ssdnm
środowiskowe
studia doktoranckie
z nauk matematycznych

Joanna Ochremiak

Uniwersytet Warszawski

Algebraic properties of valued constraint satisfaction problem

Praca semestralna nr 3

(semestr letni 2012/13)

Opiekun pracy: Marcin Kozik

KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI

UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY

ALGEBRAIC PROPERTIES OF VALUED CONSTRAINT SATISFACTION PROBLEM

JOANNA OCHREMIAK

Abstract. The Valued Constraint Satisfaction Problem is an optimisation version of CSP, where the constraints not only determine the allowed combinations of values, but also specify associated costs. The goal is to minimise the aggregate cost. Following Feder and Vardi, we assume that all constraints in an instance must belong to a fixed *valued constraint language*. We show that, while studying the computational complexity of valued constraint languages, it is enough to consider *rigid cores*. This generalizes a result obtained by Huber, Krokhin and Powell for finite-valued CSP. We consider operations called *weighted polymorphisms*, that are known to determine the complexity of VCSPs. We introduce notions of *weighted algebra* and *weighted variety*, and show a connection, with respect to the complexity of the associated languages, between a weighted algebra and a weighted variety it generates.

1. Introduction

In the Valued Constraint Satisfaction Problem (VCSP) each problem instance is given by a set of variables, a domain of values and a set of constraints. Each constraint determines which combinations of values can be taken by certain subsets of variables. Moreover, every combination of values, that is allowed by a constraint, has an associated cost. The goal is to find an assignment that satisfies all constraints and has a minimal total cost [5, 7]. The valued constraint satisfaction problem provides a framework that can be used to express many combinatorial and discrete optimisation problems arising in different fields of computer science.

In the special case when all cost are 0, VCSP is equivalent to the classical Constraint Satisfation Problem (CSP), where each combination of values is either allowed or not. In this case there is no optimisation aspect — the objective is to determine whether there exists an assignment that meets all the constraints. The classical CSP has been studied by computer scientists for over forty years.

Since in general the VCSP is NP-hard, the major line of research tries to identify the restrictions that give rise to tractable classes of problems. In the classical CSP, one way to deal with this is so called *non-uniform* CSP [9], where the set of constraints that may appear in an instance is fixed and finite. The set of allowed constraints is called a *constraint language*. The Dichotomy Conjecture of Feder and Vardi [9] states that, for every constraint language, the CSP defined by it is either NP-complete or solvable in polynomial time. So far there are many partial dichotomy results, but the central question is still open.

In this paper we follow the same approach and study the language-based restrictions on VCSPs. The *valued constraint language* [7] is a set of partial, rational-valued functions on a fixed domain. The functions specify costs associated with allowed combinations of values. Since the VCSP framework is a lot more general than CSP, it is not surprising that much less is known about the complexity of VCSPs defined by certain valued constraint languages. Existing partial results include a full classification of languages on a two-element domain [7] and conservative (containing all $\{0,1\}$-valued unary functions) languages [12]. Recently also finite-valued languages (where the functions are total, and hence allow every combination of values) have been completely classified with respect to exact solvability [14].

The most successful approach to classifying non-uniform CSPs is the algebraic approach [11, 3, 4]. It has been shown that the complexity of any constraint language depends only on a set of operations called *polymorphisms* [11]. Hence, instead of studying the constraint languages, one can study the corresponding finite algebras. We say that an algebra is tractable if the associated CSP is solvable in polynomial time, and we call it NP-complete if the associated CSP is NP-complete.

A lot of recent dichotomy results in the classical CSP framework were obtained by considering whole varities of algebras. This approach is based on the fact that, if an algebra is tractable then so is every finite algebra from the variety that it generates, and if a variety contains an NP-complete algebra then its generating algebra is also NP-complete [2]. The Algebraic Dichotomy Conjecture [4] states that, whenever an algebra associated with a *core* constraint language does not lie in a Taylor variety, then the associated CSP is NP-complete, and otherwise it is solvable in polynomial time[1].

As the algebraic tools turned out to be so useful in search for tractable fragments of CSP, an adaptation of those ideas to VCSP came as a natural consequence. *Weighted polymorphism* were introduced and it has been shown that they determine the complexity of VCSP, for a fixed valued constraint language [5].

In [10] and [14] the new algebraic theory of valued constraint languages have been used in the setting of finite-valued CSPs, where the constraints allow every combination of values (only with different costs). In both papers the generalized notion of a *core* language was introduced. It was proven that for each finite-valued constraint language there exists an equivalent core language.

In this paper we generalize the above result to arbitrary VCSPs and prove that the class of languages under consideration might be even further reduced to *rigid cores*. Moreover, we introduce a notion of a *weighted variety* and show how the connection between the complexity of CSPs and properties of varieties generated by finite algebras may be adopted to VCSP. Our results suggest a possible new ways of

[1]The first part of the Algebraic Dichotomy Conjecture is a theorem that was proven by Bulatov, Jeavons and Krokhin in the same paper where they state the general conjecture [4].

characterizing tractable cases of valued constraint satisfaction problem, in the spirit of the Algebraic Dichotomy Conjecture.

The paper is organized as follows. In section 2 we define the valued constraint satisfaction problem. In sections 3 and 4 we focus on the algebraic theory of valued constraint languages, introduce *weighted relational clones*, *weighted clones*, and explain how the complexity of any valued constraint language is determined by an associated weighted clone. In section 5 we briefly recall some basic notions of universal algebra. In section 6 we reduce the class of languages that we need to consider to core languages and then to rigid cores. Finally, in section 7 we define weighted algebras and varieties, and establish a connection, with respect to the complexity, between a weighted algebra and a weighted variety that it generates.

