mathbf{y}, \mathbf{z}) = \mathbf{1}^r$ (this can be achieved with twists) and $\mathbf{z} \neq \mathbf{0}^r$. Note that $\mathbf{0}^r, \mathbf{1}^r \in \mathrm{Feas}(\gamma)$ because Mj, Mn are polymorphisms of γ. Tuples $\mathbf{x}, \mathbf{y}, \mathbf{0}^r$ agree at all coordinates i where $z_i = 1$, and hence they satisfy (2.11) as equality, i.e.

$$\gamma(\mathbf{x}) + \gamma(\mathbf{y}) + \gamma(\mathbf{0}^r) = 2 \cdot \gamma(\mathrm{Mj}(\mathbf{x}, \mathbf{y}, \mathbf{0}^r)) + \gamma(\mathrm{Mn}(\mathbf{x}, \mathbf{y}, \mathbf{0}^r)) = 2 \cdot \gamma(\mathbf{0}^r) + \gamma(\neg(\mathbf{z})) . \tag{4.24}$$

Because $\gamma(\mathbf{x}) + \gamma(\mathbf{y}) + \gamma(\mathbf{z}) \neq 2 \cdot \gamma(\mathbf{0}^r) + \gamma(\mathbf{1}^r)$, this implies $\gamma(\mathbf{z}) + \gamma(\neg(\mathbf{z})) \neq \gamma(\mathbf{0}^r) + \gamma(\mathbf{1}^r)$. We are going to apply Lemma 4.20 for this disequality. Language $\mathrm{wRelClone_p}(\Gamma)$ is conservative (as it contains both γ_0, γ_1), and hence $\mathrm{wRelClone_p}(\Gamma)^* = \mathrm{wRelClone_p}(\Gamma)$. It admits a majority polymorphism, therefore every relation in $\mathrm{wRelClone_p}(\Gamma)$ is 2-decomposable [64]. We partition coordinates $[r]$ into $I = \{i \mid z_i = 0\}$ and $J = \{j \mid z_j = 1\}$. By Lemma 4.20, there is a binary weighted relation $\gamma' \in \{\gamma\}^* \subseteq \mathrm{wRelClone_p}(\Gamma)$ with $\mathrm{Feas}(\gamma') = D^2$ and $\gamma'(0,1) + \gamma'(1,0) \neq \gamma'(0,0) + \gamma'(1,1)$. We may assume that $\gamma'(0,1) + \gamma'(1,0) < \gamma'(0,0) + \gamma'(1,1)$, otherwise we apply a twist. As in the proof of Lemma 4.33, weighted relation $\gamma_{\neq}$ can be obtained from γ'. Then we planarly construct $\rho_{\text{1-in-3}} \in \mathrm{wRelClone_p}(\Gamma)$ as

$$\rho_{\text{1-in-3}}(x,y,z) = \operatorname{Opt}\left(\gamma_{\neq}(x,y) + \gamma_{\neq}(y,z) + \gamma_{\neq}(z,x) + \gamma_0(x) + \gamma_0(y) + \gamma_0(z)\right) . \tag{4.25}$$

□

Finally, we prove the first part of our necessary condition.

Theorem 4.36. *Let Γ be an intractable language. If Γ is not self-complementary, then it is p-intractable.*

Proof. Since Γ is intractable we know, by Theorem 2.30, that Γ admits neither of the multimorphisms $\langle c_0\rangle$, $\langle c_1\rangle$, $\langle \min, \min\rangle$, $\langle \max, \max\rangle$, $\langle \min, \max\rangle$, $\langle \mathrm{Mn}, \mathrm{Mn}, \mathrm{Mn}\rangle$, $\langle \mathrm{Mj}, \mathrm{Mj}, \mathrm{Mj}\rangle$, $\langle \mathrm{Mj}, \mathrm{Mj}, \mathrm{Mn}\rangle$. By assumption, Γ is not self-complementary, and hence it does not admit the $\langle\neg\rangle$ multimorphism.

By Lemmas 4.32 to 4.34, we have $\rho_0, \rho_1, \rho_{\neq} \in \mathrm{wRelClone_p}(\Gamma)$, and hence $\rho_{\text{1-in-3}} \in \mathrm{wRelClone_p}(\Gamma)$ by Lemma 4.35. The planar 1-in-3 Positive 3-Sat problem is NP-complete [84], and therefore Γ is p-intractable by Theorem 4.15. □

Now we prove the second part of the necessary condition, stated as Theorem 4.41. Suppose that Γ is an intractable self-complementary language, and hence it does not admit any of the multimorphisms from Theorem 2.30. As we show in Lemma 4.38, this implies that $\rho_{\neq}$ is p-expressible over Γ and, if Γ is not crisp, then $\gamma_=, \gamma_{\neq}$ are also p-expressible. Moreover, suppose that $\mathrm{diff}(\Gamma)$ contains a weighted relation that is not a valuated delta-matroid. In the $\{0, \infty\}$-valued case, Dvořák and Kupec [41] proved that relation $\rho_\times$ is then p-expressible over Γ, and hence Γ is p-intractable by Lemma 4.4.

Lemma 4.37 ([41, Lemma 7])**.** *Let Γ be a self-complementary language such that $\rho_{\neq} \in \Gamma$. If $\mathrm{diff}(\Gamma)$ contains a relation that is not an even delta-matroid, then $\rho_\times \in \mathrm{wRelClone_p}(\Gamma)$.*

In Lemma 4.40, we extend this result to the general case (i.e., to valuated delta-matroids).

Lemma 4.38. *Let Γ be a self-complementary language that admits neither of the multimorphisms $\langle c_0\rangle$, $\langle c_1\rangle$. Then $\rho_{\neq} \in \mathrm{wRelClone_p}(\Gamma)$. Moreover, if Γ is not crisp, then $\gamma_=, \gamma_{\neq} \in \mathrm{wRelClone_p}(\Gamma)$.*

Proof. By Lemma 4.32, it must hold $\rho_{\neq} \in \mathrm{wRelClone_p}(\Gamma)$, as Γ (and hence also $\mathrm{wRelClone_p}(\Gamma)$) is self-complementary, while ρ_0, ρ_1 are not.

In the rest we assume that $\mathrm{wRelClone_p}(\Gamma)$ contains a non-crisp weighted relation γ, i.e., there exist $\mathbf{x}, \mathbf{y} \in \mathrm{Feas}(\gamma)$ such that $\gamma(\mathbf{x}) < \gamma(\mathbf{y})$. We choose γ so that its arity $\mathrm{ar}(\gamma) = r$ is minimum, and $\mathbf{x}, \mathbf{y}$ so that their Hamming distance $|\Delta(\mathbf{x}, \mathbf{y})| = d$ is minimum.

Clearly, $d \geq 1$. Suppose that $d \geq 2$, and let $i \in \Delta(\mathbf{x}, \mathbf{y})$. It must hold $\gamma(\mathbf{y} \oplus \{i\}) \geq \gamma(\mathbf{y})$, otherwise we would get a contradiction with the choice of $\mathbf{x}, \mathbf{y}$ (as $|\Delta(\mathbf{y} \oplus \{i\}, \mathbf{y})| = 1 < d$). But then the minimisation of γ at coordinate i results in an $(r-1)$-ary non-crisp weighted relation, which is a contradiction. Hence, it holds $d = 1$.

Because γ is self-complementary, it holds $r \geq 2$. Suppose that $r \geq 3$. Then there exists $i \in [r]$ such that $\mathbf{x}$ and $\mathbf{y}$ agree at both coordinates i, $i+1$ (considered cyclically), and hence the $=$-restriction or the $\neq$-restriction (depending on whether $x_i = x_{i+1}$ or $x_i \neq x_{i+1}$) of γ at coordinate i followed by the minimisation results in an $(r-1)$-ary non-crisp weighted relation, which is a contradiction. Hence, it holds $r = 2$.

As $\mathrm{wRelClone_p}(\Gamma)$ is closed under addition of constants and non-negative scaling, we may assume that $\gamma(\mathbf{x}) = 0$, $\gamma(\mathbf{y}) = 1$. Therefore, we have $\gamma \in \{\gamma_=, \gamma_{\neq}\}$, and the other weighted relation can be obtained from γ by a twist. $\square$

For any r-ary weighted relation γ and $p : [r] \to \mathbb{Q}$, we denote by $\gamma[p]$ the r-ary weighted relation defined by

$$\gamma[p](\mathbf{x}) = \gamma(\mathbf{x}) + \sum_{x_i = 1} p(i)\,, \tag{4.26}$$

where $\mathbf{x} = (x_1, \dots, x_r) \in D^r$ and the sum is over $i \in [r]$.

Suppose that γ is a valuated delta-matroid. It is easy to show that, for any $p : [r] \to \mathbb{Q}$, $\gamma[p]$ is also a valuated delta-matroid, and hence $\mathrm{Opt}(\gamma[p])$ is an even delta-matroid. The following theorem establishes that this condition in fact characterises valuated delta-matroids.

Theorem 4.39 ([85, Theorem 2.2])**.** *Let γ be an r-ary weighted relation such that* $\mathrm{Feas}(\gamma)$ *is an even delta-matroid. Then γ is a valuated delta-matroid if and only if* $\mathrm{Opt}(\gamma[p])$ *is an even delta-matroid for all $p : [r] \to \mathbb{Q}$.*

Lemma 4.40. *Let Γ be a self-complementary language that admits neither of the multimorphisms $\langle c_0 \rangle$, $\langle c_1 \rangle$. If* $\mathrm{diff}(\Gamma)$ *contains a weighted relation that is not a valuated delta-matroid, then $\rho_\times \in \mathrm{wRelClone_p}(\Gamma)$.*

Proof. By Lemma 4.38, it holds $\rho_{\neq} \in \mathrm{wRelClone_p}(\Gamma)$. We claim that there exists a relation $\rho \in \mathrm{wRelClone_p}(\Gamma)$ such that $\mathrm{diff}(\rho)$ is not an even delta-matroid. Then, by Lemma 4.37, we get

$$\rho_\times \in \mathrm{wRelClone_p}(\{\rho_{\neq}, \rho\}) \subseteq \mathrm{wRelClone_p}(\Gamma)\,. \tag{4.27}$$

To prove the claim, let $\gamma \in \Gamma$ be an r-ary weighted relation such that $\mathrm{diff}(\gamma)$ is not a valuated delta-matroid. If $\mathrm{Feas}(\mathrm{diff}(\gamma))$ is not an even delta-matroid, then we are done, as $\mathrm{Feas}(\gamma) \in \mathrm{wRelClone_p}(\Gamma)$ and $\mathrm{diff}(\mathrm{Feas}(\gamma)) = \mathrm{Feas}(\mathrm{diff}(\gamma))$. Otherwise, γ is not crisp, and hence $\gamma_=, \gamma_{\neq} \in \mathrm{wRelClone_p}(\Gamma)$ by Lemma 4.38. We define auxiliary binary weighted relations $\gamma_+, \gamma_- \in \mathrm{wRelClone_p}(\Gamma)$ by $\gamma_+ = \gamma_=$ and $\gamma_- = \gamma_{\neq} - 1$, i.e., it holds

$$\gamma_+(x,y) = \begin{cases} 0 & \text{if } x = y \\ 1 & \text{if } x \neq y \end{cases}, \qquad \gamma_-(x,y) = \begin{cases} 0 & \text{if } x = y \\ -1 & \text{if } x \neq y \end{cases}. \tag{4.28}$$

By Theorem 4.39, there exists $p : [r] \to \mathbb{Q}$ such that $\mathrm{Opt}(\mathrm{diff}(\gamma)[p])$ is not an even delta-matroid. Let P_+ (P_-) denote the set of coordinates $i \in [r]$ for which $p(i)$ is positive (negative). We define an r-ary weighted relation γ' by

$$\gamma'(\mathbf{x}) = \gamma(\mathbf{x}) + \sum_{i \in P^+} p(i) \cdot \gamma_+(x_i, x_{i+1}) - \sum_{i \in P^-} p(i) \cdot \gamma_-(x_i, x_{i+1}) \,, \tag{4.29}$$

where $\mathbf{x} = (x_1, \ldots, x_r)$ and $x_{r+1} = x_1$. It holds $\gamma' \in \mathrm{wRelClone}_{\mathrm{p}}(\Gamma)$, as γ' can be obtained from γ by non-negative scaling and 2-addition of γ_+, γ_-. Because $\mathrm{diff}(\gamma') = \mathrm{diff}(\gamma)[p]$, relation $\mathrm{diff}(\mathrm{Opt}(\gamma'))$ is not an even delta-matroid. □

Theorem 4.41. *Let Γ be an intractable self-complementary language. If* $\mathrm{diff}(\Gamma)$ *contains a weighted relation that is not a valuated delta-matroid, then* Γ *is p-intractable.*

Proof. Language Γ is intractable, and hence, by Theorem 2.30, it does not admit either of the multimorphisms $\langle c_0 \rangle$, $\langle c_1 \rangle$. By Lemma 4.40, we have $\rho_\times \in \mathrm{wRelClone}_{\mathrm{p}}(\Gamma)$. By Lemma 4.4 and Theorem 4.15, Γ is p-intractable. □

4.3 Conservative languages

A language is called *conservative* if it contains all $\{0,1\}$-valued unary weighted relations. In this section, we establish a complexity classification of conservative languages on any finite domain D in the planar setting (see Theorem 4.45). Our classification precisely matches the one given in [73] for the general setting (see Theorem 4.43). This means that the planarity restriction does not add any tractable classes of conservative languages.

Definition 4.42. A k-ary operation $f : D^k \to D$ if called *conservative* if $f(x_1, \ldots, x_k) \in \{x_1, \ldots, x_k\}$ for every $x_1, \ldots, x_k \in D$. A multimorphism $\langle f_1, \ldots, f_k \rangle$ is called *conservative* if applying $\langle f_1, \ldots, f_k \rangle$ to $(x_1, \ldots, x_k)$ returns a permutation of $(x_1, \ldots, x_k)$.

A binary multimorphism $\langle f, g \rangle$ is called a *symmetric tournament pair* (STP) if it is conservative and both f and g are commutative operations.

A ternary multimorphism $\langle f, g, h \rangle$ is called *MJN* if it is conservative, f and g are (possibly different) majority operations, and h is a minority operation.

It was shown in [27] that languages admitting an STP multimorphism are tractable. In fact, languages admitting a certain combination of STP and MJN multimorphisms are also tractable.

Theorem 4.43 ([73])**.** *Let* Γ *be a conservative language. If* Γ *admits a conservative binary multimorphism* $\langle f, g\rangle$ *and a conservative ternary multimorphism* $\langle f', g', h'\rangle$*, and there is a family* M *of 2-element subsets of* D *such that*

- *for every* $\{a, b\} \in M$*,* $\langle f, g\rangle$ *restricted to* $\{a, b\}$ *is an STP multimorphism, and*
- *for every* $\{a, b\} \notin M$*,* $\langle f', g', h'\rangle$ *restricted to* $\{a, b\}$ *is an MJN multimorphism,*

then Γ *is tractable. Otherwise,* Γ *is intractable.*

The idea of the proof of Theorem 4.43 is as follows: Given a conservative language Γ, [73] defines a certain graph G_Γ whose vertices are pairs of distinct labels from D and an edge $(a, b) - (c, d)$ is present if there exists a binary weighted relation $\gamma \in \text{wRelClone}(\Gamma)$ that is "non-submodular with respect to the order $a < b$ and $c < d$". The edges of G_Γ are then classified as soft and hard. It is shown that a soft self-loop implies intractability of Γ. Otherwise, the vertices of G_Γ are partitioned into $M \cup \overline{M}$, where M denotes the set of loopless vertices and $\overline{M}$ denotes the rest (i.e. vertices with hard loops). It is then shown that G_Γ restricted to M is bipartite, which is in turn used to construct a binary multimorphism and a ternary multimorphism of Γ such that the binary multimorphism is an STP on M and the ternary multimorphism is an MJN on $\overline{M}$. (Proving that the constructed objects are multimorphisms of Γ is the most technical part of the proof.) Any such language is then tractable via an involved algorithm from [73] that relies on [27], or by an LP relaxation [101].

Our approach is to follow the above-described proof and adapt it to the planar setting. We remark that similar graphs to G_Γ have been important in other studies of the (V)CSP. In particular, in the classification of the conservative CSP [17] and in the classification of Minimum Cost Homomorphism problems [96]. In [17], the graph has labels as vertices and three types of edges depending on three types of polymorphisms. In [96], the graph has, as in our case, pairs of labels as vertices but the edges of the graph are defined, informally, via a min/max polymorphism rather than a $\langle \min, \max\rangle$ multimorphism. Also, edges in [96] are not classified as soft or hard.

It is natural to replace $\text{wRelClone}(\Gamma)$ by $\text{wRelClone}_\text{p}(\Gamma)$ in the definition of G_Γ. However, this simple change does not immediately yield the desired result. There are two main obstacles. First, the proof of Theorem 4.43 from [73] heavily relies on [96], which (in the tractable case) guarantees the existence of a majority polymorphism, and hence that the language is 2-decomposable. Second, some of the gadgets (for instance, the "i-expansion" from [73, Section 6.4]) are not necessarily planar. In more detail, [96] builds a similar graph to ours (as described above) and argues that, in the tractable case, this graph is bipartite (part of our G_Γ is also bipartite). This property is then used in [96] to argue about the existence of a majority polymorphism. However, this is proved in [96] using clones and depends on the Galois connection between clones and relational clones; such a connection is not known for p-expressibility!

To avoid these obstacles, we modify, significantly simplify, and generalise the proof so that it works in the planar setting. The key changes are the following. We define our graph based on a language closure Γ^* (see Definition 4.16), which is a subset of the planar weighted relational clone of a conservative language. We do *not* rely on Takhanov's result on the existence of a majority polymorphism [96] but instead prove directly that (the closure of) Γ is 2-decomposable. We define different STP and MJN multimorphisms, which allows us to simplify the proof that these are indeed multimorphisms of Γ. In particular, we prove modularity of weighted relations on $\overline{M}$, and show that the ternary MJN multimorphism satisfies (2.11) with equality, thus obtaining a better structural understanding of tractable languages. The main simplification is that we define the MJN multimorphism as close to projection operations as possible; in particular, we define it independently from the STP multimorphism (as opposed to [73]).

We remark that it is not clear how to derive non-trivial properties of graph G_Γ used in our proofs from the related graph defined in [73] apart from the obvious fact that our graph is a subgraph of the graph from [73]. We believe that with more work one can derive, using techniques and proofs from this section, that the two graphs are in fact the same, but we have not done so since our goal was to obtain a complexity classification.

Definition 4.44. Let Γ be a conservative language; recall that Γ^* denotes the closure of Γ as defined in Definition 4.16. We define an undirected graph G_Γ on vertices (a, b) for all $a, b \in D, a \neq b$. For any vertex $v = (a, b)$, we denote by $\overline{v}$ vertex (b, a). Graph G_Γ is allowed to have self-loops. It contains edge $(a_1, b_1) - (a_2, b_2)$ if there is a binary weighted relation $\gamma \in \Gamma^*$ such that $(a_1, b_2), (b_1, a_2) \in \text{Feas}(\gamma)$ and

$$\gamma(a_1, b_2) + \gamma(b_1, a_2) < \gamma(a_1, a_2) + \gamma(b_1, b_2)\,. \tag{4.30}$$

If there exists such a weighted relation γ with at least one of $(a_1, a_2), (b_1, b_2)$ belonging to $\text{Feas}(\gamma)$, we call the edge *soft*, otherwise the edge is *hard*. We denote by $\overline{M}$ and M the set of vertices with and without self-loops respectively.

Theorem 4.45. *If G_Γ has a soft self-loop, language Γ is p-intractable. Otherwise, Γ is tractable; in particular, Γ admits a conservative binary multimorphism $\langle \sqcap, \sqcup \rangle$ that is STP on M, and a conservative ternary multimorphism $\langle \mathrm{Mj}_1, \mathrm{Mj}_2, \mathrm{Mn}_3 \rangle$ that is MJN on $\overline{M}$.*

Proof. The p-intractable case is proved in Theorem 4.47. In the case when G_Γ has no soft self-loop, we establish multimorphisms $\langle \sqcap, \sqcup \rangle$ and $\langle \mathrm{Mj}_1, \mathrm{Mj}_2, \mathrm{Mn}_3 \rangle$ in Theorems 4.52 and 4.55 respectively; the tractability then follows from Theorem 4.43. $\square$

The following lemma gives a useful alternative characterisation of an edge in G_Γ.

Lemma 4.46. *Graph G_Γ contains edge $(a_1, b_1) - (a_2, b_2)$ if, and only if, binary relation $\{(a_1, b_2), (b_1, a_2)\}$ belongs to Γ^*. The edge is soft if, and only if, at least one of binary relations $\{(a_1, a_2), (a_1, b_2), (b_1, a_2)\}$, $\{(b_1, b_2), (a_1, b_2), (b_1, a_2)\}$ belongs to Γ^*.*

Proof. Both *if* implications follow directly from the definition of G_Γ; we need to prove the *only if* part. Let γ be a weighted relation establishing edge $(a_1, b_1) - (a_2, b_2)$ such that $\mathrm{Feas}(\gamma) \subseteq \{a_1, b_1\} \times \{a_2, b_2\}$ (this can be always achieved by domain restriction). Note that we may add to γ any unary finite-valued weighted relation without invalidating (4.30). We choose any $\lambda \in \mathbb{Q}$ such that $\lambda < \gamma(b_1, b_2)$ and $\gamma(a_1, b_2) + \gamma(b_1, a_2) - \lambda < \gamma(a_1, a_2)$. Note that such λ exists due to (4.30). We define unary weighted relations γ_1, γ_2 such that $\gamma_1(a_1) = \lambda - \gamma(a_1, b_2)$, $\gamma_2(a_2) = \lambda - \gamma(b_1, a_2)$, and $\gamma_1(x) = \gamma_2(x) = 0$ otherwise. Now consider binary weighted relation γ' defined as $\gamma'(x, y) = \gamma(x, y) + \gamma_1(x) + \gamma_2(y)$. We have $\gamma'(a_1, b_2) = \gamma'(b_1, a_2) = \lambda$ and $\lambda < \gamma'(a_1, a_2), \gamma'(b_1, b_2)$, so then $\mathrm{Opt}(\gamma') = \{(a_1, b_2), (b_1, a_2)\} \in \Gamma^*$.

If the edge is soft and $(a_1, a_2), (b_1, b_2) \in \mathrm{Feas}(\gamma)$, we proceed as above with $\lambda = \gamma(b_1, b_2)$, so that $\mathrm{Opt}(\gamma') = \{(b_1, b_2), (a_1, b_2), (b_1, a_2)\} \in \Gamma^*$. Otherwise we simply take $\mathrm{Feas}(\gamma) \in \Gamma^*$. $\square$

We show that the absence of soft self-loops is a necessary condition for p-tractability.