2. The Valued Constraint Satisfaction Problem

Let D be a finite set. A *weighted relation* on D of arity r is a partial function from D^r to $\mathbb{Q}$. We denote by Φ_D the set of all weighted relations on D.

Definition 2.1. An *instance of the valued constraint satisfaction problem (VCSP)* is a triple $\mathcal{I} = (V, D, \mathcal{C})$ with V a finite set of *variables*, D a finite *domain* and $\mathcal{C}$ a finite list of *constraints*, where each constraint is a pair $C = (\sigma, \varrho)$ with σ a tuple of variables of length r and ϱ a weighted relation on D of arity r.

An *assignment* for $\mathcal{I}$ is a mapping $s\colon V \to D$. We say that s *satisfies* a constraint $C = (\sigma, \varrho)$ if the weighted relation ϱ is defined for $s(\sigma)$ (where s is applied component-wise). An assignment which satisfies all the constraints of the instance is called *feasible*. The *cost* of an assignment s, denoted $Cost_{\mathcal{I}}(s)$, is given by $Cost_{\mathcal{I}}(s) = \sum_{(\sigma,\varrho)\in\mathcal{C}} \varrho(s(\sigma))$. The goal is to find a feasible assignment with a minimal cost.

Example 2.2. (Max-Cut) In the Max-Cut problem, one needs to find a partition of the vertices of a given graph into two sets, such that the number of edges with ends in different sets is maximal. This problem is NP-hard.

The Max-Cut problem can be expressed as an instance of VCSP. The domain has two elements 0 and 1. Variables in the instance are vertices of the graph and for each edge e there is a constraint of a form $(e, \varrho_{\mathrm{Max}})$, where ϱ_{Max} is a binary weighted relation defined by

$$\varrho_{\mathrm{Max}}(x, y) = \begin{cases} 1 & \text{if } x = y \\ 0 & \text{otherwise.} \end{cases}$$

Any assignment of the values 0 and 1 to the variables corresponds to a partition of the graph. The cost of an assignment is equal to the number of edges of the graph minus the number of cut edges.

A set Γ of weighted relations over a fixed set D is called a *valued constraint language*. An *instance of* VCSP(Γ) is an instance of the VSCP in which all weighted relations in all constraints are from Γ. For such an instance $\mathcal{I}$ we denote by $\mathrm{Opt}_\Gamma(\mathcal{I})$

the minimal cost of a feasible assignment. If all assignments are infeasible then $\mathrm{Opt}_\Gamma(\mathcal{I})$ is undefined.

We say that a valued constraint language Γ is *tractable* if, for every finite subset $\Gamma' \subseteq \Gamma$, there exist an algorithm solving any instance $\mathcal{I} \in \mathrm{VCSP}(\Gamma')$ in polynomial time. Conversely, Γ is said to be *NP-hard* if $\mathrm{VCSP}(\Gamma')$ is NP-hard for some finite $\Gamma' \subseteq \Gamma$. The example 2.2 shows that the valued constraint language $\{\varrho_{\mathrm{MAX}}\}$ is NP-hard.

3. Weighted Relational Clones

In this section we will define two closure operators on valued constraint languages that preserve tractability.

Definition 3.1. For a valued constraint language $\Gamma \subseteq \Phi_D$, let $\langle\Gamma\rangle$ be the set of all weighted relations ϱ for which we can find an instance $\mathcal{I}_\varrho \in \mathrm{VCSP}(\Gamma)$ and a list $(v_1, \ldots, v_r)$ of variables of $\mathcal{I}_\varrho$, such that

$$\varrho(x_1, \ldots, x_r) = \min_{\{s\colon V \to D \;|\; (s(v_1),\ldots,s(v_r))=(x_1,\ldots,x_r)\}} Cost_{\mathcal{I}_\varrho}(s).$$

We say that a weighted relation ϱ is *expressible* over Γ and call $\langle\Gamma\rangle$ the *expressive power* of Γ.

Note that the list of variables $(v_1, \ldots, v_r)$ in the definition above might contain repeated entries. Hence, there can be no assignments s, such that $(s(v_1), \ldots, s(v_r)) = (x_1, \ldots, x_r)$. We assume that the minimum over an empty set is undefined.

For any valued constraint language Γ, over any domain D, there is an instance $\mathcal{I}$ of $\mathrm{VCSP}(\Gamma)$ with a single variable v and no constraints. Therefore, considering a list of variables (v, v), we may express a weighted relation

$$\varrho_=(x, y) = \begin{cases} 0 & \text{if } x = y \\ undefined & \text{otherwise.} \end{cases}$$

The relation $\varrho_=$ is called the *weighted equality relation* and is expressible over any valued constraint language.

Theorem 3.2. (Cohen, Cooper, Creed, Jeavons and Živný [5]). *A valued constraint language Γ is tractable if and only if its expressive power $\langle\Gamma\rangle$ is tractable, and it is NP-hard if and only if $\langle\Gamma\rangle$ is NP-hard.*

Definition 3.3. If a weighted relation $\varrho' \in \Phi_D$ can be obtained from a weighted relation $\varrho \in \Phi_D$ by *non-negative scaling* and *addition of constants* i.e. $\varrho' = a \cdot \varrho + b$ for some $a, b \in \mathbb{Q}$ with $a \geq 0$, then we write $\varrho' \equiv \varrho$.

For $\Gamma \subseteq \Phi_D$ we denote by $\Gamma_\equiv$ the set $\{\varrho' \mid \varrho' \equiv \varrho \text{ for some } \varrho \in \Gamma\}$, which is the smallest set of weighted relations containing Γ and closed under non-negative scaling and addition of constants.

Theorem 3.4. (Cohen, Cooper, Creed, Jeavons and Živný [5]). *A valued constraint language* Γ *is tractable if and only if* $\Gamma_{\equiv}$ *is tractable, and it is NP-hard if and only if* $\Gamma_{\equiv}$ *is NP-hard.*

The Theorems 3.2 and 3.4 show that we can restrict our attention to languages of the form $\langle\Gamma_{\equiv}\rangle$.

Definition 3.5. If $\varrho(x_1,\dots,x_r) = \varrho_1(y_1,\dots,y_s)+\varrho_2(z_1,\dots,z_t)$ for some fixed choice of arguments $y_1,\dots,y_s$ and $z_1,\dots,z_t$ from amongst $x_1,\dots,x_r$ then we say that the weighted relation ϱ is obtained by *addition* from the weighted relations ϱ_1 and ϱ_2.

Definition 3.6. A set $\Gamma \subseteq \Phi_D$ is a *weighted relational clone* if it contains the weighted equality relation and is closed under non-negative scaling, addition of constants, addition, and minimisation over arbitrary arguments.

For a valued constraint language Γ we denote by $\mathrm{wRelClo}(\Gamma)$ the smallest weighted relational clone containing Γ. As an easy corollary of Definitions 3.1, 3.3 and 3.6 we obtain the following:

Proposition 3.7. *For any valued constraint language* Γ*, we have* $\langle\Gamma_{\equiv}\rangle = \mathrm{wRelClo}(\Gamma)$.

4. Weighted polymorphisms

The notion of a *weighted polymorphism*, which we introduce in this section, provide an alternative characterisation for weighted relational clones. First let us recall the basic definitions of the algebraic approach to the classical constraint satisfaction problem. For any set D, a k-ary *operation* on D is a function $f\colon D^k \to D$. We denote by $\mathcal{O}_D$ the set of all finitary operations on D and by $\mathcal{O}_D^{(k)}$ the set of all k-ary operations on D.

Definition 4.1. Let $f \in \mathcal{O}_D^{(k)}$ and $g_1,\dots,g_k \in \mathcal{O}_D^{(l)}$. The l-ary operation $f[g_1,\dots,g_k]$, called the *superposition* of f and $g_1,\dots,g_k$, is defined by $f[g_1,\dots,g_k](x_1,\dots,x_l) = f(g_1(x_1,\dots,x_l),\dots,g_k(x_1,\dots,x_l))$.

Definition 4.2. A set $C \subseteq \mathcal{O}_D$ is a *clone of operations* (or simply a *clone*) if it contains all projections on D and is closed under superposition. The set of k-ary operations in a clone C will be denoted $C^{(k)}$.

The smallest possible clone of operations over a fixed set D is the set of all projections on D, which we denote Π_D.

Definition 4.3. Let ϱ be a weighted relation of arity r on a set D and let f be a k-ary operation on D. We call f a *polymorphism* of ϱ if, for any list of r-tuples $\mathbf{x_1},\dots,\mathbf{x_k}$ for which ϱ is defined, $\varrho(f(\mathbf{x_1},\dots,\mathbf{x_k}))$ is also defined (where f is applied coordinatewise).

For a valued constraint language Γ we denote by $\mathrm{Pol}(\Gamma)$ the set of operations which are polymorphisms of all weighted relations $\varrho \in \Gamma$. The set of k-ary operations in $\mathrm{Pol}(\Gamma)$ is denoted $\mathrm{Pol}^{(k)}(\Gamma)$.

Having defined a clone of operations and a polymorphism, we can now introduce the relatively new algebraic tools designed for analysing the complexity of the valued constraint satisfaction problem [5, 6, 7].

Definition 4.4. A k-ary *weighting* of a clone C is a function $\omega\colon C^{(k)} \to \mathbb{Q}$ such that $\Sigma_{f\in C^{(k)}}\ \omega(f) = 0$, and if $\omega(f) < 0$ then f is a projection.

Note that, if we multiply a weighting ω by a non-negative rational, we get a new weighting. The same happens if we add two weightings of the same arity. A new weighting may be also obtained by a *superposition* with operations from the clone.

Definition 4.5. Let ω be a k-ary weighting of a clone C and let $g_1, \ldots, g_k \in C^{(l)}$. A *superposition* of ω and $g_1, \ldots, g_k$ is a function $\omega[g_1, \ldots, g_k]\colon C^{(l)} \to \mathbb{Q}$ defined by

$$\omega[g_1, \ldots, g_k](f') \;=\; \sum_{\{f\in C^{(k)}\ \mid\ f[g_1,\ldots,g_k]=f'\}} \omega(f).$$

Since the sum of weights that any superposition $\omega[g_1, \ldots, g_k]$ assigns to the operations in $C^{(l)}$ is equal to the sum of weights in ω, it is easy to see that $\omega[g_1, \ldots, g_k]$ satisfies the first condition in the Definition 4.4. It may happen that a superposition assigns a negative value to an operation that is not a projection, violating the second condition. A superposition is said to be *proper* if the result is a valid weighting.

Definition 4.6. A non-empty set of weightings over a fixed clone C is called a *weighted clone* if it is closed under non-negative scaling, addition of weightings of equal arity and proper superposition with operations from C.

Example 4.7. For any clone of operations C, consider the set of all weightings over C and the set of all zero-valued weightings of C. Both those sets are weighted clones. The clone of zero-valued weightings contains exactly one weighting of each arity.