Theorem 4.47. *If G_Γ has a soft self-loop, language Γ is p-intractable.*

Proof. Let (a, b) be a vertex of G_Γ with a soft self-loop. Without loss of generality, we have $\rho = \{(a, a), (a, b), (b, a)\} \in \Gamma^*$ by Lemma 4.46. We denote by γ_a, γ_b the unary weighted relations defined as $\gamma_a(a) = \gamma_b(b) = 0$, $\gamma_a(b) = \gamma_b(a) = 1$, and $\gamma_a(x) = \gamma_b(x) = \infty$ for $x \notin \{a, b\}$.

Set $\Gamma' = \{\rho, \gamma_a, \gamma_b\} \subseteq \Gamma^*$ can be viewed as a conservative language over a Boolean domain $\{a, b\}$. By Theorem 2.30, Γ' is intractable. (In particular, Γ' does not fall into either of the two tractable cases for Boolean conservative languages, corresponding to the $\langle \min, \max\rangle$ and $\langle \mathrm{Mj}, \mathrm{Mj}, \mathrm{Mn}\rangle$ multimorphisms.) As Γ' is not self-complementary, it is p-intractable by Theorem 4.36. Therefore, Γ is p-intractable by Theorem 4.15. $\square$

It remains to show that this condition is also sufficient for tractability. This is done by establishing multimorphisms $\langle \sqcap, \sqcup\rangle$ and $\langle \mathrm{Mj}_1, \mathrm{Mj}_2, \mathrm{Mn}_3\rangle$ in Theorems 4.52 and 4.55 respectively. First, we prove some properties of Γ^* and graph G_Γ. From now on we assume that G_Γ has no soft self-loop.

Lemma 4.48. *For any vertex v, graph G_Γ contains edge $v - \overline{v}$. There is no edge between M and $\overline{M}$, no odd cycle in M, and no soft edge in $\overline{M}$.*

Proof. As the binary equality relation belongs to Γ^*, we have edge $v - \overline{v}$ for all vertices v.

Consider any sequence of vertices v_1, v_2, v_3, v_4 such that there is an edge between every two consecutive ones, and denote $v_i = (a_i, b_i)$. By Lemma 4.46, there exist binary relations $\rho_i = \{(a_i, b_{i+1}), (b_i, a_{i+1})\} \in \Gamma^*$ for $i \in \{1, 2, 3\}$. Their join equals $\{(a_1, b_4), (b_1, a_4)\} \in \Gamma^*$, and hence G_Γ contains edge $v_1 - v_4$. If any of edges $v_1 - v_2, v_2 - v_3, v_3 - v_4$ is soft, we can replace the corresponding relation ρ_i with $\{(a_i, a_{i+1}), (a_i, b_{i+1}), (b_i, a_{i+1})\}$ or $\{(b_i, b_{i+1}), (a_i, b_{i+1}), (b_i, a_{i+1})\}$ to show that $v_1 - v_4$ is also soft.

Suppose that there is an edge between $s \in M$ and $t \in \overline{M}$. Then we have edges $s - t, t - t, t - s$, and hence also self-loop $s - s$, which is a contradiction.

If there is an odd cycle in M, let us choose a shortest one and denote its vertices $v_1, \dots, v_k$ ($k \geq 3$). We have a sequence of adjacent vertices v_k, v_1, v_2, v_3, and hence v_3 and v_k are also adjacent. But that means there is a shorter odd cycle (or a self-loop) $v_3, \dots, v_k$; a contradiction.

Finally, suppose that $s, t \in \overline{M}$ and edge $s - t$ is soft. Then we have edges $s - t, t - t, t - s$, and hence a soft self-loop at s, which is a contradiction. $\square$

Lemma 4.49. *Every relation in Γ^* is 2-decomposable.*

Proof. Let $\rho \in \Gamma^*$ be an r-ary relation. By definition, $\mathbf{x} \in \rho$ implies $\bigwedge_{1 \leq i,j \leq r} (x_i, x_j) \in \mathrm{Pr}_{i,j}(\rho)$. We prove the converse implication by induction on r. If $r \leq 2$, relation ρ is trivially 2-decomposable. Let $r = 3$. Suppose for the sake of contradiction that $(x_1, x_2, x_3) \notin \rho$ even though $(y_1, x_2, x_3), (x_1, y_2, x_3), (x_1, x_2, y_3) \in \rho$ for some $y_1, y_2, y_3 \in D$. Let $\rho_1 \in \Gamma^*$ be the binary relation obtained from ρ by pinning it at the first coordinate to label x_1; we have $(x_2, y_3), (y_2, x_3) \in \rho_1$, $(x_2, x_3) \notin \rho_1$, and thus graph G_Γ contains edge $(x_2, y_2) - (x_3, y_3)$. Analogously, the graph contains edges $(x_3, y_3) - (x_1, y_1)$ and $(x_1, y_1) - (x_2, y_2)$. This is an odd cycle, so it must hold $(x_1, y_1), (x_2, y_2), (x_3, y_3) \in \overline{M}$. Let γ be a unary weighted relation with

$\gamma(x_1) = 0, \gamma(y_1) = 1$ and $\gamma(z) = \infty$ for all $z \in D \setminus \{x_1, y_1\}$. By adding γ to ρ at the first coordinate and then minimising over it we show that edge $(x_2, y_2) - (x_3, y_3)$ is soft, which is a contradiction.

It remains to prove the lemma for $r \geq 4$. Let $\rho_1 \in \Gamma^*$ be the relation obtained from ρ by minimisation over the first coordinate. Relation ρ_1 is 2-decomposable by the induction hypothesis, so $(x_2, \ldots, x_r) \in \rho_1$, and hence $(y_1, x_2, \ldots, x_r) \in \rho$ for some $y_1 \in D$. Analogously, we have $(x_1, y_2, x_3, \ldots, x_r), (x_1, x_2, y_3, x_4, \ldots, x_r) \in \rho$ for some $y_2, y_3 \in D$. Pinning ρ at every coordinate $k \geq 4$ to its respective label x_k gives a ternary 2-decomposable relation ρ' such that $(x_i, x_j) \in \mathrm{Pr}_{i,j}(\rho')$ for all $i, j \in \{1, 2, 3\}$. Therefore, $(x_1, x_2, x_3) \in \rho'$ and $\mathbf{x} \in \rho$. ☐

The following lemma involves an extension of the definition of an edge in G_Γ to non-binary weighted relations.

Lemma 4.50. *Let $\gamma \in \Gamma^*$ be an r-ary weighted relation and $I \cup J$ a partition of its coordinates $[r]$. If $\mathbf{x}, \mathbf{y} \in \mathrm{Feas}(\gamma)$ and*

$$\gamma(\mathbf{x}) + \gamma(\mathbf{y}) < \gamma(\mathbf{x}_I \cdot \mathbf{y}_J) + \gamma(\mathbf{y}_I \cdot \mathbf{x}_J)\,, \tag{4.31}$$

then graph G_Γ contains edge $(x_i, y_i) - (y_j, x_j)$ for some $i \in I, j \in J$. If at least one of $\mathbf{x}_I \cdot \mathbf{y}_J, \mathbf{y}_I \cdot \mathbf{x}_J$ belongs to $\mathrm{Feas}(\gamma)$, the edge is soft.

Proof. By Lemma 4.20, there are coordinates $i \in I, j \in J$ and a binary weighted relation $\gamma_{i,j} \in \Gamma^*$ such that $(x_i, x_j), (y_i, y_j) \in \mathrm{Feas}(\gamma_{i,j})$ and

$$\gamma_{i,j}(x_i, x_j) + \gamma_{i,j}(y_i, y_j) < \gamma_{i,j}(x_i, y_j) + \gamma_{i,j}(y_i, x_j)\,, \tag{4.32}$$

so graph G_Γ contains edge $(x_i, y_i) - (y_j, x_j)$. If $\mathbf{x}_I \cdot \mathbf{y}_J$ or $\mathbf{y}_I \cdot \mathbf{x}_J$ belongs to $\mathrm{Feas}(\gamma)$, then (x_i, y_j) or (y_i, x_j) belongs to $\mathrm{Feas}(\gamma_{i,j})$ (as $\{\gamma\}^*$ is 2-decomposable by Lemma 4.49), and hence the edge is soft. ☐

Lemma 4.51. *Let $\gamma \in \Gamma^*$ be an r-ary weighted relation and $I \cup J$ a partition of its coordinates $[r]$. If $\mathbf{x}, \mathbf{y}, \mathbf{x}_I \cdot \mathbf{y}_J, \mathbf{y}_I \cdot \mathbf{x}_J \in \mathrm{Feas}(\gamma)$ and, for all $i \in I$, $(x_i, y_i) \in \overline{M}$, then*

$$\gamma(\mathbf{x}) + \gamma(\mathbf{y}) = \gamma(\mathbf{x}_I \cdot \mathbf{y}_J) + \gamma(\mathbf{y}_I \cdot \mathbf{x}_J)\,. \tag{4.33}$$

Proof. Suppose for the sake of contradiction that the equality does not hold. Without loss of generality, we may assume that $\gamma(\mathbf{x}) + \gamma(\mathbf{y}) < \gamma(\mathbf{x}_I \cdot \mathbf{y}_J) + \gamma(\mathbf{y}_I \cdot \mathbf{x}_J)$. By Lemma 4.50, graph G_Γ contains a soft edge incident to (x_i, y_i) for some $i \in I$, which contradicts Lemma 4.48. ☐

Graph G_Γ does not have any odd cycle on vertices M. Therefore, there is a partition of M into two independent sets M_1, M_2. (In fact, it can be shown that every connected component of G_Γ restricted to M is a complete bipartite graph but we do not need this fact here.) Note that $(a, b) \in M_1$ if, and only if, $(b, a) \in M_2$, as every vertex $v \in M$ is adjacent to $\overline{v}$. We define multimorphism $\langle \sqcap, \sqcup \rangle$ as follows:

$$\langle \sqcap, \sqcup \rangle(x, y) = \begin{cases} (x, y) & \text{if } (x, y) \in M_1, \quad (4.34a) \\ (y, x) & \text{if } (x, y) \in M_2, \quad (4.34b) \\ (x, y) & \text{otherwise.} \quad (4.34c) \end{cases}$$

By definition, $\langle \sqcap, \sqcup \rangle$ is commutative on M.

Theorem 4.52. *$\langle \sqcap, \sqcup \rangle$ is a multimorphism of Γ.*

Proof. Let $\gamma \in \Gamma$ be an r-ary weighted relation and $\mathbf{x}, \mathbf{y} \in \text{Feas}(\gamma)$. Suppose for the sake of contradiction that (2.11) does not hold. We partition set $[r]$ into I and J: Set J consists of all coordinates j such that case (4.34b) applies to (x_j, y_j); set I covers the other two cases. For any $i \in I$, either $x_i = y_i$ or $(x_i, y_i) \in M_1 \cup \overline{M}$. For any $j \in J$, $(x_j, y_j) \in M_2$ and hence $(y_j, x_j) \in M_1$. $\langle \sqcap, \sqcup \rangle$ maps $\mathbf{x}, \mathbf{y}$ to $\mathbf{x}_I{\cdot}\mathbf{y}_J, \mathbf{y}_I{\cdot}\mathbf{x}_J$, so we have $\gamma(\mathbf{x}) + \gamma(\mathbf{y}) < \gamma(\mathbf{x}_I{\cdot}\mathbf{y}_J) + \gamma(\mathbf{y}_I{\cdot}\mathbf{x}_J)$. By Lemma 4.50, graph G_Γ contains an edge $(x_i, y_i) - (y_j, x_j)$ for some $i \in I, j \in J$. Since M_1 is an independent set, this implies $(x_i, y_i) \in \overline{M}$, which contradicts Lemma 4.48. □

The following definition corresponds to the "μ function" from [73, Section 6].

Definition 4.53. For any $a, b, c \in D$, we say that $ab|c$ holds if a, b, c are all different labels and there exist $(s, t) \in \overline{M}$ such that binary relation $\{(a, s), (b, s), (c, t)\}$ belongs to Γ^*.

The intuition is that if $ab|c$ holds, then any minority operation on $\overline{M}$ must map any permutation of $\{a, b, c\}$ to c in order to be a polymorphism of Γ.

Lemma 4.54. *For any $a, b, c \in D$, at most one of $ab|c$, $ca|b$, $bc|a$ holds. If $ab|c$, then $(a, c), (b, c) \in \overline{M}$.*

Proof. Suppose that both $ca|b$ and $bc|a$ hold. Then there are $(s_1, t_1), (s_2, t_2) \in \overline{M}$ and binary relations $\rho_1, \rho_2 \in \Gamma^*$ such that $\rho_1 = \{(c, s_1), (a, s_1), (b, t_1)\}$, $\rho_2 = \{(b, s_2), (c, s_2), (a, t_2)\}$. We construct their join ρ as $\rho(x, y) = \min_{z \in D}(\rho_1(z, x) + \rho_2(z, y))$. We have $\rho \in \Gamma^*$ and $\rho = \{(s_1, s_2), (s_1, t_2), (t_1, s_2)\}$, which implies a soft edge in $\overline{M}$ and hence a contradiction.

If $ab|c$, then there are $(s,t) \in \overline{M}$ such that $\{(a,s),(b,s),(c,t)\} \in \Gamma^*$. By restricting this relation at the first coordinate to labels $\{a,c\}$ we get edge $(a,c) - (t,s)$ and thus $(a,c) \in \overline{M}$; analogously by restricting to $\{b,c\}$ we get $(b,c) \in \overline{M}$. ☐

We define multimorphism $\langle \mathrm{Mj}_1, \mathrm{Mj}_2, \mathrm{Mn}_3 \rangle$ as follows:

$$\langle \mathrm{Mj}_1, \mathrm{Mj}_2, \mathrm{Mn}_3 \rangle(x,y,z) = \begin{cases} (x,y,z) & \text{if } x = y \wedge (y,z) \in \overline{M} \text{ or } xy|z, \quad (4.35a) \\ (z,x,y) & \text{if } z = x \wedge (x,y) \in \overline{M} \text{ or } zx|y, \quad (4.35b) \\ (y,z,x) & \text{if } y = z \wedge (z,x) \in \overline{M} \text{ or } yz|x, \quad (4.35c) \\ (x,y,z) & \text{otherwise.} \quad (4.35d) \end{cases}$$

Note that the operations of $\langle \mathrm{Mj}_1, \mathrm{Mj}_2, \mathrm{Mn}_3 \rangle$ are majorities and a minority on $\overline{M}$. Also note that in the subcase $x = y \wedge (y,z) \in \overline{M}$ of case (4.35a), the output has to be (x,y,z) for $\langle \mathrm{Mj}_1, \mathrm{Mj}_2, \mathrm{Mn}_3 \rangle$ to be an MJN multimorphism of Γ on $\overline{M}$ (and similarly for the first subcase of case (4.35b) and case (4.35c)). It is the other cases where there is some freedom and where we differ from [73].

Theorem 4.55. *$\langle \mathrm{Mj}_1, \mathrm{Mj}_2, \mathrm{Mn}_3 \rangle$ is a multimorphism of Γ.*

In fact, we prove that (2.11) holds with *equality*.

Proof. Suppose for the sake of contradiction this is not true for some r-ary weighted relation $\gamma \in \Gamma^*$ and $\mathbf{x}, \mathbf{y}, \mathbf{z} \in \mathrm{Feas}(\gamma)$; we choose γ so that it has the minimum arity among such counterexamples. We denote the r-tuples to which $\langle \mathrm{Mj}_1, \mathrm{Mj}_2, \mathrm{Mn}_3 \rangle$ maps $(\mathbf{x}, \mathbf{y}, \mathbf{z})$ by $(\mathbf{f}, \mathbf{g}, \mathbf{h})$.

First we show that case (4.35b) does not occur. Let I be the set of coordinates i such that case (4.35b) applies to (x_i, y_i, z_i) and let J cover the remaining cases. Suppose that I is non-empty, and note that $\mathbf{f}_I = \mathbf{z}_I, \mathbf{g}_I = \mathbf{x}_I, \mathbf{h}_I = \mathbf{y}_I$. For every $i \in I$, it holds $(x_i, y_i), (z_i, y_i) \in \overline{M}$ (directly or by Lemma 4.54), and either $z_i = x_i$ or $z_i x_i | y_i$.

We claim that $\{x_i, y_i, z_i\} \times \{x_j, y_j, z_j\} \subseteq \mathrm{Pr}_{i,j}(\mathrm{Feas}(\gamma))$ for all $i \in I, j \in J$. We already have $(x_i, x_j), (y_i, y_j), (z_i, z_j) \in \mathrm{Pr}_{i,j}(\mathrm{Feas}(\gamma))$. It holds

$$(x_i, y_j) \in \mathrm{Pr}_{i,j}(\mathrm{Feas}(\gamma)) \iff (y_i, x_j) \in \mathrm{Pr}_{i,j}(\mathrm{Feas}(\gamma)), \tag{4.36}$$
$$(z_i, y_j) \in \mathrm{Pr}_{i,j}(\mathrm{Feas}(\gamma)) \iff (y_i, z_j) \in \mathrm{Pr}_{i,j}(\mathrm{Feas}(\gamma)), \tag{4.37}$$

otherwise there would be a soft edge in $\overline{M}$ (i.e. soft edge $(x_i, y_i) - (y_j, x_j)$ and $(z_i, y_i) - (y_j, z_j)$ respectively).

If $(x_i, y_j), (z_i, y_j) \notin \mathrm{Pr}_{i,j}(\mathrm{Feas}(\gamma))$, then there are edges $(x_i, y_i) - (y_j, x_j)$, $(z_i, y_i) - (y_j, z_j)$, and hence $(x_j, y_j), (z_j, y_j) \in \overline{M}$. Because case (4.35b) does not apply at coordinate j, it holds $z_j \neq x_j$, and therefore labels

x_j, y_j, z_j are all distinct. But then $(x_i, z_j) \notin \mathrm{Pr}_{i,j}(\mathrm{Feas}(\gamma))$, otherwise we would get $\{(x_i, z_j), (x_i, x_j), (y_i, y_j)\} \in \Gamma^*$ (obtained by domain restriction of $\mathrm{Pr}_{i,j}(\mathrm{Feas}(\gamma))$), and thus $z_j x_j | y_j$ would hold. Analogously, we have $(z_i, x_j) \notin \mathrm{Pr}_{i,j}(\mathrm{Feas}(\gamma))$. This implies $z_i \neq x_i$, and hence $z_i x_i | y_i$ holds. By domain restriction of $\mathrm{Pr}_{i,j}(\mathrm{Feas}(\gamma))$ we obtain a bijection relation

$$\{(x_i, x_j), (y_i, y_j), (z_i, z_j)\} \in \Gamma^* \, ; \tag{4.38}$$

joining it with a binary relation showing that $z_i x_i | y_i$ gives us $z_j x_j | y_j$, which is a contradiction.

If $(x_i, y_j) \in \mathrm{Pr}_{i,j}(\mathrm{Feas}(\gamma))$ and $(z_i, y_j) \notin \mathrm{Pr}_{i,j}(\mathrm{Feas}(\gamma))$, then we have $z_i \neq x_i$, $z_i x_i | y_i$, and $(z_j, y_j) \in \overline{M}$. It must also hold $(x_i, z_j) \notin \mathrm{Pr}_{i,j}(\mathrm{Feas}(\gamma))$, otherwise there would be a soft edge incident to vertex (z_j, y_j). But then we have $\{(x_i, y_j), (y_i, y_j), (z_i, z_j)\} \in \Gamma^*$, which implies $x_i y_i | z_i$ and contradicts Lemma 4.54. The case when $(x_i, y_j) \notin \mathrm{Pr}_{i,j}(\mathrm{Feas}(\gamma))$ and $(z_i, y_j) \in \mathrm{Pr}_{i,j}(\mathrm{Feas}(\gamma))$ can be ruled out by an analogous argument.

Therefore, we have $(x_i, y_j), (z_i, y_j) \in \mathrm{Pr}_{i,j}(\mathrm{Feas}(\gamma))$. It must also hold $(x_i, z_j), (z_i, x_j) \in \mathrm{Pr}_{i,j}(\mathrm{Feas}(\gamma))$, otherwise there would be a soft edge in $\overline{M}$ (incident to vertex (x_i, y_i) and (z_i, y_i) respectively). Hence, we have shown that $\{x_i, y_i, z_i\} \times \{x_j, y_j, z_j\} \subseteq \mathrm{Pr}_{i,j}(\mathrm{Feas}(\gamma))$.

Because $\mathrm{Feas}(\gamma)$ is 2-decomposable by Lemma 4.49, we have $\mathbf{u}_I \cdot \mathbf{v}_J \in \mathrm{Feas}(\gamma)$ for any $\mathbf{u}, \mathbf{v} \in \{\mathbf{x}, \mathbf{y}, \mathbf{z}\}$. It must hold

$$\gamma(\mathbf{y}_I \cdot \mathbf{x}_J) + \gamma(\mathbf{y}_I \cdot \mathbf{y}_J) + \gamma(\mathbf{y}_I \cdot \mathbf{z}_J) = \gamma(\mathbf{y}_I \cdot \mathbf{f}_J) + \gamma(\mathbf{y}_I \cdot \mathbf{g}_J) + \gamma(\mathbf{y}_I \cdot \mathbf{h}_J) \, , \tag{4.39}$$

otherwise we would obtain a smaller counterexample by pinning γ at every coordinate $i \in I$ to its respective label y_i. This gives $\mathbf{y}_I \cdot \mathbf{f}_J, \mathbf{y}_I \cdot \mathbf{g}_J, \mathbf{y}_I \cdot \mathbf{h}_J \in \mathrm{Feas}(\gamma)$; by an analogous argument we get $\mathbf{u}_I \cdot \mathbf{v}_J \in \mathrm{Feas}(\gamma)$ for any $\mathbf{u} \in \{\mathbf{x}, \mathbf{y}, \mathbf{z}\}$ and $\mathbf{v} \in \{\mathbf{f}, \mathbf{g}, \mathbf{h}\}$. By Lemma 4.51, it holds

$$\gamma(\mathbf{x}_I \cdot \mathbf{x}_J) + \gamma(\mathbf{y}_I \cdot \mathbf{g}_J) = \gamma(\mathbf{x}_I \cdot \mathbf{g}_J) + \gamma(\mathbf{y}_I \cdot \mathbf{x}_J) \, , \tag{4.40}$$

$$\gamma(\mathbf{z}_I \cdot \mathbf{z}_J) + \gamma(\mathbf{y}_I \cdot \mathbf{f}_J) = \gamma(\mathbf{z}_I \cdot \mathbf{f}_J) + \gamma(\mathbf{y}_I \cdot \mathbf{z}_J) \, . \tag{4.41}$$

By adding (4.39), (4.40), and (4.41) we get

$$\gamma(\mathbf{x}_I \cdot \mathbf{x}_J) + \gamma(\mathbf{y}_I \cdot \mathbf{y}_J) + \gamma(\mathbf{z}_I \cdot \mathbf{z}_J) = \gamma(\mathbf{z}_I \cdot \mathbf{f}_J) + \gamma(\mathbf{x}_I \cdot \mathbf{g}_J) + \gamma(\mathbf{y}_I \cdot \mathbf{h}_J) \, , \tag{4.42}$$

and hence (2.11) holds as equality (note that $\mathbf{f}_I = \mathbf{z}_I, \mathbf{g}_I = \mathbf{x}_I, \mathbf{h}_I = \mathbf{y}_I$). This is a contradiction; therefore case (4.35b) does not apply at any coordinate.