Definition 4.8. Take ϱ to be a weighted relation of arity r on a set D, and let $C \subseteq \mathrm{Pol}(\{\varrho\})$ be a clone of operations. A weighting $\omega\colon C^{(k)} \to \mathbb{Q}$ is called a *weighted polymorphism* of ϱ if, for any $\mathbf{x_1}, \ldots, \mathbf{x_k} \in D^r$ such that $\varrho(\mathbf{x_i})$ is defined, we have

$$\sum_{f\in C^{(k)}} \omega(f) \cdot \varrho(f(\mathbf{x_1}, \ldots, \mathbf{x_k})) \leq 0.$$

For a valued constraint language Γ we denote by $\mathrm{wPol}(\Gamma)$ the set of those weightings of the clone $\mathrm{Pol}(\Gamma)$ that are weighted polymorphisms of all weighted relations $\varrho \in \Gamma$. The set of k-ary weightings in $\mathrm{wPol}(\Gamma)$ is denoted $\mathrm{wPol}^{(k)}(\Gamma)$.

Example 4.9. For any lattice-ordered set D, a function $\varrho\colon D^r \to \mathbb{Q}$ is called *submodular* if

$$\varrho(\min(\mathbf{x_1}, \mathbf{x_2})) + \varrho(\max(\mathbf{x_1}, \mathbf{x_2})) - \varrho(\mathbf{x_1}) - \varrho(\mathbf{x_2}) \leq 0 \quad \text{for all } \mathbf{x_1}, \mathbf{x_2} \in D^r.$$

The above condition can be equivalently expressed by saying that the set of submodular functions on D is the set of weighted relations with a binary weighted polymorphism ω, defined as follows:

$$\omega(f) = \begin{cases} -1 & \text{if } f \text{ is a projection} \\ 1 & \text{if } f \text{ is one of the operations } \min \text{ or } \max \\ 0 & \text{otherwise.} \end{cases}$$

Definition 4.10. A weighted relation ϱ is said to be *improved* by a weighting ω if ω is a weighted polymorphism of ϱ. For any set W of weightings over a fixed clone $C \subseteq \mathcal{O}_D$ we denote by $\mathrm{Imp}(W)$ the set of weighted relations on D which are improved by all weightings $\omega \in W$.

Now we can finally state a result which, together with Theorems 3.2 and 3.4, implies that tractable valued constraint languages can be characterised by weighted polymorphisms.

Theorem 4.11. (Cohen, Cooper, Creed, Jeavons and Živný [5]). *For any finite valued constraint language* Γ*, we have* $\mathrm{Imp}(\mathrm{wPol}(\Gamma)) = \mathrm{wRelClo}(\Gamma)$.

5. Algebras and varieties

In this section we introduce the basic concepts of universal algebra that will serve us as tools later on in this paper. An *algebraic signature* is a set of function symbols together with (finite) arities. An *algebra* $\mathbf{A}$ over a fixed signature Σ consists of a set A, called the *universe* of $\mathbf{A}$, and a set of *basic operations* that correspond to the symbols in the signature i.e. if the signature contains a k-ary symbol f then the algebra has a basic operation $f^{\mathbf{A}}$, which is a function $f^{\mathbf{A}}\colon A^k \to A$.

Definition 5.1. If the set of basic operations of an algebra $\mathbf{A}$ is a clone of operations, we call the algebra $\mathbf{A}$ a *c-algebra.*

Example 5.2. Let Γ be a valued constraint language over a domain D. The set D together with the set of polymorphisms $\mathrm{Pol}(\Gamma)$ is an algebra. This algebra is obviously a c-algebra.

Now let us recall the definitions of three basic operations on algebras over a fixed signature Σ.

Definition 5.3. A subset B of the universe of an algebra $\mathbf{A}$ is called a *subuniverse* of $\mathbf{A}$ if it is closed under all operations of $\mathbf{A}$. An algebra $\mathbf{B}$ is a *subalgebra* of $\mathbf{A}$ if B is a subuniverse of $\mathbf{A}$ and the operations of $\mathbf{B}$ are restrictions of all the operations of $\mathbf{A}$ to B.

For any class $\mathcal{K}$ of algebras over a fixed signature Σ we denote by $S(\mathcal{K})$ the set of all subalgebras of algebras in $\mathcal{K}$. If $\mathcal{K} = \{\mathbf{A}\}$ we write $S(\mathbf{A})$ instead of $S(\{\mathbf{A}\})$.

Definition 5.4. Let $(\mathbf{A}_i)_{i \in I}$ be a family of algebras (of the same signature). Their *product* $\Pi_{i \in I} \mathbf{A}_i$ is an algebra with the universe equal to the cartesian product of the A_i's and operations computed coordinatewise. A product of n copies of $\mathbf{A}$ will be denoted $\mathbf{A}^n$ and called a *power* of $\mathbf{A}$.

For any class $\mathcal{K}$ of algebras over a fixed signature Σ we denote by $P(\mathcal{K})$ the set of all products of algebras in $\mathcal{K}$ and by $P_{fin}(\mathcal{K})$ the set of all finite products. If $\mathcal{K} = \{\mathbf{A}\}$ we write $P(\mathbf{A})$ instead of $P(\{\mathbf{A}\})$.

Definition 5.5. We say that an equivalence relation $\sim$ on A is a *congruence* of $\mathbf{A}$ if the following condition is satisfied for all operations f of $\mathbf{A}$:

$$\text{if for all } i \in \{1, \ldots, k\}, \text{ we have } a_i \sim b_i, \text{ then } f(a_1, \ldots, a_k) \sim f(b_1, \ldots, b_k),$$

where k is the arity of f.