Suppose that case (4.35c) applies at some coordinate i. $\langle \mathrm{Mj}_1, \mathrm{Mj}_2, \mathrm{Mn}_3 \rangle$ maps $(\mathbf{y}, \mathbf{x}, \mathbf{z})$ to $(\mathbf{g}, \mathbf{f}, \mathbf{h})$, which gives us another smallest counterexample to the theorem. However, at coordinate i is now applied case (4.35b), which was proved impossible.

Finally, we have that only cases (4.35a) and (4.35d) may occur in a smallest counterexample. But then $\langle \mathrm{Mj}_1, \mathrm{Mj}_2, \mathrm{Mn}_3 \rangle$ maps $(\mathbf{x}, \mathbf{y}, \mathbf{z})$ to $(\mathbf{x}, \mathbf{y}, \mathbf{z})$, and hence the stated equality holds. $\square$

Chapter 5

Surjective VCSP

The well-known (s,t)-Min-Cut problem [93] can be expressed in the framework of the language-restricted VCSP quite straightforwardly: Let $D = \{0,1\}$ be the domain; its labels represent the two parts of a cut. For every vertex of a given graph, we introduce a corresponding variable. We impose constraints $\rho_0(s)$ and $\rho_1(t)$ to ensure that vertices s and t end up in different parts. In order to determine the number of edges that are cut, we impose a constraint $\gamma_=(u,v)$ for every edge $\{u,v\}$ of the graph, where $\gamma_= = \mathrm{Soft}(\rho_=)$. Clearly, optimal solutions to such an instance of $\mathrm{VCSP}(\{\rho_0,\rho_1,\gamma_=\})$ correspond to minimum (s,t)-cuts in the given graph. (See also Example 2.11.)

The (global) Min-Cut problem [93], however, evades such a natural representation. Indeed, applying the same approach as for the (s,t)-Min-Cut problem fails, as we do not know which pair of vertices to choose as s and t. If we do not force any two vertices to be assigned different labels, an optimal solution will correspond to a trivial cut with all the vertices in one part. The *surjective* VCSP, denoted by VCSP_s, enables us to circumvent this issue by imposing an additional global constraint: A labelling $s : V \to D$ must be surjective, i.e., every label must be assigned to at least one variable. Hence, the Min-Cut problem can be expressed by $\mathrm{VCSP}_\mathrm{s}(\{\gamma_=\})$.

Analogously to the VCSP, it is natural to investigate the computational complexity of the VCSP_s restricted to a constraint language. It comes as no surprise that, for a fixed language, the complexity of these two frameworks is closely related. Specifically, the VCSP reduces to the VCSP_s, and the VCSP_s reduces to the VCSP if the language contains all the constants, i.e., unary relations ρ_d for all $d \in D$ (see Lemmas 5.2 and 5.3). Consequently, only languages that are tractable (for the VCSP) but do not include all the constants are interesting to study in the surjective setting, as they may become intractable when the surjectivity requirement is imposed.

The understanding of the surjective setting is, compared to the general

VCSP, far less advanced. For instance, the algebraic approach (see Chapter 3), which has proved extremely successful in the study of the VCSP, saw little progress after Chen's initial attempt to extend it to the surjective CSP [23] (see [79] for an application of Chen's result). Another obstacle to a complete classification are problems of open complexity that are captured by the VCSP$_\mathrm{s}$, for example the 3-No-Rainbow-Colouring problem [7]. This special case of the surjective CSP on a domain of size 3 asks whether the vertices of a given 3-uniform hypergraph can be surjectively labelled using 3 colours so that no hyperedge contains vertices of all the colours.

For Boolean languages, a complexity classification has been attained in some special cases. For $\{0, \infty\}$-valued languages (i.e., the surjective CSP), Creignou and Hébrard [30] proved that all the interesting languages (in the sense described above) are in fact intractable in the surjective setting (see Theorem 5.7). In particular, the constants ρ_0, ρ_1 admit four out of the six polymorphisms characterising the tractable classes of the Boolean CSP (see Theorem 2.32), namely min, max, Mn, and Mj. The remaining two classes, i.e. languages invariant under c_0 or c_1, are, without the surjectivity constraint, trivially tractable by assigning all the variables label 0 or 1 respectively. However, it is NP-hard to find a surjective feasible assignment.

Uppman [102] established a complexity classification of Boolean $\{0, 1\}$-valued languages. As in the case of $\{0, \infty\}$-valued languages, the surjectivity constraint reduces the two trivially tractable classes that exclude the constants, namely languages improved by multimorphism $\langle c_0 \rangle$ or $\langle c_1 \rangle$ (see Theorem 2.31). However, Uppman discovered that two subclasses (called almost-min-min and dual almost-min-min) maintain their tractability even in the surjective setting. The algorithm developed in [102] solves them by finding all near-optimal solutions to a generalisation of the Min-Cut problem.

In this chapter, we extend the previously known classifications of Boolean $\{0, \infty\}$- and $\{0, 1\}$-valued languages and encompass all Boolean ($\overline{\mathbb{Q}}$-valued) languages. According to our classification (see Theorem 5.11), the two trivially tractable classes that exclude the constants, namely languages improved by multimorphism $\langle c_0 \rangle$ or $\langle c_1 \rangle$ (see Theorem 2.30), contain two subclasses for which the VCSP$_\mathrm{s}$ is tractable. We call the condition characterising these subclasses *EDS*, which stands for *essentially a downset*. A relation that is essentially a downset can be obtained from a downset by introducing redundant copies of its coordinates (see Definition 5.18). A *finite* EDS language consists of weighted relations such that their Feas and Opt relations are essentially downsets; for the general definition, see Definition 5.10. The notion of EDS generalises that of almost-min-min [102]. Our main result also implies a complexity classification of Boolean $\mathbb{Q}_{\geq 0}$-valued languages for the surjective Max-VCSP problem with respect to approximation (see Section 5.2.3). This generalises the classification of Boolean $\{0,1\}$-valued languages by Bach and Zhou [1].

As in [102], our proof of the tractability of EDS languages in the surjective setting is based on a reduction to a (further) generalisation of the Min-Cut problem. The objective in this problem is to minimise the sum of the total weight of edges that are cut and a superadditive set function (see Definition 5.43). For our reduction, we need to find all α-optimal solutions (for a fixed α depending on the EDS language). As we show in Section 5.4.2, this can be achieved in polynomial time; in fact, there are at most $O(n^{20\alpha})$ such solutions, where n is the number of variables. This improves the bound of $O(n^{27^{\alpha}})$ established in [102] for a special case of the problem. Our reduction also implies that, given a $\mathrm{VCSP_s}$ instance over an EDS language, one can enumerate *all* its optimal solutions with polynomial delay [65] (see Theorem 5.13). The remaining tractable classes have this property as well, although in their case it simply follows from the fact they include the constants.

To prove that the EDS property is necessary for tractability in the surjective setting, we define a language closure operation that captures some of the possible gadget constructions while preserving the tractability of the language. This closure operation differs from the standard definition of the weighted relational clone (see Definitions 3.1 and 3.26), as minimisation may not preserve the complexity of the $\mathrm{VCSP_s}$.[1] We prove that the $\mathrm{VCSP_s}$ over a language Γ not included in any of the tractable classes is NP-hard by showing that at least one of the following cases applies: even a crisp (i.e., $\{0,\infty\}$-valued) version of Γ is intractable in the surjective setting, or the constants ρ_0, ρ_1 can be simulated over Γ ($\Gamma \cup \{\rho_0, \rho_1\}$ is intractable), or the NP-hard Minimum Distance problem[2] [104] can be reduced to $\mathrm{VCSP_s}(\Gamma)$.

The tractability of a constraint language is commonly defined in terms of its finite subsets: A language Γ is tractable if, for every finite $\Gamma' \subseteq \Gamma$, $\mathrm{VCSP}(\Gamma')$ can be solved in polynomial time (see Definition 2.7). Alternatively, one could adopt a stronger notion of *global tractability* [15], which requires that $\mathrm{VCSP}(\Gamma)$ be solvable in polynomial time. In the case of the VCSP, all tractable classes of languages are also globally tractable; however, we show that the situation is different in the surjective setting. We give an example of an infinite non-EDS language Γ such that every finite subset $\Gamma' \subseteq \Gamma$ is EDS (see Example 5.23). While $\mathrm{VCSP_s}(\Gamma)$ is NP-hard, every $\mathrm{VCSP_s}(\Gamma')$ can be solved in polynomial time. To draw this distinction, we consider both notions of tractability. Hence, we take into account infinite constraint languages as

---

[1] This is also an issue for the aforementioned algebraic approach.

[2] The Minimum Distance problem asks to find a smallest *non-zero* vector in a linear subspace over the field $\{0,1\}$.

well, but only if they are of bounded arity. Weighted relations in an instance are assumed to be represented explicitly (by tables of values).[3]

Section 5.1 introduces tools that are applicable to the surjective VCSP in general; the rest of the chapter deals only with a Boolean domain $D = \{0, 1\}$. In Section 5.2, we present our results. The hardness part of our classification is proved in Section 5.3, and the tractability in Section 5.4.

5.1 Preliminaries

In this chapter, we consider only languages of bounded arity.

Definition 5.1. A language Γ is called *s-tractable* if, for every finite $\Gamma' \subseteq \Gamma$, $\mathrm{VCSP_s}(\Gamma')$ can be solved in polynomial time. If $\mathrm{VCSP_s}(\Gamma)$ can be solved in polynomial time, language Γ is called *globally s-tractable.*

If there exists a finite $\Gamma' \subseteq \Gamma$ such that $\mathrm{VCSP_s}(\Gamma')$ is NP-hard, language Γ is called *s-intractable.* If $\mathrm{VCSP_s}(\Gamma)$ is NP-hard, language Γ is called *globally s-intractable.*

Note that a globally s-tractable language is s-tractable, and an s-intractable language is globally s-intractable.

Lemmas 5.2 and 5.3 establish a relation between the complexity of the VCSP and $\mathrm{VCSP_s}$.

Lemma 5.2. *For any constraint language* Γ*,*

$$\mathrm{VCSP}(\Gamma) \leq_p \mathrm{VCSP_s}(\Gamma)\,. \tag{5.1}$$

Proof. Given an instance I of $\mathrm{VCSP}(\Gamma)$, we construct an instance I' of $\mathrm{VCSP_s}(\Gamma)$ by adding $|D|$ extra variables. Any solution to I can be extended to a surjective solution to I' of the same value and, conversely, any (surjective) solution to I' induces a solution to I of the same value. □

Lemma 5.3. *Let* $\mathcal{C}_D$ *be the set of constants on* D*, i.e.,* $\mathcal{C}_D = \{\rho_d \mid d \in D\}$*. For any constraint language* Γ*,*

$$\mathrm{VCSP_s}(\Gamma) \leq_p \mathrm{VCSP}(\Gamma \cup \mathcal{C}_D)\,. \tag{5.2}$$

---

[3] The bounded arity restriction is vital in some of our proofs. Also, unbounded arity presents new challenges to complexity classification. For example, explicitly representing a weighted relation of an arity that is super-logarithmic in the number of variables requires a super-polynomial space.

Proof. Given an instance $I = (V, D, \phi_I)$ of $\mathrm{VCSP_s}(\Gamma)$, we iterate through all $O\left(|V|^{|D|}\right)$ injective mappings $f : D \to V$. For each mapping f, we construct an instance I'_f of $\mathrm{VCSP}(\Gamma \cup \mathcal{C}_D)$ by adding constraints $\rho_d(f(d))$ for all $d \in D$. The additional constraints guarantee that only surjective solutions to I'_f are feasible. Conversely, any surjective solution to I is a feasible solution to I'_f for some mapping f. Therefore, a solution of the smallest value among optimal solutions to I'_f for all f is an optimal surjective solution to I. $\square$

Corollary 5.4. *Any (globally) tractable language* Γ *with* $\mathcal{C}_D \subseteq \Gamma$ *is also (globally) s-tractable.*

In contrast to the VCSP, minimisation may not preserve the complexity of the $\mathrm{VCSP_s}$. Hence, we need to define a language closure that excludes this operation (cf. Definitions 3.1 and 3.26).

Definition 5.5. A constraint language Γ is called *closed* if it is closed under addition, coordinate mapping, non-negative scaling, addition of a constant, operation Opt, and, for all $d \in D$ such that $\rho_d \in \Gamma$, d-pinning.

We define Γ^* to be the smallest closed language containing Γ.

Note that we require a language to be closed under d-pinning *only* if it contains ρ_d. A weighted relational clone that contains ρ_d is also closed under d-pinning, as that can be achieved by adding ρ_d and minimising. Therefore, $\Gamma^* \subseteq \mathrm{wRelClone}_{\mathbb{R}}(\Gamma)$.

Now we show that these closure operations preserve the complexity of the $\mathrm{VCSP_s}$.

Lemma 5.6. *For any constraint language* Γ,

$$\mathrm{VCSP_s}(\Gamma^*) \leq_p \mathrm{VCSP_s}(\Gamma)\,. \tag{5.3}$$

Proof. For most of the closure operations, standard reductions for the VCSP (see Section 3.3.1) apply to the surjective setting as well. In case of pinning, however, we need a reduction that does not rely on minimisation.

Suppose that $\rho_d \in \Gamma$. Let γ' be a d-pinning of a weighted relation $\gamma \in \Gamma$; without loss of generality, let it be a pinning at the first coordinate. We show that $\mathrm{VCSP_s}(\Gamma \cup \{\gamma'\}) \leq_p \mathrm{VCSP_s}(\Gamma)$.

Let $I = (V, D, \phi_I)$ be an instance of $\mathrm{VCSP_s}(\Gamma \cup \{\gamma'\})$ with $V = \{x_1, \ldots, x_n\}$. For every $i \in [n]$, we construct an instance $I_i = (V, D, \phi_{I_i})$ of $\mathrm{VCSP_s}(\Gamma)$ by replacing all constraints of the form $\gamma'(\mathbf{x}_j)$ with $\gamma(x_i, \mathbf{x}_j)$ and adding a constraint $\rho_d(x_i)$ to force variable x_i to take label d. A solution of the smallest value among optimal solutions to $I_1, \ldots, I_n$ is an optimal solution to I, as at least one variable needs to be assigned label d. $\square$

Unless explicitly stated otherwise, we consider only Boolean languages (i.e., $D = \{0, 1\}$) in the rest of the chapter.

Our hardness proof relies on the following classification of $\{0, \infty\}$-valued languages by Creignou and Hébrard [30].

Theorem 5.7 ([30])**.** *Let* Γ *be a Boolean* $\{0, \infty\}$*-valued language. Then* Γ *is s-tractable if it is invariant under any of the following operations:* min*,* max*,* Mn*,* Mj*. Otherwise,* Γ *is s-intractable.*

Remark 5.8. We also rely on the classification of Boolean VCSP (see Theorems 2.30 and 2.31), which is stated only for the weaker notion of tractability (i.e., for finite languages) in [28]. However, the proofs there actually establish the same classification for the stronger notion of global tractability as well.

In particular, all the tractable cases (given by the eight multimorphisms stated in Theorem 2.30) are globally tractable. Conversely, any globally intractable language is also intractable: If a language Γ does not admit any of the eight multimorphisms, then there exists a finite subset $\Gamma' \subseteq \Gamma$ with $|\Gamma'| \leq 8$ that does not admit any of the eight multimorphisms (as a single weighted relation suffices to violate a multimorphism).

5.2 Results

We present our results in three parts: Section 5.2.1 defines the EDS property and states the main classification theorem, Section 5.2.2 focuses on finite EDS languages, and Section 5.2.3 gives a classification in terms of approximability for the surjective Max-VCSP.

5.2.1 Boolean surjective VCSP

We first define the property EDS (which stands for *essentially a downset*, see Definition 5.18) characterising the newly discovered tractable class of weighted relations.

Let sub (short for subtraction) be a binary operation defined by $\mathrm{sub}(x, y) = \min(x, \neg y)$. For $\mathbf{x} = (x_1, \ldots, x_r), \mathbf{y} = (y_1, \ldots, y_r) \in D^r$, we define $\mathbf{x} \leq \mathbf{y}$ if and only if $x_i \leq y_i$ for all $i \in [r]$. (Recall that $D = \{0, 1\}$ and $0 < 1$.)

Lemma 5.9. *If a weighted relation admits polymorphism* sub*, then it also admits polymorphisms* c_0 *and* min*.*

Proof. For every $x, y \in D$, it holds $c_0(x) = 0 = \mathrm{sub}(x, x)$ and $\min(x, y) = \mathrm{sub}(x, \mathrm{sub}(x, y))$. □

Definition 5.10. For any $\alpha \geq 1$, an r-ary weighted relation γ is α-*EDS* if, for every $\mathbf{x}, \mathbf{y} \in \operatorname{Feas}(\gamma)$, it holds $\mathbf{0}^r \in \operatorname{Feas}(\gamma)$ and

$$\alpha \cdot (\gamma(\mathbf{x}) + \gamma(\mathbf{y}) - 2 \cdot \gamma(\mathbf{0}^r)) \geq \gamma(\operatorname{sub}(\mathbf{x}, \mathbf{y})) - \gamma(\mathbf{0}^r)\,. \tag{5.4}$$

A weighted relation is *EDS* if it is α-EDS for some $\alpha \geq 1$. A language is EDS if there exists $\alpha \geq 1$ such that every weighted relation in the language is α-EDS.

Although this definition does not involve the notion of (weighted) polymorphisms, it is stated in a similar vein. Let f be a binary operation defined by $f(x, y) = 0$; then the requirement "for every $\mathbf{x}, \mathbf{y} \in \operatorname{Feas}(\gamma)$, it holds $\mathbf{0}^r \in \operatorname{Feas}(\gamma)$" translates to "$\gamma$ is invariant under f" (or, equivalently, "γ is invariant under c_0").[4][5] In the case of $\alpha = 1$, inequality (5.4) translates to that of admitting multimorphism $\langle \operatorname{sub}, f \rangle$. Note also that any EDS weighted relation admits polymorphism sub.

For more intuition behind this notion in the general case, see the corresponding definition of EDS for set functions (Definition 5.53) in Section 5.4.3. Finite EDS languages admit a simpler equivalent definition, see Corollary 5.21.

The following classification of $\overline{\mathbb{Q}}$-valued languages is our main result.

Theorem 5.11. *Let Γ be a Boolean $\overline{\mathbb{Q}}$-valued language. Then Γ is globally s-tractable if it is EDS, or $\neg(\Gamma)$ is EDS, or Γ admits any of the following multimorphisms:* $\langle \min, \min \rangle$, $\langle \max, \max \rangle$, $\langle \min, \max \rangle$, $\langle \mathrm{Mn}, \mathrm{Mn}, \mathrm{Mn} \rangle$, $\langle \mathrm{Mj}, \mathrm{Mj}, \mathrm{Mj} \rangle$, $\langle \mathrm{Mj}, \mathrm{Mj}, \mathrm{Mn} \rangle$. *Otherwise, Γ is globally s-intractable.*

Proof. The global s-tractability of languages admitting any of the six multimorphisms in the statement of the theorem follows from Theorem 2.30 (see Remark 5.8) by Lemma 5.3. The global s-tractability of EDS languages (whether Γ or $\neg(\Gamma)$, which is symmetric) follows from Theorem 5.59, proved in Section 5.4. Finally, the global s-intractability of the remaining languages follows from Theorem 5.39, proved in Section 5.3. $\square$

Theorem 5.11 gives us also a classification in terms of s-tractability. As noted above, any globally s-tractable language is s-tractable. Consider now a globally s-intractable language Γ. It does not admit any of the six multimorphisms, and hence there exists a finite subset of Γ that does not admit them either (see Remark 5.8). If there exists a finite subset $\Gamma' \subseteq \Gamma$ such that neither Γ' nor $\neg(\Gamma')$ is EDS, then Γ is s-intractable; otherwise Γ is s-tractable. Equivalently (by Corollary 5.21), Γ is s-intractable if neither $\operatorname{Feas}(\Gamma) \cup \operatorname{Opt}(\Gamma)$

---

[4] In fact, any EDS weighted relation admits multimorphism $\langle c_0 \rangle$ (see Lemma 5.15).

[5] Note that the unary empty relation $\rho_\emptyset$ is vacuously α-EDS for all $\alpha \geq 1$, as $\operatorname{Feas}(\rho_\emptyset) = \emptyset$.

nor $\mathrm{Feas}(\neg(\Gamma)) \cup \mathrm{Opt}(\neg(\Gamma))$ admit polymorphism sub, and it is s-tractable otherwise.

To see how EDS languages fit into the classification of $\{0, \infty\}$-valued languages established in Theorem 5.7, note the following. Any $\{0, \infty\}$-valued language of bounded arity is finite. By Corollary 5.21, any EDS $\{0, \infty\}$-valued language admits polymorphism sub, and hence also polymorphism min (by Lemma 5.9).

Theorem 5.11 implies a tighter classification for $\mathbb{Q}$-valued languages (the only reasons for global s-tractability are EDS and submodularity):

Theorem 5.12. *Let Γ be a Boolean $\mathbb{Q}$-valued language. Then Γ is globally s-tractable if it is EDS, or $\neg(\Gamma)$ is EDS, or Γ admits the $\langle \min, \max \rangle$ multimorphism. Otherwise, Γ is globally s-intractable.*

Proof. We need to show that in the case of $\mathbb{Q}$-valued languages, the remaining globally s-tractable classes from Theorem 5.11 (which are characterised by polymorphisms $\langle \min, \min \rangle$, $\langle \max, \max \rangle$, $\langle \mathrm{Mn}, \mathrm{Mn}, \mathrm{Mn} \rangle$, $\langle \mathrm{Mj}, \mathrm{Mj}, \mathrm{Mj} \rangle$, and $\langle \mathrm{Mj}, \mathrm{Mj}, \mathrm{Mn} \rangle$) collapse.

If a $\mathbb{Q}$-valued r-ary weighted relation γ admits the $\langle \min, \min \rangle$ multimorphism, then it holds $\gamma(\mathbf{x}) \geq \gamma(\mathbf{y})$ for all $\mathbf{x} \geq \mathbf{y}$. This implies that, for all $\mathbf{x}, \mathbf{y} \in \mathrm{Feas}(\gamma)$, it holds $\gamma(\mathbf{x}) \geq \gamma(\mathrm{sub}(\mathbf{x}, \mathbf{y}))$ and $\gamma(\mathbf{y}) \geq \gamma(\mathbf{0}^r)$. Hence, γ is 1-EDS. If γ admits the $\langle \max, \max \rangle$ multimorphism, then $\neg(\gamma)$ admits the $\langle \min, \min \rangle$ multimorphism. Therefore, if a $\mathbb{Q}$-valued language Γ admits $\langle \min, \min \rangle$ or $\langle \max, \max \rangle$ as a multimorphism, then Γ or $\neg(\Gamma)$ is EDS.

Multimorphisms $\langle \mathrm{Mn}, \mathrm{Mn}, \mathrm{Mn} \rangle$, $\langle \mathrm{Mj}, \mathrm{Mj}, \mathrm{Mj} \rangle$, and $\langle \mathrm{Mj}, \mathrm{Mj}, \mathrm{Mn} \rangle$ are covered by the $\langle \min, \max \rangle$ multimorphism: Weighted relations that admit $\langle \mathrm{Mn}, \mathrm{Mn}, \mathrm{Mn} \rangle$ or $\langle \mathrm{Mj}, \mathrm{Mj}, \mathrm{Mj} \rangle$ as a multimorphism are crisp [28, Propositions 6.20 and 6.22], and hence, in the $\mathbb{Q}$-valued case, they are constant functions. $\mathbb{Q}$-valued weighted relations that admit the $\langle \mathrm{Mj}, \mathrm{Mj}, \mathrm{Mn} \rangle$ multimorphism are modular [28, Corollary 6.26], and hence they are submodular. □

Enumerating all optimal solutions to an instance with polynomial delay is a fundamental problem [65, 105] studied in the context of CSP [36, 19]. An algorithm outputting a sequence of solutions works with polynomial delay if the time it takes to output the first solution as well as the time it takes between every two consecutive solutions is bounded by a polynomial in the input size.

It is known that, for a tractable constraint language Γ that includes the constants $\mathcal{C}_D$, one can enumerate all optimal solutions with polynomial delay [24]. Our results imply that the newly discovered globally s-tractable EDS languages enjoy the same property (despite *not* including the constants).

Theorem 5.13. *Let Γ be a Boolean $\overline{\mathbb{Q}}$-valued language. If Γ is globally s-tractable then there is a polynomial-delay algorithm that enumerates all optimal solutions to any instance of* $\mathrm{VCSP_s}(\Gamma)$.

The theorem is proved in Section 5.4.3.

5.2.2 Finite EDS languages

The EDS property can be described in a simpler way for languages of finite size; see the following observation and Corollary 5.21.

Observation 5.14. *A language of finite size is EDS if and only if it consists of EDS weighted relations.*

In the following we prove several useful properties EDS weighted relations.

Lemma 5.15. *Any EDS weighted relation admits multimorphism $\langle c_0 \rangle$.*

Proof. Let γ be an r-ary α-EDS weighted relation. For any $\mathbf{x} \in \mathrm{Feas}(\gamma)$, it holds $\mathbf{0}^r \in \mathrm{Feas}(\gamma)$ and

$$\alpha \cdot (2 \cdot \gamma(\mathbf{x}) - 2 \cdot \gamma(\mathbf{0}^r)) \geq \gamma(\mathrm{sub}(\mathbf{x}, \mathbf{x})) - \gamma(\mathbf{0}^r) = 0 \tag{5.5}$$

as $\mathrm{sub}(\mathbf{x}, \mathbf{x}) = \mathbf{0}^r$, and therefore $\gamma(\mathbf{x}) \geq \gamma(\mathbf{0}^r)$. □

Lemma 5.16. *A crisp weighted relation is EDS if and only if it admits polymorphism* sub.

Proof. Any EDS weighted relation admits polymorphism sub. For the converse implication, note that any crisp weighted relation that admits polymorphism sub (and thus, by Lemma 5.9, also polymorphism c_0) satisfies (5.4) for any $\alpha \geq 1$. □

Lemma 5.17. *A weighted relation γ is EDS if and only if both* $\mathrm{Feas}(\gamma)$ *and* $\mathrm{Opt}(\gamma)$ *are EDS.*

Proof. Let γ be an r-ary α-EDS weighted relation. For any $\mathbf{x}, \mathbf{y} \in \mathrm{Feas}(\gamma)$, it holds $\mathbf{0}^r \in \mathrm{Feas}(\gamma)$ and

$$\infty > \alpha \cdot (\gamma(\mathbf{x}) + \gamma(\mathbf{y}) - 2 \cdot \gamma(\mathbf{0}^r)) \geq \gamma(\mathrm{sub}(\mathbf{x}, \mathbf{y})) - \gamma(\mathbf{0}^r)\,, \tag{5.6}$$

and hence $\mathrm{sub}(\mathbf{x}, \mathbf{y}) \in \mathrm{Feas}(\gamma)$. By Lemma 5.16, $\mathrm{Feas}(\gamma)$ is EDS. Similarly, for any $\mathbf{x}, \mathbf{y} \in \mathrm{Opt}(\gamma)$, it holds $\mathbf{0}^r \in \mathrm{Opt}(\gamma)$ (by Lemma 5.15) and

$$0 = \alpha \cdot (\gamma(\mathbf{x}) + \gamma(\mathbf{y}) - 2 \cdot \gamma(\mathbf{0}^r)) \geq \gamma(\mathrm{sub}(\mathbf{x}, \mathbf{y})) - \gamma(\mathbf{0}^r)\,; \tag{5.7}$$

therefore $\operatorname{sub}(\mathbf{x},\mathbf{y}) \in \operatorname{Opt}(\gamma)$ and $\operatorname{Opt}(\gamma)$ is EDS.

To prove the converse implication, let us assume that $\operatorname{Feas}(\gamma)$, $\operatorname{Opt}(\gamma)$ are EDS and consider any $\mathbf{x},\mathbf{y} \in \operatorname{Feas}(\gamma)$. As $\operatorname{Opt}(\gamma)$ admits polymorphism c_0, it holds $\mathbf{0}^r \in \operatorname{Opt}(\gamma) \subseteq \operatorname{Feas}(\gamma)$. Therefore, the left-hand side of (5.4) is non-negative. Moreover, if it equals 0, then $\mathbf{x},\mathbf{y} \in \operatorname{Opt}(\gamma)$, and hence $\operatorname{sub}(\mathbf{x},\mathbf{y}) \in \operatorname{Opt}(\gamma)$ and the right-hand side equals 0 as well. Therefore, (5.4) holds for large enough α, as there are only finitely many choices of $\mathbf{x},\mathbf{y} \in \operatorname{Feas}(\gamma)$. ☐

We show that relations invariant under sub have a simple structure.

Definition 5.18. An r-ary relation ρ is a *downset* if, for any r-tuples $\mathbf{x},\mathbf{y}$ such that $\mathbf{x} \geq \mathbf{y}$ and $\mathbf{x} \in \rho$, it holds $\mathbf{y} \in \rho$.

An r-ary relation ρ is *essentially a downset* if it can be written as the sum of a downset and binary equality relations, i.e.,

$$\rho(x_1,\dots,x_r) = \rho'\left(x_{\pi(1)},\dots,x_{\pi(r')}\right) + \sum_{i=r'+1}^{r} \rho_{=}\left(x_{\pi(i)}, x_{a_i}\right) , \tag{5.8}$$

where $r' \leq r$, ρ' is a downset, π is a permutation of $[r]$, and $a_i \in \{\pi(1),\dots,\pi(r')\}$ for every $r'+1 \leq i \leq r$. In other words, removing duplicate coordinates[6] of ρ results in a downset.

Example 5.19. Relation $\rho' = \{(0,0),(0,1),(1,0)\}$ is a downset, while $\rho = \{(0,0,0),(0,1,1),(1,0,0)\}$ is only essentially a downset (as $\rho(x,y,z) = \rho'(x,y) + \rho_{=}(y,z)$).

Lemma 5.20. *A relation is essentially a downset if and only if it admits polymorphism* sub*.*

Proof. For any r-ary relation ρ that is essentially a downset and $\mathbf{x},\mathbf{y} \in \rho$, we prove that $\mathbf{z} = \operatorname{sub}(\mathbf{x},\mathbf{y}) \in \rho$. Let $\mathbf{x} = (x_1,\dots,x_r)$, $\mathbf{y} = (y_1,\dots,y_r)$, $\mathbf{z} = (z_1,\dots,z_r)$. It holds $\mathbf{x} \geq \mathbf{z}$. Moreover, for any coordinates i,j such that $x_i = x_j$ and $y_i = y_j$, it holds $z_i = z_j$. Since ρ can be written as a sum of a downset and equality relations, we have $\mathbf{z} \in \rho$.

We prove the converse implication by contradiction. Suppose that ρ is a smallest-arity relation that admits polymorphism sub but is not essentially a downset; let us denote its arity by r. If there are distinct coordinates i,j such that $z_i = z_j$ for all $\mathbf{z} = (z_1,\dots,z_r) \in \rho$, identifying these coordinates yields an $(r-1)$-ary relation ρ' such that ρ can be written as the sum of ρ' and a

---

[6] A coordinate i is a duplicate of a coordinate j if, for every $(x_1,\dots,x_r) \in \rho$, it holds $x_i = x_j$.

binary equality relation. However, ρ' also admits sub, and hence is essentially a downset by the choice of ρ, which implies that ρ is essentially a downset as well. Therefore, for any distinct coordinates i, j, there exists $\mathbf{z}^{(i,j)} \in \rho$ with $z_i^{(i,j)} \neq z_j^{(i,j)}$.

As ρ is not a downset, for some r-tuples $\mathbf{x}, \mathbf{y}$ it holds $\mathbf{x} \geq \mathbf{y}$, $\mathbf{x} \in \rho$, $\mathbf{y} \notin \rho$. We may assume without loss of generality that, for some $n \in [r]$, the set of coordinates with label 1 equals $[n]$ for $\mathbf{x}$ and $[n-1]$ for $\mathbf{y}$. Let $\mathbf{e} = (e_1, \ldots, e_r) \in \rho$ be a tuple with the smallest number of coordinates labelled 1 such that $e_n = 1$. We claim that $e_i = 0$ for all $i \neq n$: Otherwise, either $\operatorname{sub}\left(\mathbf{e}, \mathbf{z}^{(i,n)}\right) = \min\left(\mathbf{e}, \neg(\mathbf{z}^{(i,n)})\right)$ or $\operatorname{sub}\left(\mathbf{e}, \operatorname{sub}\left(\mathbf{e}, \mathbf{z}^{(i,n)}\right)\right) = \min\left(\mathbf{e}, \mathbf{z}^{(i,n)}\right)$ contradicts the minimality of $\mathbf{e}$. But then $\operatorname{sub}(\mathbf{x}, \mathbf{e}) = \mathbf{y} \in \rho$, which is a contradiction. □

Corollary 5.21. *Let Γ be a finite language. The following conditions are equivalent.*

1. *Language Γ is EDS.*
2. *For every $\gamma \in \Gamma$, weighted relation γ is EDS.*
3. *For every $\gamma \in \Gamma$, both* $\operatorname{Feas}(\gamma)$ *and* $\operatorname{Opt}(\gamma)$ *admit polymorphism* sub.
4. *For every $\gamma \in \Gamma$, both* $\operatorname{Feas}(\gamma)$ *and* $\operatorname{Opt}(\gamma)$ *are essentially downsets.*

Remark 5.22. In [53], a weighted relation γ is called PDS if both $\operatorname{Feas}(\gamma)$ and $\operatorname{Opt}(\gamma)$ are essentially downsets. For a $\{0,1\}$-valued weighted relation, this condition is equivalent to that of being almost-min-min [102]. By Corollary 5.21, PDS and EDS are equivalent concepts for languages of finite size.

As we show in the following example, there exists an infinite non-EDS language Γ such that every finite subset $\Gamma' \subseteq \Gamma$ is EDS. Hence, Γ is s-tractable, although it is globally s-intractable ($\text{VCSP}_\text{s}(\Gamma)$ is NP-hard).

Example 5.23. For any $w \in \mathbb{Z}_{\geq 1}$, we define a ternary weighted relation μ_w on $D = \{0,1\}$ by

$$\mu_w(x,y,z) = \begin{cases} 2 & \text{if } z = 1 \text{ and } x = y, \\ 1 & \text{if } z = 1 \text{ and } x \neq y, \\ 0 & \text{if } z = 0 \text{ and } x = y = 0, \\ w & \text{otherwise.} \end{cases} \tag{5.9}$$

Note that $\operatorname{Feas}(\mu_w) = D^3$ and $\operatorname{Opt}(\mu_w) = \{(0,0,0)\}$ are downsets, and hence μ_w is EDS. However, it is not α-EDS for any $\alpha < w/2$: For $\mathbf{x} = (0,1,1)$,

$\mathbf{y} = (1, 0, 1)$, we have $\mu_w(\mathbf{x}) + \mu_w(\mathbf{y}) = 2$ but $\mu_w(\text{sub}(\mathbf{x}, \mathbf{y})) = \mu_w(0, 1, 0) = w$. Language $\Gamma = \{\mu_w \mid w \in \mathbb{Z}_{\geq 1}\}$ is therefore not EDS.

By our classification (Theorem 5.11), language Γ is globally s-intractable; here we show it directly by a reduction from the NP-hard Max-Cut problem. Given an undirected graph $G = (V, E)$ with no isolated vertices, we construct a $\text{VCSP}_\text{s}(\Gamma)$ instance I as follows. Let $w = 2|E| + 1$. We introduce a corresponding variable for every vertex in V, and add a special variable z. For every edge $\{x, y\} \in E$, we impose a constraint $\mu_w(x, y, z)$.

Cuts in G are in one-to-one correspondence with assignments to I satisfying $z = 1$. In particular, a cut of size k corresponds to an assignment to I with value $k + 2(|E| - k) = 2|E| - k$. Any surjective assignment with $z = 0$ is of value at least $w > 2|E| - k$. Thus, solving I amounts to solving Max-Cut in G.

5.2.3 Approximability of maximising surjective VCSP

Although the VCSP is commonly defined with a minimisation objective, it is easy to see that, for exact solvability, its maximisation variant is essentially an identical problem: Minimising a $\mathbb{Q}$-valued function ϕ_I corresponds to maximising $-\phi_I$. When studying approximability, however, the two variants vastly differ (see [81] for a survey).

We focus on maximisation of the $\mathbb{Q}_{\geq 0}$-valued VCSP. This problem generalises the Max-CSP, in which the objective is to maximise the number of satisfied constraints; in particular, the Max-CSP corresponds to maximisation of the $\{0, 1\}$-valued VCSP. The complexity of exactly maximising the $\mathbb{Q}_{\geq 0}$-valued VCSP was established by Thapper and Živný [100]. Raghavendra [89] showed that, assuming the unique games conjecture, the basic semidefinite programming relaxation achieves the optimal approximation ratio for the problem. In this section, we consider approximate maximisation of the *surjective* $\mathbb{Q}_{\geq 0}$-valued VCSP.

Definition 5.24. An *instance* $I = (V, D, \phi_I)$ of the Max-VCSP on domain D is given by a finite set of variables $V = \{x_1, \ldots, x_n\}$ and an objective function $\phi_I : D^n \to \mathbb{Q}_{\geq 0}$ expressed as a weighted sum of *constraints* over V, i.e.,

$$\phi_I(x_1, \ldots, x_n) = \sum_{i=1}^{q} w_i \cdot \gamma_i(\mathbf{x}_i)\,, \tag{5.10}$$

where γ_i is a $\mathbb{Q}_{\geq 0}$-valued weighted relation, $w_i \in \mathbb{Q}_{\geq 0}$ is the *weight* and $\mathbf{x}_i \in V^{\text{ar}(\gamma_i)}$ the *scope* of the ith constraint.

Given an instance I, the goal is to find an assignment $s : V \to D$ of domain labels to the variables that *maximises* ϕ_I. We denote the maximum value of the objective function by opt_I. For any $r \in [0,1]$, an assignment s is an r-*approximate solution* to I if $\phi_I(s) \geq r \cdot \mathrm{opt}_I$.

An assignment s is *surjective* if its image equals D. We denote the maximum objective value of surjective assignments by s-opt_I. For any $r \in (0,1]$, a surjective assignment s is an r-*approximate surjective solution* to I if $\phi_I(s) \geq r \cdot \text{s-opt}_I$.

We denote by $\text{Max-VCSP}_\text{s}(\Gamma)$ the surjective Max-VCSP problem on instances over a language Γ.

Following the standard definitions, we say that $\text{Max-VCSP}_\text{s}(\Gamma)$ belongs to APX if, for some $r \in (0,1]$, there exists a polynomial-time algorithm that finds an r-approximate surjective solution to every $\text{Max-VCSP}_\text{s}(\Gamma)$ instance. If such an algorithm exists for every $r < 1$, we say that the problem admits a polynomial-time approximation scheme (PTAS). $\text{Max-VCSP}_\text{s}(\Gamma)$ is APX-hard if there exists a PTAS reduction (an approximation-preserving reduction, see [32]) from every problem in APX to $\text{Max-VCSP}_\text{s}(\Gamma)$.

First, we prove that a polynomial-time algorithm for exactly maximising the $\mathbb{Q}_{\geq 0}$-valued VCSP over a language Γ implies a PTAS for $\text{Max-VCSP}_\text{s}(\Gamma)$. Second, we establish a complexity classification of Boolean languages in Theorem 5.27.

Lemma 5.25. *Let Γ be a $\mathbb{Q}_{\geq 0}$-valued language and fix $r, \epsilon \in \mathbb{R}$ such that $0 < \epsilon \leq r \leq 1$. There is a polynomial-time algorithm that, given a Max-VCSP instance $I = (V, D, \phi_I)$ over Γ and an r-approximate solution s to I, outputs an $(r - \epsilon)$-approximate* surjective *solution s' to I.*

Proof. Let $a_{\max}$ denote the maximum arity of weighted relations in Γ, and n the number of variables of I. If $n < \frac{r \cdot |D| \cdot a_{\max}}{\epsilon}$, we find an optimal surjective assignment to I by trying all $O(|D|^n)$ assignments.

Otherwise, we modify the given assignment s in order to obtain a surjective assignment s'. For any variable $x \in V$, let $B_x \subseteq [q]$ be the set of indices of constraints in whose scopes x appears. We define the contribution of x by

$$c(x) = \sum_{i \in B_x} w_i \cdot \gamma_i(s(\mathbf{x}_i)) \,. \tag{5.11}$$

It follows that the total contribution of all variables is at most $a_{\max} \cdot \phi_I(s)$.

Let U be a set of $|D|$ variables with the smallest contribution. We assign to them labels D bijectively. The resulting assignment s' is surjective, and it holds

$$\begin{aligned}
\phi_I(s') &\geq \phi_I(s) - \sum_{x \in U} c(x) && (5.12)\\
&\geq \phi_I(s) - \frac{|D|}{n} \cdot a_{\max} \cdot \phi_I(s) && (5.13)\\
&\geq \left(1 - \frac{|D|}{n} \cdot a_{\max}\right) \cdot r \cdot \mathrm{opt}_I && (5.14)\\
&\geq (r - \epsilon) \cdot \text{s-opt}_I\ . && (5.15)
\end{aligned}$$

□

Applying this lemma to an *optimal* solution to an Max-VCSP instance (i.e., $r = 1$) gives us the following corollary.

Corollary 5.26. *If the* Max-VCSP *over a* $\mathbb{Q}_{\geq 0}$*-valued language* Γ *is solvable in polynomial time, then there is a PTAS for* Max-VCSP$_\mathrm{s}(\Gamma)$.

Finally, we classify Boolean $\mathbb{Q}_{\geq 0}$-valued languages by the complexity of the corresponding Max-VCSP$_\mathrm{s}$. Since multimorphisms and the EDS property are defined in the context of minimisation, the following theorem applies them to language $-\Gamma$ instead of Γ (where $-\Gamma = \{-\gamma \mid \gamma \in \Gamma\}$ and $(-\gamma)(\mathbf{x}) = -\gamma(\mathbf{x})$).

Theorem 5.27. *Let* Γ *be a Boolean* $\mathbb{Q}_{\geq 0}$*-valued language. Then*

1. Max-VCSP$_\mathrm{s}(\Gamma)$ *is solvable exactly in polynomial time if* $-\Gamma$ *is EDS, or* $-(\neg(\Gamma))$ *is EDS, or* $-\Gamma$ *admits the* $\langle \min, \max \rangle$ *multimorphism;*
2. *otherwise it is NP-hard to solve exactly, but*
   - *(a) it is in PTAS if* $-\Gamma$ *admits* $\langle c_0 \rangle$ *or* $\langle c_1 \rangle$*,*
   - *(b) and is APX-hard otherwise.*

Proof. Theorem 5.12 implies the case (1) and NP-hardness in the case (2). Case (2a) follows from Theorem 2.31 and Corollary 5.26. By Theorem 2.31, if Γ does not admit either of $\langle c_0 \rangle$, $\langle c_1 \rangle$ and $\langle \min, \max \rangle$, then Max-VCSP$(\Gamma)$ is NP-hard. The proof of Theorem 2.31 in [28] actually establishes that Max-VCSP(Γ) is APX-hard. By the approximation-preserving reduction in the proof of Lemma 5.2, this implies that Max-VCSP$_\mathrm{s}(\Gamma)$ is APX-hard as well. □

Theorem 5.27 generalises the result of Bach and Zhou [1, Theorem 16] in two respects. Firstly, we classify all $\mathbb{Q}_{\geq 0}$-valued languages as opposed to $\{0,1\}$-valued languages. Secondly, we classify constraint languages as being in P, in PTAS, or being APX-hard; [1] only distinguishes admitting a PTAS versus being APX-hard. Finally, the main technical component of Theorem 5.27, Lemma 5.25, has a slightly simpler proof compared to [1].

5.3 Hardness proof

Consider a Boolean language Γ that admits multimorphism $\langle c_0\rangle$ (the case of multimorphism $\langle c_1\rangle$ is symmetric), but does not admit any of the following multimorphisms: $\langle \min, \min\rangle$, $\langle \max, \max\rangle$, $\langle \min, \max\rangle$, $\langle \mathrm{Mn}, \mathrm{Mn}, \mathrm{Mn}\rangle$, $\langle \mathrm{Mj}, \mathrm{Mj}, \mathrm{Mj}\rangle$, $\langle \mathrm{Mj}, \mathrm{Mj}, \mathrm{Mn}\rangle$. Suppose that Γ is not EDS. We prove that $\mathrm{VCSP}_{\mathrm{s}}(\Gamma)$ is NP-hard, i.e., Γ is globally s-intractable.