Every congruence $\sim$ of $\mathbf{A}$ determines a *quotient* algebra $\mathbf{A}/\sim$. Its universe is the set of the equivalence classes $A/\sim$ and operations are defined using their arbitrarly chosen representatives.

Definition 5.6. For two algebras $\mathbf{A}$ and $\mathbf{B}$ (of the same signature), a *homomorphism* from $\mathbf{A}$ to $\mathbf{B}$ is a function $h\colon A \to B$ that preserves all operations.

It is easy to see, that an image of a homomorphism $h\colon A \to B$ is a subalgebra of $\mathbf{B}$. For any class $\mathcal{K}$ of algebras over a fixed signature Σ we denote by $H(\mathcal{K})$ the set of all homomorphic images of algebras in $\mathcal{K}$. If $\mathcal{K} = \{\mathbf{A}\}$ we write $H(\mathbf{A})$ instead of $H(\{\mathbf{A}\})$.

Definition 5.7. A *variety* is a class of algebras of the same signature closed under taking subalgebras, products and homomorphic images. For an algebra $\mathbf{A}$ we denote by $\mathcal{V}(\mathbf{A})$ the least variety which contains $\mathbf{A}$, and by $\mathcal{V}_{fin}(\mathbf{A})$ — the class of finite algebras from $\mathcal{V}(\mathbf{A})$.

The variety $\mathcal{V}(\mathbf{A})$ can be characterised as follows:

Proposition 5.8. (Tarski [13]). *For any finite algebra* $\boldsymbol{A}$*, we have*

$$\mathcal{V}(\boldsymbol{A}) = HSP(\boldsymbol{A}) \quad \textit{and} \quad \mathcal{V}_{fin}(\boldsymbol{A}) = HSP_{fin}(\boldsymbol{A}).$$

Definition 5.9. A *term* t in a signature Σ is a formal expression built from variables and symbols in Σ that syntactically describe the composition of basic operations. For an algebra $\mathbf{A}$ over the signature Σ a *term operation* $t^{\mathbf{A}}$ is an operation obtained by composing the basic operations of $\mathbf{A}$ according to t.

Note that if an algebra $\mathbf{A}$ is a c-algebra, then for each term operation $t^{\mathbf{A}}$ there exists an equal basic operation $f^{\mathbf{A}}$.

Definition 5.10. Let s and t be a pair of terms in a signature Σ, and $\mathbf{A}$ — an algebra over this signature. We say that $\mathbf{A}$ satisfies the *identity* $s \approx t$ if the term operations $s^{\mathbf{A}}$ and $t^{\mathbf{A}}$ are equal.

Theorem 5.11. (Birkhoff [1]). *A class of algebras $\mathcal{V}$ is a variety if and only if there exists a set of identities such that $\mathcal{V}$ contains precisely those algebras that satisfy all the identities from this set.*

Corollary 5.12. *The variety $\mathcal{V}(\boldsymbol{A})$ is the class of algebras that satisfy all the identities satisfied by $\boldsymbol{A}$.*

Example 5.13. Let $\mathbf{A}$ be an algebra over a signature Σ and let a basic operation $\pi^{\mathbf{A}}\colon A^k \to A$ be a projection on the i-th coordinate. The algebra $\mathbf{A}$ satisfies an identity $\pi(x_1, \dots, x_k) \approx x_i$. Hence, for any algebra $\mathbf{B} \in \mathcal{V}(\mathbf{A})$ the same identity is satisfied, and the basic operation $\pi^{\mathbf{B}}$ is always a projection.

Take $\mathbf{A}$ to be a c-algebra. For each term t there exist a symbol f in the signature, such that $\mathbf{A}$ satisfies the identity $t \approx f$. Therefore, all algebras in $\mathcal{V}(\mathbf{A})$ satisfy $t \approx f$, which means that for all algebras $\mathbf{B} \in \mathcal{V}(\mathbf{A})$, we have $t^{\mathbf{B}} = f^{\mathbf{B}}$.

6. Core Valued Constraint Languages

In this section, unless stated otherwise, we assume that the considered valued constraint languages are finite.

6.1. Positive Clone.

Definition 6.1. The set of operations to which a weighted polymorphism ω assigns positive weights is called the *support* of ω and denoted $\operatorname{supp}(\omega)$.

We will now prove that the operations contained in the supports of all weighted polymorphisms of a given language Γ, together with all the projections, form a clone. We call it the *positive clone* of Γ and denote $\operatorname{Pol}(\Gamma)^+$.

Proposition 6.2. *For any valued constraint language Γ over a domain D the set $\bigcup_{\omega \in \operatorname{wPol}(\Gamma)} \operatorname{supp}(\omega) \cup \Pi_D$ is a clone of operations.*

Proof. We need to show that the set $\operatorname{Pol}(\Gamma)^+$ is closed under superposition. Take an k-ary operation f and a list of l-ary operations $g_1, \dots, g_k$ that all belong to $\operatorname{Pol}(\Gamma)^+$. Let f' be a superposition of those operations.

If f' is a projection, then it clearly belongs to $\operatorname{Pol}(\Gamma)^+$. If f is a projection, or more generally $f' = g_i$ for some $i \in \{1, \dots, k\}$, then also $f' \in \operatorname{Pol}(\Gamma)^+$.