We start by showing that there exists a relation such that it is not invariant under sub and it can be added to Γ without changing the complexity of $\mathrm{VCSP}_{\mathrm{s}}(\Gamma)$ (see Corollary 5.31). For finite Γ, this follows simply from Corollary 5.21 and Lemma 5.6, as there exists $\gamma \in \Gamma$ such that $\mathrm{Feas}(\gamma)$ or $\mathrm{Opt}(\gamma)$ is not invariant under sub. In general, however, a different argument is necessary. We prove it by showing that Γ contains weighted relations arbitrarily "similar" to a relation which is not invariant under sub, and that this relation may be thus added to Γ.

Definition 5.28. For any $\alpha \geq 1$, an r-ary weighted relation γ is *α-crisp* if its image $\gamma(D^r)$ lies in $[0,1] \cup (\alpha, \infty]$. We will denote by $\mathrm{Round}_\alpha(\gamma)$ the r-ary relation defined as

$$\mathrm{Round}_\alpha(\gamma)(\mathbf{x}) = \begin{cases} 0 & \text{if } \gamma(\mathbf{x}) \in [0,1], \\ \infty & \text{if } \gamma(\mathbf{x}) \in (\alpha, \infty]. \end{cases} \tag{5.16}$$

Note that an α-crisp weighted relation is α'-crisp for any $\alpha' \leq \alpha$. Moreover, a crisp weighted relation ρ is α-crisp for any $\alpha \geq 1$, and $\mathrm{Round}_\alpha(\rho) = \rho$.

Lemma 5.29. *Let Γ be a language and ρ a relation such that, for any $\alpha \geq 1$, there exists an α-crisp weighted relation $\gamma \in \Gamma$ with $\mathrm{Round}_\alpha(\gamma) = \rho$. Then $\mathrm{VCSP}_{\mathrm{s}}(\Gamma \cup \{\rho\}) \leq_p \mathrm{VCSP}_{\mathrm{s}}(\Gamma)$.*

Proof. Let I be an instance of $\mathrm{VCSP}_{\mathrm{s}}(\Gamma \cup \{\rho\})$ with k constraints that apply relation ρ. By scaling and adding constants to weighted relations in I, we ensure that all the assigned values are non-negative integers. Let $w \in \mathbb{Z}$ be a strict upper bound on the maximum value of a feasible solution to I (e.g., the weighted sum of the maximum finite values assigned by the constraints of

I, increased by one). Let $\gamma \in \Gamma$ be a $w \cdot (k+1)$-crisp weighted relation such that $\mathrm{Round}_{w\cdot(k+1)}(\gamma) = \rho$. In each constraint applying relation ρ, we replace it by γ with weight $1/(k+1)$, and thus obtain an instance I' of $\mathrm{VCSP_s}(\Gamma)$.

The value of any feasible assignment increases by at most $k/(k+1) < 1$ and the value of any infeasible assignment becomes larger than w (as γ is $w \cdot (k+1)$-crisp). Since the value of any feasible assignment to I is an integer less than w, any optimal solution to I' is also optimal for I. ☐

Lemma 5.30. *Let Γ be a language such that it admits multimorphism $\langle c_0 \rangle$ but is not EDS. Then there exists a relation ρ that is invariant under c_0 but not under* sub *and, for any $\alpha \geq 1$, there exists an α-crisp weighted relation $\gamma \in \Gamma^*$ with* $\mathrm{Round}_\alpha(\gamma) = \rho$.

Proof. We will show that for any $\alpha \geq 1$, there exists an α-crisp weighted relation $\gamma \in \Gamma^*$ such that $\mathrm{Round}_\alpha(\gamma)$ is a relation of arity at most 4 that is invariant under c_0 but not under sub. As there are only finitely many such relations, the claim of the lemma will follow.

Consider any $\alpha \geq 1$. Language Γ^* admits multimorphism $\langle c_0 \rangle$ as well but is not EDS; in particular, it is not α^{17}-EDS. Therefore, there exists an r-ary weighted relation $\gamma \in \Gamma^*$ and $\mathbf{u}, \mathbf{v} \in \mathrm{Feas}(\gamma)$ such that $\gamma(\mathbf{0}^r) = 0$ (as Γ^* is closed under adding constants) and

$$0 \leq \alpha^{17} \cdot (\gamma(\mathbf{u}) + \gamma(\mathbf{v})) < \gamma(\mathrm{sub}(\mathbf{u}, \mathbf{v}))\,. \tag{5.17}$$

We may assume that there are no distinct coordinates i, j where $u_i = u_j$ and $v_i = v_j$ (otherwise we identify them), and hence $r \leq 4$. As Γ^* is closed under scaling, we may also assume that $\gamma(\mathbf{u}), \gamma(\mathbf{v}) \leq 1$ and $\gamma(\mathrm{sub}(\mathbf{u}, \mathbf{v})) > \alpha^{17}$.

Let us consider, for any $0 \leq i \leq 16$, the intersection of the image $\gamma(D^r)$ with the interval $(\alpha^i, \alpha^{i+1}]$. Because $|D^r| \leq 2^4 = 16$, the intersection is empty for some i. Scaling γ by $1/\alpha^i$ then yields an α-crisp weighted relation $\gamma' \in \Gamma^*$ such that $\mathrm{Round}_\alpha(\gamma')$ is invariant under c_0 but not under sub, as $\gamma'(\mathbf{0}^r), \gamma'(\mathbf{u}), \gamma'(\mathbf{v}) \leq 1$ and $\gamma'(\mathrm{sub}(\mathbf{u}, \mathbf{v})) > \alpha$. ☐

Corollary 5.31. *Let Γ be a language such that it admits multimorphism $\langle c_0 \rangle$ but is not EDS. Then* $\mathrm{VCSP_s}(\Gamma \cup \{\rho\}) \leq_p \mathrm{VCSP_s}(\Gamma)$ *for some relation ρ that is invariant under c_0 but not under* sub.

Proof. By Lemmas 5.29 and 5.30, we have that $\mathrm{VCSP_s}(\Gamma^* \cup \{\rho\}) \leq_p \mathrm{VCSP_s}(\Gamma^*)$ for some relation ρ that is invariant under c_0 but not under sub. By Lemma 5.6, it holds $\mathrm{VCSP_s}(\Gamma^*) \leq_p \mathrm{VCSP_s}(\Gamma)$. ☐

Let us define weighted relations $\gamma_0 = \mathrm{Soft}(\rho_0)$, $\gamma_1 = \mathrm{Soft}(\rho_1)$, and $\gamma_= = \mathrm{Soft}(\rho_=)$; a binary relation $\rho_\leq = \{(0,0), (0,1), (1,1)\}$, and, for $r \in \{3, 4\}$, an

r-ary relation

$$A_r = \left\{ (x_1, \ldots, x_r) \in \{0,1\}^r \;\middle|\; \sum_{i=1}^{r} x_i \equiv 0 \,(\mathrm{mod}\ 2) \right\} . \tag{5.18}$$

Assuming that Γ does not admit polymorphism sub, we prove that $\mathrm{VCSP_s}(\Gamma)$ is NP-hard (see Lemma 5.38). The proof makes use of several sources of hardness. More specifically, we show that at least one of the following cases applies:

- $\mathrm{VCSP_s}(\mathrm{Feas}(\Gamma) \cup \mathrm{Opt}(\Gamma))$ is NP-hard (by the classification of $\{0,\infty\}$-valued languages, see Theorem 5.7).
- $\mathrm{VCSP}(\Gamma \cup \{\rho_0, \rho_1\})$ reduces to $\mathrm{VCSP_s}(\Gamma)$. In particular, it holds $\rho_\leq \in \Gamma^*$, which can be used to simulate the constants (see Lemma 5.36). The intractability of $\mathrm{VCSP}(\Gamma \cup \{\rho_0, \rho_1\})$ follows from Theorem 2.30.
- The NP-hard Minimum Distance problem [104] reduces to $\mathrm{VCSP_s}(\Gamma)$. In particular, it holds $\{A_3, \gamma_0\} \subseteq \Gamma^*$ or $\{A_4, \gamma_=\} \subseteq \Gamma^*$; the reduction from the Minimum Distance problem to these languages is given in Lemma 5.37.

Before proving Lemma 5.38, we need a few auxiliary lemmas to establish the existence of certain weighted relations in Γ^*.

Lemma 5.32. *Let ρ be a relation invariant under c_0 but not under $\neg$. Then $\rho_0 \in \{\rho\}^*$ or $\rho_\leq \in \{\rho\}^*$.*

Proof. Let r denote the arity of ρ. There exists an r-tuple $\mathbf{u} \in \rho$ such that $\neg(\mathbf{u}) \notin \rho$. If $\mathbf{1}^r \notin \rho$, we obtain ρ_0 by identifying all coordinates of ρ. Otherwise, we obtain $\rho_\leq$ by identifying all coordinates where $u_i = 0$ and identifying all coordinates where $u_i = 1$. $\square$

Lemma 5.33. *Let γ be a non-crisp weighted relation such that it admits multimorphism $\langle c_0 \rangle$. Then $\gamma_0 \in \{\gamma, \rho_0\}^*$. If in addition* $\mathrm{Feas}(\gamma)$ *and* $\mathrm{Opt}(\gamma)$ *are invariant under $\neg$, then $\gamma_= \in \{\gamma\}^*$.*

Proof. Let r denote the arity of γ. There exists an r-tuple $\mathbf{u}$ such that $\gamma(\mathbf{0}^r) < \gamma(\mathbf{u}) < \infty$. By 0-pinning at all coordinates where $u_i = 0$ and identifying all coordinates where $u_i = 1$, we obtain a unary weighted relation $\gamma' \in \{\gamma, \rho_0\}^*$ such that $\gamma'(0) < \gamma'(1) < \infty$. From it, we can obtain γ_0 by adding a constant and scaling, as $\gamma_0 = \frac{\gamma' - \gamma'(0)}{\gamma'(1) - \gamma'(0)}$.

If $\mathrm{Feas}(\gamma)$ and $\mathrm{Opt}(\gamma)$ are invariant under $\neg$, it holds $\gamma(\mathbf{1}^r) = \gamma(\mathbf{0}^r)$ and $\gamma(\mathbf{0}^r) < \gamma(\neg(\mathbf{u})) < \infty$. By identifying all coordinates where $u_i = 0$ and identifying all coordinates where $u_i = 1$, we obtain a binary weighted relation $\gamma' \in \{\gamma\}^*$. Consider $\gamma'' \in \{\gamma\}^*$ defined as $\gamma''(x, y) = \gamma'(x, y) + \gamma'(y, x)$. It holds $\gamma''(0, 0) = \gamma''(1, 1) < \gamma''(0, 1) = \gamma''(1, 0) < \infty$. From it, we can obtain $\gamma_=$ by adding a constant and scaling. □

Lemma 5.34. *Let ρ be a relation invariant under c_0, $\neg$, and* Mn*, but not under* sub*. Then $A_4 \in \{\rho\}^*$.*

Proof. Let ρ' be a smallest-arity relation in $\{\rho\}^*$ that is not invariant under sub, and denote its arity by r. As $\mathbf{0}^r \in \rho'$ and $\mathrm{Mn}(\mathbf{x}, \mathbf{y}, \mathbf{0}^r) = \mathbf{x} \oplus \mathbf{y}$, relation ρ' is closed under the $\oplus$ operation. Let $\mathbf{u}, \mathbf{v} \in \rho'$ be r-tuples such that $\mathrm{sub}(\mathbf{u}, \mathbf{v}) \notin \rho'$. There are no distinct coordinates i, j where $u_i = u_j$ and $v_i = v_j$, otherwise we could identify them to obtain an $(r-1)$-ary relation not invariant under sub. For any $a, b \in \{0, 1\}$, there is a coordinate i where $u_i = a$ and $v_i = b$, otherwise $\mathrm{sub}(\mathbf{u}, \mathbf{v})$ would be equal to $\neg(\mathbf{v})$, $\mathbf{u} \oplus \mathbf{v}$, $\mathbf{0}^r$, or $\mathbf{u}$ respectively, which would imply $\mathrm{sub}(\mathbf{u}, \mathbf{v}) \in \rho'$. Therefore, $r = 4$, and we may assume without loss of generality that $\mathbf{u} = (0, 0, 1, 1)$, $\mathbf{v} = (0, 1, 0, 1)$. As

$$\begin{aligned} \mathrm{sub}(\mathbf{u}, \mathbf{v}) &= (0, 0, 1, 0) \\ &= (0, 0, 0, 1) \oplus \mathbf{u} \\ &= (0, 1, 0, 0) \oplus (\mathbf{u} \oplus \mathbf{v}) \\ &= (1, 0, 0, 0) \oplus \neg(\mathbf{v}) \,, \end{aligned}$$

it holds $(0, 0, 0, 1), (0, 0, 1, 0), (0, 1, 0, 0), (1, 0, 0, 0) \notin \rho'$. Because ρ' is closed under $\neg$, we have $\rho' = A_4$. □

Lemma 5.35. *Let ρ be a relation invariant under c_0 but not under* sub*. If ρ is invariant under* Mn*, then $A_3 \in \{\rho, \rho_0\}^*$. If ρ is invariant under* min *or* max*, then $\rho_\leq \in \{\rho, \rho_0\}^*$.*

Proof. Let ρ' be a smallest-arity relation in $\{\rho, \rho_0\}^*$ that is not invariant under sub, and denote its arity by r. Let $\mathbf{u}, \mathbf{v} \in \rho'$ be r-tuples such that $\mathrm{sub}(\mathbf{u}, \mathbf{v}) \notin \rho'$. There are no distinct coordinates i, j where $u_i = u_j$ and $v_i = v_j$, otherwise we could identify them to obtain an $(r-1)$-ary relation not invariant under sub. For any $b \in \{0, 1\}$, there is a coordinate i where $u_i = 1$ and $v_i = b$, otherwise $\mathrm{sub}(\mathbf{u}, \mathbf{v})$ would be equal to $\mathbf{0}^r$ or $\mathbf{u}$ respectively, which would imply $\mathrm{sub}(\mathbf{u}, \mathbf{v}) \in \rho'$. However, there is no coordinate i where $u_i = v_i = 0$, otherwise we could obtain an $(r-1)$-ary relation not invariant under sub by 0-pinning ρ' at coordinate i. Therefore, $r = 2$ or $r = 3$. If $r = 2$, we have $\rho_\leq \in \{\rho, \rho_0\}^*$, and ρ is not invariant under Mn (as neither is $\rho_\leq$).

If $r = 3$, we may assume without loss of generality that $\mathbf{u} = (0, 1, 1)$ and $\mathbf{v} = (1, 0, 1)$. Relation ρ is not invariant under min, otherwise it would hold

$\min(\mathbf{u}, \mathbf{v}) = (0, 0, 1) \in \rho'$ and we could obtain a binary relation not invariant under sub by 0-pinning ρ' at the first coordinate. Similarly, relation ρ is not invariant under max, otherwise it would hold $\max(\mathbf{u}, \mathbf{v}) = (1, 1, 1) \in \rho'$ and we could obtain a binary relation not invariant under sub by identifying the first and third coordinate. Finally, assume that relation ρ is invariant under Mn. Then ρ' is also closed under the $\oplus$ operation, as $\mathrm{Mn}(\mathbf{x}, \mathbf{y}, \mathbf{0}^r) = \mathbf{x} \oplus \mathbf{y}$, and we have $\mathbf{u} \oplus \mathbf{v} = (1, 1, 0) \in \rho'$. Because $\mathrm{sub}(\mathbf{u}, \mathbf{v}) = (0, 1, 0) = (0, 0, 1) \oplus \mathbf{u} = (1, 1, 1) \oplus \mathbf{v} = (1, 0, 0) \oplus (\mathbf{u} \oplus \mathbf{v})$, it holds $(0, 0, 1), (1, 1, 1), (1, 0, 0) \notin \rho'$, and therefore $\rho' = A_3$. □

Lemma 5.36. *If $\rho_{\leq} \in \Gamma$, then* $\mathrm{VCSP}(\Gamma \cup \{\rho_0, \rho_1\}) \leq_p \mathrm{VCSP}_{\mathrm{s}}(\Gamma)$.

Proof. For a given instance of $\mathrm{VCSP}(\Gamma \cup \{\rho_0, \rho_1\})$ with variables V, we construct an instance of $\mathrm{VCSP}_{\mathrm{s}}(\Gamma)$ as follows: We introduce new variables y_0, y_1 and impose constraints $\rho_{\leq}(y_0, x)$, $\rho_{\leq}(x, y_1)$ for all $x \in V$ to ensure that $y_0 = 0$, $y_1 = 1$ in any feasible surjective assignment. Then we replace each constraint of the form $\rho_0(x)$ with $\rho_{\leq}(x, y_0)$ and each constraint of the form $\rho_1(x)$ with $\rho_{\leq}(y_1, x)$. □

Lemma 5.37. *Languages $\{A_3, \gamma_0\}$ and $\{A_4, \gamma_=\}$ are both s-intractable.*

Proof. First we show a reduction from the optimisation variant of the Minimum Distance problem, which is NP-hard [104], to $\mathrm{VCSP}_{\mathrm{s}}(\{A_3, \gamma_0\})$. A problem instance is given as an $m \times n$ matrix H over the field $D = \{0, 1\}$, and the objective is to find a non-zero vector $\mathbf{x} = (x_1, \ldots, x_n) \in D^n$ satisfying $H \cdot \mathbf{x} = \mathbf{0}^m$ with the minimum weight (i.e. $\sum_{i=1}^n x_i$).

Note that $\rho_0 = \mathrm{Opt}(\gamma_0)$, and therefore we may use relation ρ_0 as well (by Lemma 5.6). We construct a $\mathrm{VCSP}_{\mathrm{s}}$ instance I as follows: Let $x_1, \ldots, x_n$ be variables corresponding to the elements of the sought vector $\mathbf{x}$. The requirement $H \cdot \mathbf{x} = \mathbf{0}^m$ can be seen as a system of m linear equations, each in the form $\bigoplus_{i=1}^k x_{a_i} = 0$ for a set $\{a_1, \ldots, a_k\} \subseteq [n]$ (the set may differ for each equation). We encode such an equation by introducing new variables $y_0, \ldots, y_k$ and imposing constraints $\rho_0(y_0)$, $A_3(y_{i-1}, x_{a_i}, y_i)$ for all $i \in [k]$, and $\rho_0(y_k)$. These ensure that each variable y_j is assigned the value of the prefix sum $\bigoplus_{i=1}^j x_{a_i}$, and that the total sum equals 0. Finally, we encode the objective function of the Minimum Distance problem by imposing constraints $\gamma_0(x_1), \ldots, \gamma_0(x_n)$.

Every vector $\mathbf{x} \in D^n$ satisfying $H \cdot \mathbf{x} = \mathbf{0}^m$ corresponds to a feasible assignment to I. If $\mathbf{x}$ is non-zero, the corresponding assignment is surjective, as at least one of variables $x_1, \ldots, x_n$ gets label 1 and, for every equation, variable y_0 gets label 0. Conversely, if a feasible assignment to I is surjective, then it corresponds to a non-zero vector $\mathbf{x}$ (labelling all variables $x_1, \ldots, x_n$

with 0 implies that all the prefix sums y_j equal 0 as well). The objective value of the assignment corresponding to a vector $\mathbf{x}$ equals the weight of $\mathbf{x}$, and hence finding an optimal surjective solution to I solves the Minimum Distance problem.

Finally, we show that $\{A_4, \gamma_=\}$ is s-intractable by a reduction from $\mathrm{VCSP}_\mathrm{s}(\{A_3, \gamma_0\})$ to $\mathrm{VCSP}_\mathrm{s}(\{A_4, \gamma_=\})$. Given an instance I, we construct an instance I' by introducing a new variable w and replacing each constraint of the form $A_3(x, y, z)$ with $A_4(x, y, z, w)$ and each constraint of the form $\gamma_0(x)$ with $\gamma_=(x, w)$. Any surjective assignment to I can be extended to a surjective assignment to I' of the same objective value by labelling w with 0. Conversely, consider a feasible surjective assignment s' to I'; we may assume $s'(w) = 0$ since language $\{A_4, \gamma_=\}$ admits multimorphism $\langle\neg\rangle$. Restricting s' to the variables of I gives us a surjective assignment to I of the same objective value. Note that if s' assigns label 1 to all the variables except w, its restriction will not be surjective; however, such s' violates constraints $\rho_0(y_0)$ and thus is not feasible. □

Lemma 5.38. *Let* Γ *be a language such that it admits multimorphism* $\langle c_0\rangle$ *but not polymorphism* sub. *If* $\Gamma \cup \{\rho_0, \rho_1\}$ *is intractable, then* $\mathrm{VCSP}_\mathrm{s}(\Gamma)$ *is NP-hard.*

Proof. Let $\Phi = \mathrm{Feas}(\Gamma) \cup \mathrm{Opt}(\Gamma) \subseteq \Gamma^*$. Suppose that Φ does not admit any of the following polymorphisms: min, max, Mn, and Mj. By the classification of $\{0, \infty\}$-valued languages (see Theorem 5.7), Φ is s-intractable. Hence, $\mathrm{VCSP}_\mathrm{s}(\Gamma)$ is NP-hard by Lemma 5.6. In the rest of the proof, we assume that Φ admits at least one of polymorphisms min, max, Mn, and Mj. Note that Φ admits polymorphism c_0 but not polymorphism sub. Since $\min(x, y) = \mathrm{Mj}(x, y, 0)$, we may assume that Φ admits at least one of polymorphisms min, max, and Mn.

Suppose that Φ admits polymorphism $\neg$. Then it does not admit min, as $\mathrm{sub}(x, y) = \min(x, \neg y)$, nor it admits max, as $\min(x, y) = \neg\max(\neg x, \neg y)$. Therefore, Φ admits polymorphism Mn. If Γ is crisp, then language $\Gamma \cup \{\rho_0, \rho_1\}$ admits multimorphism $\langle\mathrm{Mn}, \mathrm{Mn}, \mathrm{Mn}\rangle$ and thus is tractable by Theorem 2.30, which contradicts an assumption of the lemma. Hence, Γ is not crisp. By Lemmas 5.33 and 5.34, we have $\{A_4, \gamma_=\} \subseteq \Gamma^*$. Therefore, $\mathrm{VCSP}_\mathrm{s}(\Gamma)$ is NP-hard by Lemma 5.37.

If Φ does not admit polymorphism $\neg$, then, by Lemma 5.32, we have $\rho_0 \in \Gamma^*$ or $\rho_\leq \in \Gamma^*$. If $\rho_\leq \in \Gamma^*$, $\mathrm{VCSP}_\mathrm{s}(\Gamma)$ is NP-hard by Lemma 5.36 and we are done; in the rest of the proof we assume that $\rho_\leq \notin \Gamma^*$ and hence $\rho_0 \in \Gamma^*$. If Φ admits polymorphism min or max, we get $\rho_\leq \in \Gamma^*$ by Lemma 5.35, which is a contradiction. Therefore, Φ admits Mn, and thus Γ is not crisp (by the same argument as in the previous paragraph). By Lemmas 5.33 and 5.35, we have $\{A_3, \gamma_0\} \subseteq \Gamma^*$. Therefore, $\mathrm{VCSP_s}(\Gamma)$ is NP-hard by Lemma 5.37. $\square$

Theorem 5.39. *Let Γ be a language such that it is not EDS, $\neg(\Gamma)$ is not EDS, and Γ does not admit any of the following multimorphisms: $\langle \min, \min \rangle$, $\langle \max, \max \rangle$, $\langle \min, \max \rangle$, $\langle \mathrm{Mn}, \mathrm{Mn}, \mathrm{Mn} \rangle$, $\langle \mathrm{Mj}, \mathrm{Mj}, \mathrm{Mj} \rangle$, $\langle \mathrm{Mj}, \mathrm{Mj}, \mathrm{Mn} \rangle$. Then $\mathrm{VCSP_s}(\Gamma)$ is NP-hard.*

Proof. If Γ does not admit at least one of multimorphisms $\langle c_0 \rangle$ and $\langle c_1 \rangle$, it is intractable by Theorem 2.30, and hence $\mathrm{VCSP_s}(\Gamma)$ is NP-hard by Lemma 5.2. Language $\Gamma \cup \{\rho_0, \rho_1\}$ is, by the same theorem, intractable. We may assume that Γ admits multimorphism $\langle c_0 \rangle$; if it does not, we consider $\neg(\Gamma)$ instead. By Corollary 5.31 and Lemma 5.38, $\mathrm{VCSP_s}(\Gamma)$ is NP-hard. $\square$

5.4 Tractability of EDS languages

We prove that EDS languages are globally s-tractable by a reduction to a generalised variant of the Min-Cut problem. The problem is defined in Section 5.4.1, its tractability is established in Section 5.4.2, and the reduction is stated in Section 5.4.3.

5.4.1 Generalised Min-Cut problem

Let V be a finite set. A *set function* on V is a function $\gamma : 2^V \to \mathbb{Q}_{\geq 0} \cup \{\infty\}$ with $\gamma(\emptyset) = 0$.

Definition 5.40. A set function $\gamma : 2^V \to \mathbb{Q}_{\geq 0} \cup \{\infty\}$ is *symmetric* if $\gamma(X) = \gamma(V \setminus X)$ for all $X \subseteq V$; it is *increasing* if $\gamma(X) \leq \gamma(Y)$ for all $X \subseteq Y \subseteq V$; it is *superadditive* if

$$\gamma(X) + \gamma(Y) \leq \gamma(X \cup Y) \tag{5.19}$$

for all disjoint $X, Y \subseteq V$; it is *posimodular* if

$$\gamma(X) + \gamma(Y) \geq \gamma(X \setminus Y) + \gamma(Y \setminus X) \tag{5.20}$$

for all $X, Y \subseteq V$; and it is *submodular* if

$$\gamma(X) + \gamma(Y) \geq \gamma(X \cap Y) + \gamma(X \cup Y) \tag{5.21}$$

for all $X, Y \subseteq V$.

Note that any superadditive set function is also increasing, as for all $X \subseteq Y \subseteq V$ it holds $\gamma(X) \leq \gamma(X) + \gamma(Y \setminus X) \leq \gamma(Y)$ by superadditivity. In the case of symmetric set functions, submodularity implies posimodularity, as

$$\begin{aligned}
\gamma(X) + \gamma(Y) &= \gamma(X) + \gamma(V \setminus Y) && (5.22)\\
&\geq \gamma(X \cap (V \setminus Y)) + \gamma(X \cup (V \setminus Y)) && (5.23)\\
&= \gamma(X \setminus Y) + \gamma(V \setminus (Y \setminus X)) && (5.24)\\
&= \gamma(X \setminus Y) + \gamma(Y \setminus X)\,. && (5.25)
\end{aligned}$$

and, similarly, posimodularity implies submodularity.

Example 5.41. Let V be a finite set and $T \subseteq V$ a non-empty subset. We define a set function γ on V by $\gamma(X) = 1$ if $T \subseteq X$ and $\gamma(X) = 0$ otherwise. Intuitively, this corresponds to a soft NAND constraint imposed on variables T. The set function γ is superadditive, and hence also increasing.

We now formally define the Min-Cut problem.

Definition 5.42. An instance of the *Min-Cut* (MC) problem is given by an undirected graph $G = (V, E)$ with edge weights $w : E \to \mathbb{Q}_{\geq 0} \cup \{\infty\}$. The objective function g of the MC problem is a set function on V defined by

$$g(X) = \sum_{|X \cap \{u,v\}| = 1} w(u, v)\,. \tag{5.26}$$

Function g is a well-known example of a submodular function. Since it is symmetric, it is also posimodular.

A *solution* to the MC problem is a set X such that $\emptyset \subsetneq X \subsetneq V$. Note that a cut $(X, V \setminus X)$ corresponds to two solutions, namely X and $V \setminus X$. An *optimal* solution is a solution with the minimum objective value among all solutions. A *minimal* optimal solution is an optimal solution with no proper subset being an optimal solution.

Note that any two different minimal optimal solutions X, Y must be disjoint, otherwise $X \setminus Y$ or $Y \setminus X$ would be a smaller optimal solution (by the posimodularity of g).

Although the definition allows infinite weight edges, those can be easily eliminated by identifying their endpoints, and so we may assume that all edge weights are finite. Edges with weight 0 are conventionally disregarded.

Finally, we define the Generalised Min-Cut problem, which further generalises the problem introduced in [102].

Definition 5.43. An instance J of the *Generalised Min-Cut* (GMC) problem is given by an undirected graph $G = (V, E)$ with edge weights $w : E \to \mathbb{Q}_{\geq 0} \cup \{\infty\}$, and an oracle defining a superadditive set function f on V. The objective function the GMC problem is a set function on V defined by $J(X) = f(X) + g(X)$, where g is the objective function of the underlying Min-Cut problem on G.

A *solution* to the GMC problem is a set X such that $\emptyset \subsetneq X \subsetneq V$. An *optimal* solution is a solution with the minimum objective value among all solutions. We denote this minimum objective value by λ. For any $\alpha \geq 1$, an *α-optimal* solution is a solution X such that $J(X) \leq \alpha\lambda$.

We show in Theorem 5.50 that, in the case of $0 < \lambda < \infty$ and a fixed $\alpha \geq 1$, there are only polynomially many α-optimal solutions and they can be found in polynomial time.

5.4.2 Tractability of the Generalised Min-Cut problem

In this section, we present a polynomial-time algorithm that solves the Generalised Min-Cut problem. We assume that $w(u, v) \in \mathbb{Q}_{>0}$ for all edges (u, v).

Lemma 5.44. *There is a polynomial-time algorithm that, given an instance J of the GMC problem, either finds a solution X with $J(X) = \lambda = 0$, or determines that $\lambda = \infty$, or determines that $0 < \lambda < \infty$.*

Proof. A solution X with $J(X) = f(X)+g(X) = 0$ satisfies $f(X) = g(X) = 0$, and hence it does not cut any edge. Because the set function f is increasing, we may assume that X is a single connected component. The algorithm simply tries each connected component as a solution, which takes a linear number of queries to the oracle for f.

The case of $\lambda = \infty$ occurs only if $f(X) = \infty$ for all solutions X. Because f is increasing, it is sufficient to check all solutions of size 1. $\square$

In view of Lemma 5.44, we can assume that $0 < \lambda < \infty$. Our goal is to show that, for a given $\alpha \geq 1$, all α-optimal solutions to a GMC instance can be found in polynomial time. This is proved in Theorem 5.50; before that we need to prove several auxiliary lemmas on properties of the MC and GMC problems.

Lemma 5.45. *For any instance J of the GMC problem on a graph $G = (V, E)$ and any non-empty set $V' \subseteq V$, there is an instance J' on the induced subgraph $G[V']$ that preserves the objective value of all solutions $X \subsetneq V'$. In particular, any α-optimal solution X of J such that $X \subsetneq V'$ is α-optimal for J' as well.*

Proof. Edges with exactly one endpoint in V' need to be taken into account separately because they do not appear in the induced subgraph. We accomplish that by defining the new set function f' by

$$f'(X) = f(X) + \sum_{u \in X} \sum_{v \in V \setminus V'} w(u, v) \tag{5.27}$$

for all $X \subseteq V'$. By the construction, f' is superadditive, and the objective value $J'(X)$ for any $X \subsetneq V'$ equals $J(X)$.

Note that the minimum objective value for J' is greater than or equal to the minimum objective value for J. Therefore, any solution $X \subsetneq V'$ that is α-optimal for J is also α-optimal for J'. $\square$

Lemma 5.46. *Let X be an optimal solution to an instance of the GMC problem over vertices V with $\lambda < \infty$, and Y a minimal optimal solution to the underlying MC problem. Then $X \subseteq Y$, $X \subseteq V \setminus Y$, or X is an optimal solution to the underlying MC problem.*

Proof. Assume that $X \nsubseteq Y$ and $X \nsubseteq V \setminus Y$. If $Y \subseteq X$, we have $f(Y) \leq f(X)$ as f is increasing, and hence $f(Y) + g(Y) \leq f(X) + g(X) < \infty$. Therefore, Y is optimal for the GMC problem and X is optimal for the MC problem. In the rest, we assume that $Y \nsubseteq X$.

By the posimodularity of g we have $g(X) + g(Y) \geq g(X \setminus Y) + g(Y \setminus X)$. Because $Y \setminus X$ is a proper non-empty subset of Y, it holds $g(Y \setminus X) > g(Y)$, and hence $g(X) > g(X \setminus Y)$. But then $f(X) + g(X) > f(X \setminus Y) + g(X \setminus Y)$ as $\infty > f(X) \geq f(X \setminus Y)$. Set $X \setminus Y$ is non-empty, and therefore contradicts the optimality of X. $\square$

The following lemma relates the number of optimal solutions and the number of minimal optimal solutions to the MC problem. Note that this bound is tight for (unweighted) paths and cycles with at most one path attached to each vertex.

Lemma 5.47. *For any instance of the MC problem on a connected graph with $n \geq 2$ vertices and p minimal optimal solutions, there are at most $p(p-1) + 2(n-p)$ optimal solutions.*

We prove the lemma by induction on n, closely following the proof that establishes the cactus representation of minimum cuts in [47]. We note that the cactus representation could be applied directly to obtain a weaker bound of $p(p-1) + O(n)$ but we do not know how to achieve the exact bound using it.

Proof. For $n = 2$, the lemma holds as there are exactly two solutions and both are minimal optimal. Assume $n \geq 3$. We denote the number of optimal solutions by s. A solution X is called a *star* if $|X| = 1$ or $|X| = n - 1$, otherwise it is called *proper*.

First we consider the case where every optimal solution is a star. Let us denote the minimum cuts by $(\{v_1\}, V \setminus \{v_1\}), \dots, (\{v_h\}, V \setminus \{v_h\})$. If $h = 1$, then we have $s = p = 2$ and the bound holds. Otherwise, there are $2h$ optimal solutions but only h of them are minimal (i.e., $\{v_1\}, \dots, \{v_h\}$). Hence,

$$\begin{aligned} p(p-1) + 2(n-p) &= 2h + (h-1) \cdot (h-2) - 2 + 2(n-h) && (5.28) \\ &\geq 2h = s && (5.29) \end{aligned}$$

as it holds $n \geq h \geq 2$ and $n \geq 3$.

From now on we assume that there is a proper optimal solution, and hence $n \geq 4$. We say that solutions X, Y *cross* if none of $X \setminus Y$, $Y \setminus X$, $X \cap Y$, $V \setminus (X \cup Y)$ is empty. Note that only proper solutions might cross. If every proper optimal solution is crossed by some optimal solution, then the graph is a cycle with edges of equal weight [47, Lemma 7.1.3]. In that case, there are $n(n-1)$ optimal solutions (all sets of contiguous vertices except for $\emptyset$ and V) and n minimal optimal solutions (all singletons), and therefore the bound holds.

Finally, assume that there is a proper optimal solution that is not crossed by any optimal solution, and denote the corresponding minimum cut by (V_1, V_2). For any optimal solution X, it must hold either $X \subseteq V_1$, $V_1 \subseteq X$, $X \subseteq V_2$, or $V_2 \subseteq X$. For $i \in \{1, 2\}$, let G_i be the result of shrinking V_i into a new vertex t_i so that the weight of any edge (t_i, v) for $v \in V \setminus V_i$ equals the sum of weights of edges (u, v) for $u \in V_i$. Denote by n_i, p_i, and s_i the number of vertices, minimal optimal solutions, and optimal solutions to G_i. It holds $n = n_1 + n_2 - 2$. Consider any solution X' of G_i: If $t_i \notin X'$, it corresponds to a solution $X = X'$ of the original graph G; otherwise it corresponds to $X = X' \setminus \{t_i\} \cup V_i$. In both cases, the objective values of X' and X in their respective problem instances are equal. Therefore, any optimal solution X of G such that $X \subseteq V_2$ or $V_1 \subseteq X$ corresponds to an optimal solution to G_1, and any optimal solution to G such that $X \subseteq V_1$ or $V_2 \subseteq X$ corresponds to an optimal solution in G_2. Hence, $p = p_1 + p_2 - 2$ and $s = s_1 + s_2 - 2$, as only solutions V_1 and V_2 satisfy both conditions simultaneously. By the inductive hypothesis, we get

$$\begin{aligned} p(p-1)+2(n-p) &= p_1(p_1-1)+2(n_1-p_1)+p_2(p_2-1)+2(n_2-p_2) \\ &\quad + 2(p_1-2)\cdot(p_2-2)-2 \qquad (5.30) \\ &\geq s_1+s_2-2+2(p_1-2)\cdot(p_2-2) \qquad (5.31) \\ &\geq s \qquad (5.32) \end{aligned}$$

as it holds $p_1, p_2 \geq 2$. □

Lemma 5.48. *For any instance of the GMC problem on n vertices with $0 < \lambda < \infty$, the number of optimal solutions is at most $n(n-1)$. There is an algorithm that finds all of them in polynomial time.*

Note that the bound of $n(n-1)$ optimal solutions precisely matches the known upper bound of $\binom{n}{2}$ for the number of minimum cuts [68]; the bound is tight for cycles.

Proof. Let $t(n)$ denote the maximum number of optimal solutions for such instances on n vertices. We prove the bound by induction on n. If $n = 1$, there are no solutions and hence $t(1) = 0$. For $n \geq 2$, let $Y_1, \ldots, Y_p$ be the minimal optimal solutions to the underlying MC problem. As there exists at least one minimum cut and the minimal optimal solutions are all disjoint, it holds $2 \leq p \leq n$.

First, suppose that $\bigcup Y_i = V$. By Lemma 5.46, any optimal solution to the GMC problem is either a proper subset of some Y_i or an optimal solution to the underlying MC problem. Restricting solutions to a proper subset of Y_i is, by Lemma 5.45, equivalent to considering a GMC problem instance on vertices Y_i, and hence the number of such optimal solutions is bounded by $t(|Y_i|) \leq |Y_i| \cdot (|Y_i| - 1)$. Since it holds $\sum |Y_i| = n$ and $|Y_i| \geq 1$ for all i, the sum $\sum |Y_i| \cdot (|Y_i| - 1)$ is maximised when $p-1$ of the sets Y_i are singletons and the size of the remaining one equals $n-p+1$. If the graph is connected, then, by Lemma 5.47, there are at most $p(p-1)+2(n-p)$ optimal solutions to the underlying MC problem. Adding these upper bounds we get

$$\begin{aligned} & p(p-1)+2(n-p)+\sum_{i=1}^{p} |Y_i|\cdot(|Y_i|-1) \qquad (5.33) \\ & \leq p(p-1)+2(n-p)+(p-1)\cdot 1\cdot 0+(n-p+1)\cdot(n-p) \qquad (5.34) \\ & = n(n-1)-2(p-2)\cdot(n-p) \qquad (5.35) \\ & \leq n(n-1)\,. \qquad (5.36) \end{aligned}$$

If the graph is disconnected, the sets $Y_1, \ldots, Y_p$ are precisely its connected components. The optimal solutions to the underlying MC problem are precisely unions of connected components (with the exception of $\emptyset$ and V), which means that there can be exponentially many of them. However, only the sets $Y_1, \ldots, Y_p$ themselves can be optimal solutions to the GMC problem: We have $0 < \lambda \leq f(Y_i) + g(Y_i) = f(Y_i)$. Because f is superadditive, it holds

$$f(Y_{i_1} \cup \cdots \cup Y_{i_k}) \geq f(Y_{i_1}) + \cdots + f(Y_{i_k}) \geq k\lambda \tag{5.37}$$

for any distinct $i_1, \ldots, i_k$, and hence no union of two or more connected components can be an optimal solution to the GMC problem. This gives us an upper bound of $p \leq p(p-1) + 2(n-p)$, and the rest follows as in the previous case.

Finally, suppose that $\bigcup Y_i \neq V$, and hence the graph is connected. Let $Z = V \setminus \bigcup Y_i$. By Lemma 5.46, any optimal solution to the GMC problem is a proper subset of some Y_i, a proper subset of Z, set Z itself, or an optimal solution to the underlying MC problem. Similarly as before, we get an upper bound of

$$\begin{aligned}
&p(p-1) + 2(n-p) + \sum_{i=1}^{p} |Y_i| \cdot (|Y_i| - 1) + |Z| \cdot (|Z| - 1) + 1 && (5.38)\\
&\leq p(p-1) + 2(n-p) + p \cdot 1 \cdot 0 + (n-p) \cdot (n-p-1) + 1 && (5.39)\\
&= n(n-1) - 2(p-1) \cdot (n-p) + 1 && (5.40)\\
&\leq n(n-1)\,. && (5.41)
\end{aligned}$$

Using a procedure generating all minimum cuts [105], it is straightforward to turn the above proof into a recursive algorithm that finds all optimal solutions in polynomial time. ☐

Lemma 5.49. *Let $\alpha, \beta \geq 1$. Let X be an α-optimal solution to an instance J of the GMC problem over vertices V with $0 < \lambda < \infty$, and Y an optimal solution to the underlying MC problem. If $g(Y) < \lambda/\beta$, then*

$$J(X \setminus Y) + J(X \cap Y) < \left(\alpha + \frac{2}{\beta}\right)\lambda\,; \tag{5.42}$$

if $g(Y) \geq \lambda/\beta$, then X is an $\alpha\beta$-optimal solution to the underlying MC problem.

Proof. If $g(Y) \geq \lambda/\beta$, it holds $g(X) \leq J(X) \leq \alpha\lambda \leq \alpha\beta \cdot g(Y)$, and hence X is an $\alpha\beta$-optimal solution to the underlying MC problem. In the rest we assume that $g(Y) < \lambda/\beta$.

Because g is posimodular, we have

$$\begin{aligned}
g(X) + g(Y) &\geq g(X \setminus Y) + g(Y \setminus X) && (5.43)\\
g(Y) + g(Y \setminus X) &\geq g(X \cap Y) + g(\emptyset)\,, && (5.44)
\end{aligned}$$

and hence

$$g(X) + 2g(Y) \geq g(X \setminus Y) + g(X \cap Y)\,. \tag{5.45}$$

By superadditivity of f, it holds $f(X) \geq f(X \setminus Y) + f(X \cap Y)$. The claim then follows from the fact that $f(X) + g(X) + 2g(Y) < (\alpha + 2/\beta)\lambda$. ☐

Finally, we prove that α-optimal solutions to the GMC problem can be found in polynomial time.

Theorem 5.50. *For any instance J of the GMC problem on n vertices with $0 < \lambda < \infty$ and $\alpha \in \mathbb{Z}_{\geq 1}$, the number of α-optimal solutions is at most $n^{20\alpha-15}$. There is an algorithm that finds all of them in polynomial time.*

Note that for a cycle on n vertices, the number of α-optimal solutions to the MC problem is $\Theta(n^{2\alpha})$, and thus the exponent in our bound is asymptotically tight in α.

Proof. Let $\beta \in \mathbb{Z}_{\geq 3}$ be a parameter. Throughout the proof, we relax the integrality restriction on α and require only that $\alpha\beta$ is an integer. For $\alpha = 1$, the claim follows from Lemma 5.48, therefore we assume $\alpha \geq 1 + 1/\beta$ in the rest of the proof.

Define a linear function ℓ by

$$\ell(x) = \frac{2(\beta+1)}{\beta-2} \cdot (\beta x - 3)\ . \tag{5.46}$$

We prove that the number of α-optimal solutions is at most $n^{\ell(\alpha)}$; taking $\beta = 4$ then gives the claimed bound. Function ℓ was chosen as a slowest-growing function satisfying the following properties required in this proof: It holds $\ell(x) + \ell(y) \leq \ell(x + y - 3/\beta)$ for any x, y, and $\ell(x) \geq 2\beta x$ for any $x \geq 1 + 1/\beta$.

We prove the bound by induction on $n + \alpha\beta$. As it trivially holds for $n \leq 2$, we assume $n \geq 3$ in the rest of the proof. Let Y be an optimal solution to the underlying MC problem with $k = |Y| \leq n/2$. If $g(Y) \geq \lambda/\beta$ then, by Lemma 5.49, any α-optimal solution to the GMC problem is an $\alpha\beta$-optimal solution to the underlying MC problem. Because $g(Y) \geq \lambda/\beta > 0$, the graph is connected, and hence there are at most

$$2^{2\alpha\beta}\binom{n}{2\alpha\beta} \leq n^{2\alpha\beta} \leq n^{\ell(\alpha)} \tag{5.47}$$

such solutions by [68]. (In detail, [68, Theorem 6.2] shows that the number of $\alpha\beta$-optimal cuts in an n-vertex graph is $2^{2\alpha\beta-1}\binom{n}{2\alpha\beta}$, and every cut corresponds to two solutions.)

From now on we assume that $g(Y) < \lambda/\beta$, and hence inequality (5.42) holds. Upper bounds in this case may be quite loose; in particular, we use the following inequalities:

$$(k/n)^{\ell(\alpha)} \leq (k/n)^{\ell(1+1/\beta)} = (k/n)^{2(\beta+1)} \leq (k/n)^8 \leq (k/n)(1/2)^7 = k/128n \tag{5.48}$$

$$(1/n)^{2\beta} \leq (1/n)^6 \leq (1/n)(1/3)^5 < 1/128n\,. \tag{5.49}$$

Consider any α-optimal solution to the GMC problem X.

If $X \subsetneq Y$, then, by Lemma 5.45, X is an α-optimal solution to an instance on vertices Y. By the induction hypothesis, there are at most $k^{\ell(\alpha)} \leq (k/128n) \cdot n^{\ell(\alpha)}$ such solutions.

Similarly, if $X \subsetneq V \setminus Y$, then X is an α-optimal solution to an instance on vertices $V \setminus Y$, and there are at most

$$(n-k)^{\ell(\alpha)} = (1-k/n)^{\ell(\alpha)} \cdot n^{\ell(\alpha)} \leq (1-k/n) \cdot n^{\ell(\alpha)} \tag{5.50}$$

such solutions.