Otherwise, there exists a weighted polymorphism ω such that $\omega(f) > 0$. Let us consider a superposition $\omega' = \omega[g_1, \dots, g_k]$. We have

$$\omega'(f') = \sum_{\{h \in \operatorname{Pol}^{(k)}(\Gamma) \;\mid\; h[g_1,\dots,g_k] = f'\}} \omega(h).$$

Notice that one of the components in the sum above is $\omega(f)$ and, since $f' \neq g_i$ for each $i \in \{1, \dots, k\}$, there are no projections amongst the operations in the set $\{h \in \operatorname{Pol}^{(k)}(\Gamma) \mid h[g_1, \dots, g_k] = f'\}$. Therefore $\omega'(f') \geq \omega(f) > 0$. Hence, if

$\omega[g_1, \dots, g_k]$ is a proper superposition, then ω' is a weighted polymorphism of Γ with $f' \in \mathrm{supp}(\omega')$ and we are done.

It remains to deal with the case when ω' is not a valid weighting, which means that ω' assigns a negative weight to at least one operation that is not a projection. The only operations that can have negative weights are $g_1, \dots, g_k$. Suppose that $\omega'(g_i) < 0$ and g_i is not a projection. As g_i belongs to $\mathrm{Pol}(\Gamma)^+$ there exists a weighted polymorphism ω_i such that $\omega_i(g_i) > 0$. We solve our problem by adding to ω' the weighting ω_i with a large enough coefficient, and repeat this procedure until all operations g_i (that are not projections) have non-negative weights. The operation f' still has a positive weight assigned, and it is easy to check that the obtained weighting is a weighted polymorphism of Γ. □

The next result implies that if $\mathrm{Pol}(\Gamma)^+$ contains only projections then the valued constraint language Γ is NP-hard.

Proposition 6.3. (Cohen, Cooper, Creed, Jeavons and Živný [5]). *For any valued constraint language Γ, either Γ is NP-hard, or else $\mathrm{wPol}(\Gamma)$ contains a weighting which assigns a positive weight to at least one operation that is not a projection.*

6.2. **Cores.** Let Γ be a valued constraint language with a domain D. If $S \subseteq D$ we denote by $\Gamma[S]$ a valued constraint language defined on a domain S and containing the restriction of every weighted relation $\varrho \in \Gamma$ to S.

Definition 6.4. A valued constraint language Γ is a *core* if for every unary weighted polymorphism ω of Γ, $\mathrm{supp}(\omega)$ contains only bijective operations.

Proposition 6.5. *For each valued constraint language Γ there exists a core language Γ', such that the valued constraint language Γ is tractable if and only if Γ' is tractable, and it is NP-hard if and only if Γ' is NP-hard.*

To prove the result above, we need an auxiliary lemma.

Lemma 6.6. *Let Γ be a valued constraint language over a domain D, $\mathcal{I}$ — an instance of $\mathrm{VCSP}(\Gamma)$, and $f \in \mathrm{Pol}^{(1)}(\Gamma)^+$. If s is a feasible assignment for $\mathcal{I}$ that minimises the cost, then $f(s)$ has the same cost.*

Proof. Let f and $\mathcal{I}$ be like in the statement of the lemma. If $f \neq \mathrm{id}$, then there exists a weighted polymorphism ω with $\omega(f) > 0$. Let $\mathcal{C} = \{(\sigma_1, \varrho_1), \dots, (\sigma_n, \varrho_n)\}$ be the set of constraints in $\mathcal{I}$. For each $i \in \{1, \dots, n\}$ the weighted relation ϱ_i is defined for the tuple $s(\sigma_i)$. Therefore, by the definition of a weighted polymorphism, for every i the following inequality is satisfied:

$$\sum_{g \in \mathrm{Pol}^{(1)}(\Gamma)} \omega(g) \cdot \varrho_i(g(s(\sigma_i))) \leq 0.$$

Since for each $g \in \mathrm{Pol}^{(1)}(\Gamma)$, we have $Cost_{\mathcal{I}}(g(s)) = \varrho_1(g(s(\sigma_1)))+\cdots+\varrho_n(s(g(\sigma_n)))$, adding the appropriate inequalities gives:

$$\sum_{g \in \mathrm{Pol}^{(1)}(\Gamma)} \omega(g) \cdot Cost_{\mathcal{I}}(g(s)) \leq 0.$$

Without loss of generality assume that $\omega(\mathrm{id}) = -1$. Then $\sum_{g \in \mathrm{supp}(\omega)} \omega(g) = 1$ and the inequality above can be written as

$$\sum_{g \in \mathrm{supp}(\omega)} \omega(g) \cdot Cost_{\mathcal{I}}(g(s)) + \omega(\mathrm{id}) \cdot Cost_{\mathcal{I}}(s) \leq 0, \text{ hence}$$

$$Cost_{\mathcal{I}}(s) \geq \sum_{g \in \mathrm{supp}(\omega)} \omega(g) \cdot Cost_{\mathcal{I}}(g(s)) \geq \sum_{g \in \mathrm{supp}(\omega)} \omega(g) \cdot Cost_{\mathcal{I}}(s) = Cost_{\mathcal{I}}(s).$$

Therefore $Cost_{\mathcal{I}}(g(s)) = Cost_{\mathcal{I}}(s)$ for each operation $g \in \mathrm{supp}(\omega)$, which finishes the proof. □

Proof. (of Proposition 6.5) Let Γ be a valued constraint language over a domain D. Suppose that there is a unary polymorphism $f \in \mathrm{Pol}(\Gamma)^+$ that is not bijective. Let $\Gamma' = \Gamma[f(D)]$. There is a natural correspondence between the instances of VCSP(Γ') and the instances of VCSP(Γ), induced by the correspondence between relations in Γ and their restrictions in Γ'. For any instance $\mathcal{I}'$ of VCSP(Γ') the corresponding instance $\mathcal{I}$ of VCSP(Γ) has the same variables. The weighted relation ϱ' in each constraint is replaced by any relation ϱ from Γ, which is equal to ϱ' when restricted to $f(D)$.