If $Y \subsetneq X$, then $X \setminus Y$ is an $(\alpha - 1 + 2/\beta)$-optimal solution on vertices $V \setminus Y$ by (5.42) and the fact that $J(X \cap Y) \geq \lambda$. Similarly, if $V \setminus Y \subsetneq X$, then $X \cap Y$ is an $(\alpha - 1 + 2/\beta)$-optimal solution on vertices Y. In either case, we bound the number of such solutions depending on the value of α: For $\alpha < 2 - 2/\beta$, there are trivially none; for $\alpha = 2 - 2/\beta$, Lemma 5.48 gives a bound of $n(n-1) \leq n^{\ell(\alpha)-2\beta}$; and for $\alpha > 2 - 2/\beta$ we get an upper bound of $n^{\ell(\alpha-1+2/\beta)} \leq n^{\ell(\alpha)-2\beta}$ by the induction hypothesis. The number of solutions is thus at most $n^{\ell(\alpha)-2\beta} \leq (1/128n) \cdot n^{\ell(\alpha)}$ for any α.

Finally, we consider X such that $\emptyset \subsetneq X \setminus Y \subsetneq V \setminus Y$ and $\emptyset \subsetneq X \cap Y \subsetneq Y$, i.e., $X \setminus Y$ and $X \cap Y$ are solutions on vertices $V \setminus Y$ and Y respectively. Let i be the integer for which

$$\left(1 + \frac{i}{\beta}\right)\lambda \leq J(X \cap Y) < \left(1 + \frac{i+1}{\beta}\right)\lambda\,. \tag{5.51}$$

Then, by (5.42), it holds $J(X \setminus Y) < (\alpha - 1 - (i-2)/\beta)\lambda$. Therefore, $X \cap Y$ is a $(1+(i+1)/\beta)$-optimal solution on vertices Y and $X \setminus Y$ is an $(\alpha - 1 - (i-2)/\beta)$-optimal solution on vertices $V \setminus Y$. Because $0 \leq i \leq (\alpha - 2)\beta + 1$, we can bound the number of such solutions by the induction hypothesis as at most

$$k^{\ell\left(1+\frac{i+1}{\beta}\right)} \cdot (n-k)^{\ell\left(\alpha-1-\frac{i-2}{\beta}\right)} \leq \left(\frac{k}{n}\right)^{\ell\left(1+\frac{i+1}{\beta}\right)} \cdot n^{\ell\left(1+\frac{i+1}{\beta}\right)+\ell\left(\alpha-1-\frac{i-2}{\beta}\right)} \tag{5.52}$$

$$\leq \left(\frac{k}{n}\right)^{2(\beta+1)} \cdot \frac{1}{2^i} \cdot n^{\ell(\alpha)}\,, \tag{5.53}$$

which is at most $2 \cdot (k/128n) \cdot n^{\ell(\alpha)}$ in total for all i.

By adding up the bounds we get that the number of α-optimal solutions is at most $n^{\ell(\alpha)}$. A polynomial-time algorithm that finds the α-optimal solutions follows from the above proof using a procedure generating all $\alpha\beta$-optimal cuts [105]. ☐

Remark 5.51. For our reduction from the VCSP$_\mathrm{s}$ over EDS languages, we need to find all α-optimal solutions to the GMC problem. However, if one is only interested in a single optimal solution, the presented algorithm can be easily adapted to an even more general problem.

Let f, g be set functions on V given by an oracle such that $f : 2^V \to \mathbb{Q}_{\geq 0} \cup \{\infty\}$ is increasing and $g : 2^V \to \mathbb{Q}_{\geq 0}$ satisfies the posimodularity and submodularity inequalities for intersecting pairs of sets (i.e. sets X, Y such that neither of $X \cap Y, X \setminus Y, Y \setminus X$ is empty). The objective is to minimise the sum of f and g.

The case when the optimum value $\lambda = \infty$ can be recognised by checking all solutions of size 1. Assuming $\lambda < \infty$, note that the proof of Lemma 5.46 works even for this more general problem. Let Y be a minimal optimal solution to g. It follows that there is an optimal solution X to $f + g$ such that $X \subseteq Y$, $X \subseteq V \setminus Y$, or X is itself a *minimal* optimal solution to g (as f is increasing). We can find all minimal optimal solutions to g in polynomial time [87, Theorem 10.11]. Restricting f, g to a subset of V preserves the required properties, and hence we can recursively solve the problem on Y and $V \setminus Y$. Therefore, an optimal solution to $f + g$ can be found in polynomial time.

5.4.3 Reduction to the Generalised Min-Cut problem

At the heart of our reduction is an observation that EDS weighted relations can be approximated by instances of the Generalised Min-Cut problem. We define this notion of approximability in Definition 5.55. In Theorem 5.56, we show how to approximate any EDS weighted relation with a constant factor. However, that construction does not yield a sufficient bound on the approximation factor; we present it only in order to provide some intuition for the more opaque construction in Theorem 5.58. Using that, we establish the global s-tractability of EDS languages in Theorem 5.59.

In this section, we equate weighted relations admitting multimorphism $\langle c_0 \rangle$ with set functions; the correspondence is formally stated in the following definition. Note that we may without loss of generality assume that the minimum assigned value equals 0, as adding a constant to a weighted relation preserves tractability.

Definition 5.52. Let γ be an r-ary weighted relation such that, for any r-tuple $\mathbf{x}$, $\gamma(\mathbf{x}) \geq \gamma(\mathbf{0}^r) = 0$. The *corresponding set function* γ' on $[r]$ is defined by $\gamma'(X) = \gamma(\mathbf{x})$ where $x_i = 1 \iff i \in X$.

The definition of α-EDS weighted relations then translates into the following:

Definition 5.53. For any $\alpha \geq 1$, a set function γ on V is α-*EDS* if, for every $X, Y \subseteq V$, it holds

$$\alpha \cdot (\gamma(X) + \gamma(Y)) \geq \gamma(X \setminus Y) \,. \tag{5.54}$$

Remark 5.54. Inequality (5.4) could be modified so that (5.54) becomes symmetric, say

$$\alpha \cdot (\gamma(X) + \gamma(Y)) \geq \gamma(X \setminus Y) + \gamma(Y \setminus X) \,. \tag{5.55}$$

It is easy to see that, although the set of α-EDS weighted relations for a fixed α would be different, this change would not affect the set of EDS weighted relations. We opt for the shorter, albeit asymmetric, definition.

Definition 5.55. Let J be an instance of the GMC problem on vertices V and γ a set function on V. For any $\alpha \geq 1$, we say that J α-*approximates* γ if, for all $X \subseteq V$,

$$J(X) \leq \gamma(X) \leq \alpha \cdot J(X) \,. \tag{5.56}$$

A set function is α-*approximable* if there exists a GMC instance that α-approximates it, and it is *approximable* if it is α-approximable for some $\alpha \geq 1$.

Theorem 5.56. *Any α-EDS set function is approximable.*

Proof. Let γ be an α-EDS set function on $[n]$ and γ' the corresponding n-ary weighted relation. By Corollary 5.21, both $\mathrm{Feas}(\gamma')$ and $\mathrm{Opt}(\gamma')$ are essentially downsets (recall Definition 5.18). The rest of the proof relies only on this property and does not depend on the value of α. The intuition behind our construction is that the downset in (5.8) can be represented by a superadditive function on $[n]$, while the binary equality relations can be represented by edges.

There exist $A_{\text{Feas}}, A_{\text{Opt}} \subseteq [n]$ (corresponding to coordinates $\pi(1), \ldots, \pi(r')$ in (5.8)), downsets $\mathcal{S}_{\text{Feas}} \subseteq 2^{A_{\text{Feas}}}$, $\mathcal{S}_{\text{Opt}} \subseteq 2^{A_{\text{Opt}}}$ (corresponding to downset ρ'), and sets of pairs of distinct coordinates $E_{\text{Feas}}, E_{\text{Opt}}$ (corresponding to pairs

$\{\pi(i), a_i\}$ for $i \in \{r'{+}1, \dots, r\}$) such that $|A_{\text{Feas}}|{+}|E_{\text{Feas}}| = |A_{\text{Opt}}|{+}|E_{\text{Opt}}| = n$ and

$$\gamma(X) < \infty \iff X \cap A_{\text{Feas}} \in \mathcal{S}_{\text{Feas}} \;\wedge\; |X \cap \{i,j\}| \neq 1 \text{ for all } \{i,j\} \in E_{\text{Feas}} \tag{5.57}$$

$$\gamma(X) = 0 \iff X \cap A_{\text{Opt}} \in \mathcal{S}_{\text{Opt}} \;\wedge\; |X \cap \{i,j\}| \neq 1 \text{ for all } \{i,j\} \in E_{\text{Opt}}\,. \tag{5.58}$$

We construct an instance J of the GMC problem on vertices $[n]$ as follows. Let $w_{\text{Feas}}(i,j) = \infty$ if $\{i,j\} \in E_{\text{Feas}}$ and $w_{\text{Feas}}(i,j) = 0$ otherwise. Let $w_{\text{Opt}}(i,j) = 1$ if $\{i,j\} \in E_{\text{Opt}}$ and $w_{\text{Opt}}(i,j) = 0$ otherwise. Then the weight of edge (i,j) is $w(i,j) = w_{\text{Feas}}(i,j) + w_{\text{Opt}}(i,j)$. Let f_{Feas} be a set function on $[n]$ defined by $f_{\text{Feas}}(X) = 0$ if $X \cap A_{\text{Feas}} \in \mathcal{S}_{\text{Feas}}$ and $f_{\text{Feas}}(X) = \infty$ otherwise; f_{Feas} is superadditive because $\mathcal{S}_{\text{Feas}}$ is a downset. Let f_{Opt} be a set function on $[n]$ defined by $f_{\text{Opt}}(X) = 0$ if $X \cap A_{\text{Opt}} \in \mathcal{S}_{\text{Opt}}$ and $f_{\text{Opt}}(X) = |X \cap A_{\text{Opt}}|$ otherwise; f_{Opt} is superadditive because $\mathcal{S}_{\text{Opt}}$ is a downset. Then the superadditive function defining instance J is $f = f_{\text{Feas}} + f_{\text{Opt}}$.

By the construction, it holds $\gamma(X) < \infty \iff J(X) < \infty$ and $\gamma(X) = 0 \iff J(X) = 0$. Moreover, for any X such that $0 < J(X) < \infty$, it holds $1 \leq J(X) \leq n$. If the set

$$B = \{\gamma(X) \mid X \subseteq [n] \;\wedge\; 0 < \gamma(X) < \infty\} \tag{5.59}$$

is empty, then instance J 1-approximates γ; otherwise let $b_{\min}, b_{\max}$ denote the minimum and maximum of B. We scale the weights of the edges w and the superadditive function f by a factor of $b_{\min}/n$ to obtain an instance J' such that $J'(X) \leq \gamma(X)$ for all X. Instance J' then $(n \cdot b_{\max}/b_{\min})$-approximates γ. □

To establish the tractability of infinite EDS languages, we need a better bound on the approximability of α-EDS set functions than the one given in Theorem 5.56. This is achieved in Theorem 5.58, which we prove using the following technical lemma. We refer the reader to [38, Theorem 1.1] for an example of the application of this proof technique in a simpler setting.

Lemma 5.57. *Let γ be an α-EDS set function on V for some $\alpha \geq 1$. For any distinct $u, v \in V$, let $T_{\{u,v\}}$ be a subset of V such that $\left|T_{\{u,v\}} \cap \{u,v\}\right| = 1$. Then, for any $R \subseteq S \subseteq V$, it holds*

$$\alpha^{|S|+2} \cdot \left(\left(|S|^2 + 2\right) \cdot \gamma(S) + \sum_{|R \cap \{u,v\}| = 1} \gamma\left(T_{\{u,v\}}\right) \right) \geq \gamma(R)\,. \tag{5.60}$$

Proof. First, we show by induction that, for any $X, Y_1, \ldots, Y_n \subseteq V$, it holds

$$\alpha^n \cdot \left(\gamma(X) + \sum_{i=1}^{n} \gamma(Y_i)\right) \geq \gamma\left(X \setminus \bigcup_{i=1}^{n} Y_i\right) . \tag{5.61}$$

For $n = 1$, this is equivalent to (5.54). As for the inductive step, assume that (5.61) holds for $n \geq 1$. By the inductive hypothesis and (5.54), we get

$$\alpha^{n+1} \cdot \left(\gamma(X) + \sum_{i=1}^{n+1} \gamma(Y_i)\right) \geq \alpha \cdot \left(\gamma\left(X \setminus \bigcup_{i=1}^{n} Y_i\right) + \gamma(Y_{n+1})\right) \tag{5.62}$$

$$\geq \gamma\left(X \setminus \bigcup_{i=1}^{n+1} Y_i\right) . \tag{5.63}$$

If $\gamma(S) = \infty$, the inequality claimed by this lemma trivially holds. In the rest of the proof, we assume $\gamma(S) < \infty$. For any $u \in R$, $v \in S \setminus R$, we define a set T'_{uv} such that $T'_{uv} \cap \{u, v\} = \{v\}$: If $v \in T_{\{u,v\}}$, let $T'_{uv} = T_{\{u,v\}}$; otherwise let $T'_{uv} = S \setminus T_{\{u,v\}}$. We claim that

$$\alpha \cdot \left(\gamma(S) + \gamma\left(T_{\{u,v\}}\right)\right) \geq \gamma(T'_{uv}) . \tag{5.64}$$

This is trivially true in the case of $T'_{uv} = T_{\{u,v\}}$, and it follows from (5.54) in the case of $T'_{uv} = S \setminus T_{\{u,v\}}$. By (5.64), it holds

$$\sum_{|R \cap \{u,v\}|=1} \gamma\left(T_{\{u,v\}}\right) \geq \sum_{u \in R} \sum_{v \in S \setminus R} \gamma\left(T_{\{u,v\}}\right) \tag{5.65}$$

$$\geq \frac{1}{\alpha} \sum_{u \in R} \sum_{v \in S \setminus R} \gamma(T'_{uv}) - |R| \cdot |S \setminus R| \cdot \gamma(S) . \tag{5.66}$$

For any $u \in R$, let

$$W_u = S \setminus \bigcup_{v \in S \setminus R} T'_{uv} . \tag{5.67}$$

By properties of T'_{uv}, it holds $u \in W_u \subseteq R$. Moreover, we have

$$\alpha^{|S \setminus R|} \cdot \left(\gamma(S) + \sum_{v \in S \setminus R} \gamma(T'_{uv})\right) \geq \gamma(W_u) \tag{5.68}$$

by (5.61), which together with (5.66) gives us

$$\sum_{|R \cap \{u,v\}|=1} \gamma\left(T_{\{u,v\}}\right) \geq \frac{1}{\alpha^{|S \setminus R|+1}} \sum_{u \in R} \gamma(W_u) - |R| \cdot (|S \setminus R| + 1) \cdot \gamma(S) \tag{5.69}$$

$$\geq \frac{1}{\alpha^{|S \setminus R|+1}} \sum_{u \in R} \gamma(W_u) - |S|^2 \cdot \gamma(S) . \tag{5.70}$$

As it holds $\bigcup_{u\in R} W_u = R$, we have

$$\alpha^{|R|} \cdot \left(\gamma(S) + \sum_{u\in R} \gamma(W_u)\right) \geq \gamma(S \setminus R)\,, \tag{5.71}$$

and hence

$$\sum_{|R\cap\{u,v\}|=1} \gamma\left(T_{\{u,v\}}\right) \geq \frac{1}{\alpha^{|S|+1}} \cdot \gamma(S \setminus R) - \left(|S|^2 + 1\right) \cdot \gamma(S)\,. \tag{5.72}$$

As it holds $\alpha \cdot (\gamma(S) + \gamma(S \setminus R)) \geq \gamma(R)$, this proves the claimed inequality. □

Theorem 5.58. *Any α-EDS set function on V is $\alpha^{n+2}\left(n^3 + 2n\right)$-approximable, where $n = |V|$.*

Proof. Let γ be an α-EDS set function on V for some $\alpha \geq 1$. We construct an instance J of the GMC problem on vertices V such that it $\alpha^{n+2}\left(n^3 + 2n\right)$-approximates γ. The weight of edge (u, v) is

$$w(u,v) = \frac{1}{n^3 + 2n} \cdot \min\left\{\gamma(Z) \mid Z \subseteq V \wedge |Z \cap \{u,v\}| = 1\right\}\,. \tag{5.73}$$

Let f be a set function on V defined as

$$f(X) = \frac{|X|}{n^3 + 2n} \cdot \min\left\{\left(|Z|^2 + 2\right) \cdot \gamma(Z) \;\middle|\; X \subseteq Z \subseteq V\right\}\,. \tag{5.74}$$

We claim that f is a superadditive set function. As $\gamma(\emptyset) = 0$, it holds $f(\emptyset) = 0$. Consider any disjoint $X, Y \subseteq V$ and let $Z \supseteq X \cup Y$ be a minimiser in (5.74) for $f(X \cup Y)$. It holds $f(X) \leq |X| \cdot (|Z|^2 + 2) \cdot \gamma(Z) / \left(n^3 + 2n\right)$ and $f(Y) \leq |Y| \cdot (|Z|^2 + 2) \cdot \gamma(Z) / \left(n^3 + 2n\right)$, and hence

$$f(X) + f(Y) \leq \frac{|X \cup Y|}{n^3 + 2n} \cdot \left(|Z|^2 + 2\right) \cdot \gamma(Z) = f(X \cup Y)\,. \tag{5.75}$$

The edge weights w and superadditive set function f define the GMC instance J. Now we prove that it $\alpha^{n+2}\left(n^3 + 2n\right)$-approximates γ.

First, we show that $J(R) \leq \gamma(R)$ for all $R \subseteq V$. By (5.74), we have $f(R) \leq |R| \cdot (|R|^2 + 2) \cdot \gamma(R) / \left(n^3 + 2n\right)$. For any edge (u, v) cut by R (i.e. $|R \cap \{u, v\}| = 1$), it holds $w(u, v) \leq \gamma(R) / \left(n^3 + 2n\right)$ by (5.73), and hence $g(R) \leq |R| \cdot |V \setminus R| \cdot \gamma(R) / \left(n^3 + 2n\right)$. Together, this gives

$$J(R) = f(R) + g(R) \leq \frac{|R| \cdot (|R|^2 + |V \setminus R| + 2)}{n^3 + 2n} \cdot \gamma(R) \leq \gamma(R)\,. \tag{5.76}$$

Second, we show that $\alpha^{n+2}\left(n^3+2n\right)\cdot J(R)\geq\gamma(R)$ for all $R\subseteq V$. For $R=\emptyset$, the inequality holds, as $J(\emptyset)=\gamma(\emptyset)=0$. Otherwise, let $S\supseteq R$ be a minimiser in (5.74) for $f(R)$, and $T_{\{u,v\}}$ a minimiser in (5.73) for any edge (u,v). It holds

$$\left(n^3+2n\right)\cdot f(R)=|R|\cdot\left(|S|^2+2\right)\cdot\gamma(S)\geq\left(|S|^2+2\right)\cdot\gamma(S) \tag{5.77}$$

$$\left(n^3+2n\right)\cdot g(R)=\sum_{|R\cap\{u,v\}=1|}\gamma\left(T_{\{u,v\}}\right)\,, \tag{5.78}$$

and therefore, by Lemma 5.57, $\alpha^{n+2}\left(n^3+2n\right)\cdot J(R)\geq\alpha^{|S|+2}\left(n^3+2n\right)\cdot J(R)\geq\gamma(R)$. □

Theorem 5.59. *Any EDS language is globally s-tractable.*

Proof. Let Γ be an EDS language and $\alpha'\geq 1$ such that every weighted relation in Γ is α'-EDS. Without loss of generality, we may assume that $\gamma\left(\mathbf{0}^{\mathrm{ar}(\gamma)}\right)=0$ for every $\gamma\in\Gamma$, and hence identify weighted relations with their corresponding set functions. Weighted relations in Γ are of bounded arity and therefore, by Theorem 5.58, there exists α such that every $\gamma\in\Gamma$ is α-approximable. We will denote by J_γ a GMC instance that α-approximates γ.

Given a $\mathrm{VCSP_s}(\Gamma)$ instance I with an objective function

$$\phi'_I(x_1,\ldots,x_n)=\sum_{i=1}^{q}w_i\cdot\gamma_i(\mathbf{x}^i)\,, \tag{5.79}$$

we denote by ϕ_I the corresponding set function and construct a GMC instance J that α-approximates ϕ_I. For $i\in[q]$, we relabel the vertices of J_{γ_i} to match the variables in the scope $\mathbf{x}^i$ of the ith constraint (i.e., vertex j is relabelled to x^i_j) and identify vertices in case of repeated variables. As the constraint is weighted by a non-negative factor w_i, we also scale the weights of the edges of J_{γ_i} and the superadditive function by w_i. (Note that non-negative scaling preserves superadditivity.) Instance J is then obtained by adding up GMC instances J_{γ_i} for all $i\in[q]$.

Let $\mathbf{x}\in D^n$ denote a surjective assignment minimising ϕ'_I, $X\subseteq[n]$ the corresponding set $\{i\in[n]\mid x_i=1\}$, $Y\subseteq[n]$ an optimal solution to J, and $\lambda=J(Y)$. Because J α-approximates ϕ_I, it holds

$$\lambda\leq J(X)\leq\phi_I(X)\leq\phi_I(Y)\leq\alpha\cdot J(Y)=\alpha\lambda\,, \tag{5.80}$$

and hence X is an α-optimal solution to J. By Lemma 5.44, we can determine whether $\lambda=0$, in which case any optimal solution to J is also optimal for ϕ_I; and whether $\lambda=\infty$. If $0<\lambda<\infty$, we find all α-optimal solutions by Theorem 5.50. □

We now prove Theorem 5.13.

Proof. We only need to prove the theorem in the case of an EDS language (whether Γ or $\neg(\Gamma)$, which is symmetric), as the remaining classes of globally s-tractable languages include the constants ρ_0, ρ_1 and thus admit a polynomial-delay algorithm using standard self-reduction techniques [30, 24].