A feasible assignment for $\mathcal{I}'$ is also a feasible assignment for $\mathcal{I}$. On the other hand, if s is a feasible assignment for $\mathcal{I}$ then $f(s)$ is a feasible assignment for $\mathcal{I}'$. Hence, $\mathrm{Opt}_\Gamma(\mathcal{I})$ is defined if and only if $\mathrm{Opt}_{\Gamma'}(\mathcal{I}')$ is defined.

It is easy to see, that $\mathrm{Opt}_\Gamma(\mathcal{I}) \leq \mathrm{Opt}_{\Gamma'}(\mathcal{I}')$. Futhermore, by Lemma 6.6 for each s that is an optimal assignment for $\mathcal{I}$, we have

$$Cost_{\mathcal{I}}(s) = Cost_{\mathcal{I}}(f(s)) = Cost_{\mathcal{I}'}(f(s)).$$

Therefore, $\mathrm{Opt}_\Gamma(\mathcal{I}) \geq \mathrm{Opt}_{\Gamma'}(\mathcal{I}')$, and hence $\mathrm{Opt}_\Gamma(\mathcal{I}) = \mathrm{Opt}_{\Gamma'}(\mathcal{I}')$.

It follows that VCSP(Γ) is tractable if and only if VCSP(Γ') is tractable, and it is NP-hard if and only if VCSP(Γ') is NP-hard. Moreover, the valued constraint language Γ' is defined over a smaller domain. We replace Γ with Γ' and repeat this procedure, untill we obtain a language Γ' that is a core. □

The valued constraint language Γ' defined in the proof above is a *core* of Γ.

From now on, without loss of generality, let us assume that Γ is a core valued constraint language. For such a language we characterise the set of unary weighted polymorphisms.

Proposition 6.7. *Let Γ be a core valued constraint language. A unary weighting ω is a weighted polymorphism of Γ if and only if it assigns positive weights only to* *such bijective operations $f \in \mathrm{Pol}^{(1)}(\Gamma)$ that, for all weighted relations $\varrho \in \Gamma$, satisfy $\varrho \circ f = \varrho$.*

Proof. If a valid unary weighting ω assigns positive weights only to such operations $f \in \mathrm{Pol}^{(1)}(\Gamma)$ that, for all weighted relations $\varrho \in \Gamma$, satisfy $\varrho \circ f = \varrho$, then for each $\varrho \in \Gamma$ and a tuple $\mathbf{x} \in D^r$ for which ϱ is defined

$$\sum_{f \in \mathrm{Pol}^{(1)}(\Gamma)} \omega(f) \cdot \varrho(f(\mathbf{x})) = \sum_{f \in \mathrm{Pol}^{(1)}(\Gamma)} \omega(f) \cdot \varrho(\mathbf{x}) = 0,$$

and ω is clearly a weighted polymorphism of Γ.

For the other direction, let ω be a unary weighted polymorphism of Γ, such that $\mathrm{supp}(\omega) \neq \emptyset$. Without loss of generality assume that $\omega(\mathrm{id}) = -1$. Since Γ is a core language, the operations $g \in \mathrm{supp}(\omega)$ are bijective. Consider $\varrho \in \Gamma$ and a tuple $\mathbf{x} \in D^r$ for which ϱ is defined, and takes the minimal value, we have

$$\sum_{g \in \mathrm{supp}(\omega)} \omega(g) \cdot \varrho(g(\mathbf{x})) + \omega(\mathrm{id}) \cdot \varrho(\mathbf{x}) \leq 0, \text{ hence}$$

$$\varrho(\mathbf{x}) \geq \sum_{g \in \mathrm{supp}(\omega)} \omega(g) \cdot \varrho(g(\mathbf{x})) \geq \sum_{g \in \mathrm{supp}(\omega)} \omega(g) \cdot \varrho(\mathbf{x}) = \varrho(\mathbf{x}).$$

Therefore $\varrho(g(\mathbf{x})) = \varrho(\mathbf{x})$ for each $g \in \mathrm{supp}(\omega)$, which means that the operations in the support preserve the minimal weight.

Note that, since each $g \in \mathrm{supp}(\omega)$ is a bijective unary polymorphism, it determines a bijection G of the set of tuples for which ϱ is defined. We have shown that the bijection preserve the set of tuples with minimal weight. It can be similarly shown by induction that it preserves the set of tuples with any other fixed weight. Hence, we have proved that $\varrho \circ g = \varrho$ for all $g \in \mathrm{supp}(\omega)$. □

The result implies, that if we consider all unary polymorphisms of a given core language Γ and restrict our attention to those of them that are bijective and preserve all weighted relations in Γ, then those are precisely the unary polymorphisms that belong to $\mathrm{Pol}(\Gamma)^+$.

Let Γ be a core valued constraint language over a domain $D = \{d_1, \ldots, d_n\}$. We will define an n-ary weighted relation using variables $x_1, \ldots, x_n$. Let S consists of all pairs $(\varrho, (x_{i_1}, \ldots, x_{i_r}))$, such that $\varrho \in \Gamma$ is defined for the tuple $(d_{i_1}, \ldots, d_{i_r})$. Let H be a weighted relation on D defined by[2]

$$H(x_1, \ldots, x_n) = \sum_{(\varrho, \mathbf{x}) \in S} \varrho(\mathbf{x}).$$

[2] We define H so as to obtain an exact counterpart of a similar notion from the classical CSP – a relation that contains exactly those tuples $(x_1, \ldots, x_n)$ for which the operation defined by $d_i \mapsto x_i$ is a polymorphism [8].