Let Γ be an EDS language. As in the proof of Theorem 5.59, we may assume that every weighted relation in Γ assigns 0 as the minimum value. Given an instance of $\mathrm{VCSP}_{\mathrm{s}}(\Gamma)$, we can determine in polynomial time, by Lemma 5.44, whether $\lambda = 0$, $0 < \lambda < \infty$, or $\lambda = \infty$. If $\lambda = 0$, then optimal solutions incur the minimum value from every constraint. By applying Opt to all constraints, we obtain a CSP instance invariant under min (by Lemma 5.9), and hence are able to enumerate all optimal solutions with a polynomial delay by the results in [30]. If $0 < \lambda < \infty$, then the claim follows from the proof of Theorem 5.59; moreover, the number of optimal solutions is polynomially bounded (see Theorem 5.50). Finally, the case $\lambda = \infty$ is trivial. $\square$

Chapter 6

Conclusion

In Chapter 3, we showed that the Galois connection between weighted relational clones and weighted clones established in [25] does not hold for infinite languages and infinite sets of weightings. We modified the definition of weighted (relational) clones by requiring them to be topologically closed, and established a new Galois connection which applies to infinite sets as well. We additionally required that weighted relational clones be closed under operation Opt, which simplified the structure of weighted clones.

The algebraic approach has facilitated rapid progress in research on the complexity of language-restricted VCSP. Hence, extending it to other variants (such as the VCSP with global constraints, hybrid restrictions, or infinite domains) would be helpful in classifying constraint languages in those settings and lead to new insights. More generally, (weighted) polymorphisms may have interesting applications for other problems as well, because the thesis that symmetry enables efficient tractability is not limited to the (V)CSP only.

In Chapter 4, we classified all conservative languages with respect to their tractability under the planarity structural restriction. Our classification precisely matches the analogous result for general instances [73]; in other words, every intractable conservative language remains NP-hard to solve even when restricted to planar instances. On the other hand, there exist intractable (non-conservative) Boolean languages that become solvable in polynomial time under the planarity restriction. We proved a necessary condition such languages must satisfy, thus generalising the condition of Dvořák and Kupec [41] for the crisp case. For languages satisfying our condition, we showed a reduction to the edge VCSP over valuated delta-matroids.

Generalising the algorithm of Kazda, Kolmogorov, and Rolínek [69] to the case of valuated delta-matroids would prove the sufficiency of our condition and complete the classification of the Boolean planar VCSP. Another potential direction for generalisation of this algorithm is to solve the edge CSP over delta-matroids (as opposed to *even* delta-matroids), which would settle an open question about the complexity of the CSP with fanout limitation posed by Feder [45]. As for the planar (V)CSP on larger domains, a complete classification seems elusive, especially due to the 4-colouring problem being trivially tractable in consequence of the four colour theorem. However, it might be feasible to classify languages on domains of size 3. Finally, structural restrictions generalising planarity (e.g., graphs with a bounded genus or even graph classes characterised by forbidden minors) might be tackled by techniques developed for the planar setting. In particular, in the case of the Boolean planar CSP, the planarity of an instance is utilised only in the reduction to the edge CSP over even delta-matroids, which is tractable without assuming planarity.

In Chapter 5, we classified all Boolean languages with respect to the computational complexity of the corresponding surjective VCSP. We identified a condition called EDS which characterises a new tractable class of languages (i.e., those languages are trivially tractable if we allow non-surjective solutions but require a non-trivial algorithm in the surjective setting). We showed that, in contrast to tractable classes characterised by a multimorphism, EDS languages do not exhibit compactness: There exists an (intractable) infinite non-EDS language such that its every finite subset satisfies the EDS condition. We established the tractability of EDS languages by a reduction to the Generalised Min-Cut (GMC) problem, which further generalises the problem introduced by Uppman [102], and an algorithm that finds all α-optimal solutions to a GMC instance in polynomial time (for a fixed α).

The surjective VCSP over languages on larger domains captures some simple-to-state problems whose complexity remains open, for example the 3-No-Rainbow-Colouring problem [7]. Hence, even a complete classification of languages on domains of size 3 seems very challenging. On the other hand, it might be feasible to identify a new tractable class analogous to EDS languages which can be solved by a reduction to a generalisation of the minimum k-cut problem for $k \geq 3$.

Bibliography

[1] W. Bach and H. Zhou. Approximation for Maximum Surjective Constraint Satisfaction Problems. Technical report, October 2011. arXiv:1110.2953.

[2] L. Barto. The Dichotomy for Conservative Constraint Satisfaction Problems Revisited. In *Proceedings of the 26th Annual IEEE Symposium on Logic in Computer Science (LICS'11)*, pages 301–310, 2011.

[3] L. Barto and M. Kozik. Constraint Satisfaction Problems Solvable by Local Consistency Methods. *Journal of the ACM*, 61(1), 2014. Article No. 3.

[4] 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 Journal on Computing*, 38(5):1782–1802, 2009.

[5] L. Barto, A. Krokhin, and R. Willard. Polymorphisms, and how to use them. In Krokhin and Živný [76], pages 1–44.

[6] J. Berman, P. Idziak, P. Marković, R. McKenzie, M. Valeriote, and R. Willard. Varieties with few subalgebras of powers. *Transactions of the American Mathematical Society*, 362(3):1445–1473, 2010.

[7] M. Bodirsky, J. Kára, and B. Martin. The complexity of surjective homomorphism problems - a survey. *Discrete Applied Mathematics*, 160(12):1680–1690, 2012.

[8] M. Bodirsky and M. Mamino. Constraint Satisfaction Problems over Numeric Domains. In Krokhin and Živný [76], pages 79–111.

[9] V. Bodnarčuk, L. Kalužnin, V. Kotov, and B. Romov. Galois theory for Post algebras. I. *Cybernetics and Systems Analysis*, 5(3):243–252, 1969.

[10] A. Bouchet. Greedy algorithm and symmetric matroids. *Mathematical Programming*, 38(2):147–159, 1987.

[11] A. Bouchet and W. H. Cunningham. Delta-Matroids, Jump Systems, and Bisubmodular Polyhedra. *SIAM Journal on Discrete Mathematics*, 8(1):17–32, 1995.

[12] S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.

[13] A. Bulatov. A dichotomy theorem for constraint satisfaction problems on a 3-element set. *Journal of the ACM*, 53(1):66–120, 2006.

[14] A. Bulatov. A dichotomy theorem for nonuniform CSP. In *Proceedings of the 58th Annual IEEE Symposium on Foundations of Computer Science (FOCS'17)*, pages 319–330. IEEE, 2017.

[15] A. Bulatov, A. Krokhin, and P. Jeavons. Classifying the Complexity of Constraints using Finite Algebras. *SIAM Journal on Computing*, 34(3):720–742, 2005.

[16] A. A. Bulatov. Bounded relational width. Technical report, 2009.

[17] A. A. Bulatov. Complexity of conservative constraint satisfaction problems. *ACM Transactions on Computational Logic*, 12(4), 2011. Article 24.

[18] A. A. Bulatov and V. Dalmau. A Simple Algorithm for Mal'tsev Constraints. *SIAM Journal on Computing*, 36(1):16–27, 2006.

[19] A. A. Bulatov, V. Dalmau, M. Grohe, and D. Marx. Enumerating homomorphisms. *Journal of Computer and System Sciences*, 78(2):638–650, 2012.

[20] A. A. Bulatov and D. Marx. The complexity of global cardinality constraints. *Logical Methods in Computer Science*, 6(4), 2010.

[21] C. Carbonnel, M. Romero, and S. Živný. The complexity of general-valued CSPs seen from the other side. *CoRR*, abs/1710.03148, 2017.

[22] R. Chandrasekaran and S. N. Kabadi. Pseudomatroids. *Discrete Mathematics*, 71(3):205–217, 1988.

[23] H. Chen. An algebraic hardness criterion for surjective constraint satisfaction. *Algebra universalis*, 72(4):393–401, 2014.

[24] D. A. Cohen. Tractable Decision for a Constraint Language Implies Tractable Search. *Constraints*, 9:219–229, 2004.

[25] D. A. Cohen, M. C. Cooper, P. Creed, P. Jeavons, and S. Živný. An algebraic theory of complexity for discrete optimisation. *SIAM Journal on Computing*, 42(5):915–1939, 2013.

[26] D. A. Cohen, M. C. Cooper, P. Jeavons, and A. A. Krokhin. Supermodular functions and the complexity of MAX CSP. *Discrete Applied Mathematics*, 149(1-3):53–72, 2005.

[27] D. A. Cohen, M. C. Cooper, and P. G. Jeavons. Generalising submodularity and Horn clauses: Tractable optimization problems defined by tournament pair multimorphisms. *Theoretical Computer Science*, 401(1-3):36–51, 2008.

[28] D. A. Cohen, M. C. Cooper, P. G. Jeavons, and A. A. Krokhin. The Complexity of Soft Constraint Satisfaction. *Artificial Intelligence*, 170(11):983–1016, 2006.

[29] M. C. Cooper and S. Živný. Hybrid Tractable Classes of Constraint Problems. In Krokhin and Živný [76], pages 113–135.

[30] N. Creignou and J. Hébrard. On generating all solutions of generalized satisfiability problems. *Informatique Théorique et Applications*, 31(6):499–511, 1997.

[31] N. Creignou, S. Khanna, and M. Sudan. *Complexity Classification of Boolean Constraint Satisfaction Problems*, volume 7 of *SIAM Monographs on Discrete Mathematics and Applications*. SIAM, 2001.

[32] P. Crescenzi. A short guide to approximation preserving reductions. In *Proceedings of the 12th Annual IEEE Conference on Computational Complexity*, CCC '97, pages 262–, Washington, DC, USA, 1997. IEEE Computer Society.

[33] V. Dalmau. Generalized majority-minority operations are tractable. *Logical Methods in Computer Science*, 2(4), 2006.

[34] V. Dalmau and D. K. Ford. Generalized satisfiability with limited occurrences per variable: A study through delta-matroid parity. In *Proceedings of the 28th International Symposium on Mathematical Foundations of Computer Science (MFCS'03)*, pages 358–367. Springer, 2003.

[35] R. Dechter. *Constraint processing.* Elsevier Morgan Kaufmann, 2003.

[36] R. Dechter and A. Itai. Finding all solutions if you can find one. In *Workshop on Tractable Reasoning (AAAI'92)*, pages 35–39, 1992.

[37] V. G. Deineko, P. Jonsson, M. Klasson, and A. A. Krokhin. The approximability of MAX CSP with fixed-value constraints. *Journal of the ACM*, 55(4):16:1–16:37, 2008.

[38] N. R. Devanur, S. Dughmi, R. Schwartz, A. Sharma, and M. Singh. On the approximation of submodular functions. April 2013. arXiv:1304.4948.

[39] A. Dress and T. F. Havel. Some combinatorial properties of discriminants in metric vector spaces. *Advances in Mathematics*, 62(3):285 – 312, 1986.

[40] A. W. Dress and W. Wenzel. A greedy-algorithm characterization of valuated δ-matroids. *Applied Mathematics Letters*, 4(6):55 – 58, 1991.

[41] Z. Dvořák and M. Kupec. On Planar Boolean CSP. In *Proceedings of the 42nd International Colloquium on Automata, Languages and Programming (ICALP'15)*, volume 9134 of *Lecture Notes in Computer Science*, pages 432–443. Springer, 2015.

[42] J. Edmonds. Minimum partition of a matroid into independent subsets. *Journal of Research of the National Bureau of Standards*, 69:67–72, 1965.

[43] J. Edmonds. Paths, trees, and flowers. *Canadian Journal of Mathematics*, 17(3):449–467, 1965.

[44] T. Färnqvist and P. Jonsson. Bounded Tree-Width and CSP-Related Problems. In *Proceedings of the 18th International Symposium on Algorithms and Computation (ISAAC'07)*, volume 4835 of *Lecture Notes in Computer Science*, pages 632–643. Springer, 2007.

[45] T. Feder. Fanout limitations on constraint systems. *Theoretical Computer Science*, 255(1-2):281–293, 2001.

[46] T. Feder and M. Y. Vardi. The Computational Structure of Monotone Monadic SNP and Constraint Satisfaction: A Study through Datalog and Group Theory. *SIAM Journal on Computing*, 28(1):57–104, 1998.

[47] A. Frank. *Connections in Combinatorial Optimization.* Oxford Lecture Series in Mathematics and Its Applications. OUP Oxford, 2011.

[48] P. Fulla, H. Uppman, and S. Živný. The complexity of Boolean surjective general-valued CSPs. *ACM Transactions on Computation Theory.* To appear.

[49] P. Fulla and S. Živný. A Galois Connection for Valued Constraint Languages of Infinite Size. In *Proceedings of the 42nd International Colloquium on Automata, Languages, and Programming (ICALP'15)*, pages 517–528, 2015.

[50] P. Fulla and S. Živný. A Galois Connection for Weighted (Relational) Clones of Infinite Size. *ACM Transactions on Computation Theory*, 8(3), 2016. Article No. 9.

[51] P. Fulla and S. Živný. On Planar Valued CSPs. In *Proceedings of the 41st International Symposium on Mathematical Foundations of Computer Science (MFCS'16)*, pages 39:1–39:14, 2016.

[52] P. Fulla and S. Živný. On Planar Valued CSPs. *Journal of Computer and System Sciences*, 87:104–118, 2017.

[53] P. Fulla and S. Živný. The Complexity of Boolean Surjective General-Valued CSPs. In *Proceedings of the 42nd International Symposium on Mathematical Foundations of Computer Science (MFCS'17)*, pages 4:1–4:14, 2017.

[54] M. R. Garey and D. S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness.* W.H. Freeman, 1979.

[55] M. R. Garey, D. S. Johnson, and L. J. Stockmeyer. Some Simplified NP-Complete Graph Problems. *Theoretical Computer Science*, 1(3):237–267, 1976.

[56] D. Geiger. Closed systems of functions and predicates. *Pacific Journal of Mathematics*, 27(1):95–100, 1968.

[57] G. Gottlob, G. Greco, and F. Scarcello. Tractable optimization problems through hypergraph-based structural restrictions. In *Proceedings of the 36th International Colloquium on Automata, Languages and Programming (ICALP'09)*, pages 16–30, 2009.

[58] M. Grohe. The complexity of homomorphism and constraint satisfaction problems seen from the other side. *Journal of the ACM*, 54(1):1–24, 2007.

[59] F. Hadlock. Finding a maximum cut of a planar graph in polynomial time. *SIAM Journal on Computing*, 4(3):221–225, 1975.

[60] P. Hell and J. Nešetřil. On the complexity of H-coloring. *Journal of Combinatorial Theory, Series B*, 48(1):92–110, 1990.

[61] J. Hiriart-Urruty and C. Lemaréchal. *Fundamentals of Convex Analysis*. Grundlehren Text Editions. Springer Berlin Heidelberg, 2001.

[62] J. E. Hopcroft and R. E. Tarjan. Efficient planarity testing. *Journal of the ACM*, 21(4):549–568, 1974.

[63] P. M. Idziak, P. Markovic, R. McKenzie, M. Valeriote, and R. Willard. Tractability and learnability arising from algebras with few subpowers. *SIAM Journal on Computing*, 39(7):3023–3037, 2010.

[64] P. Jeavons, D. Cohen, and M. C. Cooper. Constraints, Consistency and Closure. *Artificial Intelligence*, 101(1–2):251–265, 1998.

[65] D. S. Johnson, M. Yannakakis, and C. H. Papadimitriou. On generating all maximal independent sets. *Information Processing Letters*, 27(3):119 – 123, 1988.

[66] P. Jonsson, M. Klasson, and A. A. Krokhin. The approximability of three-valued MAX CSP. *SIAM Journal on Computing*, 35(6):1329–1349, 2006.

[67] P. Jonsson, F. Kuivinen, and J. Thapper. Min CSP on four elements: Moving beyond submodularity. In *Proceedings of the 17th International Conference on Principles and Practice of Constraint Programming (CP'11)*, pages 438–453, 2011.

[68] D. R. Karger. Global Min-cuts in RNC, and Other Ramifications of a Simple Min-Cut Algorithm. In *Proceedings of the 4th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA'93)*, pages 21–30, 1993.

[69] A. Kazda, V. Kolmogorov, and M. Rolínek. Even Delta-Matroids and the Complexity of Planar Boolean CSPs. In *Proceedings of the 28th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA'17)*, 2017.

[70] V. Kolmogorov, A. A. Krokhin, and M. Rolínek. The Complexity of General-Valued CSPs. *SIAM Journal on Computing*, 46(3):1087–1110, 2017.

[71] 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'15)*, volume 9472 of *Lecture Notes in Computer Science*, pages 566–577. Springer, 2015.

[72] V. Kolmogorov, J. Thapper, and S. Živný. The power of linear programming for general-valued CSPs. *SIAM Journal on Computing*, 44(1):1–36, 2015.

[73] V. Kolmogorov and S. Živný. The complexity of conservative valued CSPs. *Journal of the ACM*, 60(2), 2013. Article No. 10.

[74] M. Kozik, A. Krokhin, M. Valeriote, and R. Willard. Characterizations of several Maltsev conditions. *Algebra universalis*, 73(3):205–224, Jun 2015.

[75] M. Kozik and J. Ochremiak. Algebraic Properties of Valued Constraint Satisfaction Problem. In *Proceedings of the 42nd International Colloquium on Automata, Languages and Programming (ICALP'15)*, volume 9134 of *Lecture Notes in Computer Science*, pages 846–858. Springer, 2015.

[76] A. A. Krokhin and S. Živný, editors. *The Constraint Satisfaction Problem: Complexity and Approximability*, volume 7 of *Dagstuhl Follow-Ups*. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2017.

[77] A. A. Krokhin and S. Živný. The Complexity of Valued CSPs. In *The Constraint Satisfaction Problem: Complexity and Approximability* [76], pages 233–266.

[78] R. E. Ladner. On the Structure of Polynomial Time Reducibility. *Journal of the ACM*, 22(1):155–171, 1975.

[79] B. Larose, B. Martin, and D. Paulusma. Surjective H-Colouring over Reflexive Digraphs. In *Proceedings of the 35th Symposium on Theoretical Aspects of Computer Science (STACS'18)*, pages 49:1–49:14, 2018.

[80] D. Lichtenstein. Planar Formulae and Their Uses. *SIAM Journal on Computing*, 11(2):329–343, 1982.

[81] K. Makarychev and Y. Makarychev. Approximation Algorithms for CSPs. In Krokhin and Živný [76], pages 287–325.

[82] D. Marx. Tractable hypergraph properties for constraint satisfaction and conjunctive queries. *Journal of the ACM*, 60(6):42:1–42:51, 2013.

[83] B. M. E. Moret. Planar NAE3SAT is in P. *SIGACT News*, 19(2):51–54, 1988.

[84] W. Mulzer and G. Rote. Minimum-weight Triangulation is NP-hard. *Journal of the ACM*, 55(2):11:1–11:29, 1998.

[85] K. Murota. Characterizing a valuated delta-matroid as a family of delta-matroids. *Journal of the Operations Research Society of Japan*, 40(4):565–578, 1997.

[86] K. Murota. M-Convex Functions on Jump Systems: A General Framework for Minsquare Graph Factor Problem. *SIAM Journal on Discrete Mathematics*, 20(1):213–226, 2006.

[87] H. Nagamochi and T. Ibaraki. *Algorithmic aspects of graph connectivity*, volume 123. Cambridge University Press New York, 2008.

[88] R. Pöschel and L. Kaluzhnin. *Funktionen- und Relationenalgebren: ein Kapitel der diskreten Mathematik*. Lehrbücher und Monographien aus dem Gebiete der exakten Wissenschaften: Mathematische Reihe. Birkhäuser, 1979.

[89] P. Raghavendra. Optimal algorithms and inapproximability results for every CSP? In *Proceedings of the 40th Annual ACM Symposium on Theory of Computing (STOC'08)*, pages 245–254. ACM, 2008.

[90] N. Robertson, D. P. Sanders, P. Seymour, and R. Thomas. Efficiently Four-coloring Planar Graphs. In *Proceedings of the 28th Annual ACM Symposium on Theory of Computing (STOC'96)*, pages 571–575, New York, NY, USA, 1996. ACM.

[91] F. Rossi, P. van Beek, and T. Walsh, editors. *The Handbook of Constraint Programming*. Elsevier, 2006.

[92] T. J. Schaefer. The Complexity of Satisfiability Problems. In *Proceedings of the 10th Annual ACM Symposium on Theory of Computing (STOC'78)*, pages 216–226. ACM, 1978.

[93] A. Schrijver. *Combinatorial Optimization: Polyhedra and Efficiency*, volume 24 of *Algorithms and Combinatorics*. Springer, 2003.

[94] H. D. Sherali and W. P. Adams. A Hierarchy of Relaxations Between the Continuous and Convex Hull Representations for Zero-One Programming Problems. *SIAM Journal on Discrete Mathematics*, 3(3):411–430, 1990.

[95] Á. Szendrei. *Clones in universal algebra*. Séminaire de mathématiques supérieures. Presses de l'Université de Montréal, 1986.

[96] R. Takhanov. A Dichotomy Theorem for the General Minimum Cost Homomorphism Problem. In *Proceedings of the 27th International Symposium on Theoretical Aspects of Computer Science (STACS'10)*, pages 657–668, 2010.

[97] R. Takhanov. Hybrid VCSPs with Crisp and Valued Conservative Templates. In *Proceedings of the 28th International Symposium on Algorithms and Computation (ISAAC'17)*, pages 65:1–65:13, 2017.

[98] J. Thapper. *Aspects of a Constraint Optimisation Problem.* PhD thesis, Department of Computer Science and Information Science, Linköping University, 2010.

[99] J. Thapper and S. Živný. Necessary Conditions on Tractability of Valued Constraint Languages. *SIAM Journal on Discrete Mathematics*, 29(4):2361–2384, 2015.

[100] J. Thapper and S. Živný. The complexity of finite-valued CSPs. *Journal of the ACM*, 63(4), 2016. Article No. 37.

[101] J. Thapper and S. Živný. The Power of Sherali-Adams Relaxations for General-Valued CSPs. *SIAM Journal on Computing*, 46(4):1241–1279, 2017.

[102] H. Uppman. Max-Sur-CSP on Two Elements. In *Proceedings of the 18th International Conference on Principles and Practice of Constraint Programming (CP'12)*, volume 7514 of *Lecture Notes in Computer Science*, pages 38–54. Springer, 2012.

[103] 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)*, volume 7965 of *Lecture Notes in Computer Science*, pages 804–815. Springer, 2013.

[104] A. Vardy. Algorithmic Complexity in Coding Theory and the Minimum Distance Problem. In *Proceedings of the 29th Annual ACM Symposium on Theory of Computing (STOC'97)*, pages 92–109, New York, NY, USA, 1997. ACM.

[105] V. V. Vazirani and M. Yannakakis. Suboptimal Cuts: Their Enumeration, Weight and Number (Extended Abstract). In *Proceedings of the 19th International Colloquium on Automata, Languages and Programming (ICALP'92)*, pages 366–377. Springer-Verlag, 1992.

[106] W. Wenzel. Δ-matroids with the strong exchange conditions. *Applied Mathematics Letters*, 6(5):67 – 70, 1993.

[107] D. Zhuk. The Proof of CSP Dichotomy Conjecture. In *Proceedings of the 58th Annual IEEE Symposium on Foundations of Computer Science (FOCS'17)*, pages 331–342. IEEE, 2017.