Example 6.8. Consider a two element domain $D = (d_1, d_2)$ and a valued constraint language $\Gamma = \{\varrho_1, \varrho_2\}$, where the weighted relations ϱ_1, ϱ_2 are both binary. Suppose that ϱ_1 is defined for tuples (d_1, d_1) and (d_1, d_2), while ϱ_2 is defined only for (d_2, d_1). Then $H(x_1, x_2) = \varrho_1(x_1, x_1) + \varrho_1(x_1, x_2) + \varrho_2(x_2, x_1)$.

Note that the weighted relation H belongs to the weighted relational clone generated by Γ and it is defined for a tuple $(x_1, \ldots, x_n)$ if and only if the unary operation $f_{(x_1,\ldots,x_n)}$ defined by $d_i \mapsto x_i$ belongs to $\mathrm{Pol}^{(1)}(\Gamma)$. Moreover, since Γ is a core, H takes the same value for all tuples $(x_1, \ldots, x_n)$ for which $f_{(x_1,\ldots,x_n)} \in \mathrm{Pol}(\Gamma)^+$.

The next result says that by a slight modification of the weighted relation H we obtain a weighted relation that precisely distinguishes the operations in the positive clone from all the other unary polymorphisms. To prove it we will need a following technical lemma, which is a variant of the well known Farkas' Lemma used in linear programming:

Lemma 6.9. (Farkas). *Let S and T be finite sets of indices, where T is a disjoint union of two subsets, $T_\geq$ and $T_=$. For all $i \in S$, and all $j \in T$, let $a_{i,j}$ and b_j be rational numbers. Exactly one of the following holds:*

- *Either there exists a set of non-negative rational numbers $\{z_i \mid i \in S\}$ and a rational number C such that*

$$\textit{for each } j \in R_\geq, \quad \sum_{i \in S} a_{i,j} z_i \geq b_j + C,$$

$$\textit{for each } j \in R_=, \quad \sum_{i \in S} a_{i,j} z_i = b_j + C.$$

- *Or else there exists a set of integers $\{y_j \mid j \in T\}$ such that $\sum_{j \in T} y_j = 0$ and:*

$$\textit{for each } j \in T_\geq, \quad y_j \geq 0,$$

$$\textit{for each } i \in S, \quad \sum_{j \in T} y_j a_{i,j} \leq 0,$$

$$\textit{and } \sum_{j \in T} y_j b_j > 0.$$

The set $\{y_j \mid j \in T\}$ defined in the lemma is called a *certificate of unsolvability.*

Proposition 6.10. *Let Γ be a core valued constraint language over an n-element domain $D = \{d_1, \ldots, d_n\}$. There exist an n-ary weighted relation $N \in \mathrm{wRelClo}(\Gamma)$, and positive rational numbers $P < Q$, such that the following conditions are satisfied:*

- *$N(x_1, \ldots, x_n) = P$ if and only if a unary operation g defined by $d_i \mapsto x_i$ belongs to $\mathrm{Pol}(\Gamma)^+$,*
- *$N(x_1, \ldots, x_n) > Q$ if and only if a unary operation h defined by $d_i \mapsto x_i$ belongs to $\mathrm{Pol}(\Gamma) \setminus \mathrm{Pol}(\Gamma)^+$,*
- *otherwise $N(x_1, \ldots, x_n)$ is undefined.*

Proof. The weighted relation N will be given by a sum of all weighted relations in Γ with positive coefficients that we define later on.

Recall that S is a set of pairs $(\varrho, (x_{i_1}, \ldots, x_{i_r}))$, such that $\varrho \in \Gamma$ is defined for the tuple $(d_{i_1}, \ldots, d_{i_r})$. Then let

$$N'(x_1, \ldots, x_n) = \sum_{(\varrho, \mathbf{x}) \in S} \varrho(\mathbf{x}) \cdot z_{(\varrho, \mathbf{x})}$$

and for each unary polymorphism f of Γ let $N'(f)$ denote the formula $N'(f(d_1), \ldots, f(d_n))$, with the set of variables $\{z_{(\varrho,\mathbf{x})} \mid (\varrho, \mathbf{x}) \in S\}$ and rational coefficients

$$\varrho(f(d_{i_1}), \ldots, f(d_{i_r})), \text{ where } (\varrho, (x_{i_1}, \ldots, x_{i_r})) \in S.$$

Consider a system of linear inequalities and equations:

$$N'(h) \geq 1 + C, \text{ for each unary } h \in \mathrm{Pol}(\Gamma) \setminus \mathrm{Pol}(\Gamma)^+,$$
$$N'(g) = C, \text{ for each unary } g \in \mathrm{Pol}(\Gamma)^+.$$

Note that the sets S and $\mathrm{Pol}^{(1)}(\Gamma)$ correspond to the sets of indices from Farkas' Lemma S and T, respectively. Therefore, by Lemma 6.9 there are two mutually exclusive possibilities.

Either there exist a set of non-negative rational numbers $\{z_{(\varrho,\mathbf{x})} \mid (\varrho, \mathbf{x}) \in S\}$ and a rational number C, such that this system is satisfied. Then a weighted relation N' with those coefficients almost satisfies our requirements. If C is not positive, we set

$$N(x_1, \ldots, x_n) = N'(x_1, \ldots, x_n) + b,$$

where b is a suitable positive rational.

Or else there exists a certificate of unsolvability $\{y_f \mid f \in \mathrm{Pol}^{(1)}(\Gamma)\}$. Then let us consider a weighting defined by $\omega(f) = y_f$. If ω is valid, then it is a unary weighted polymorphism of Γ. Moreover, ω assigns to all unary operations in $\mathrm{Pol}(\Gamma) \setminus \mathrm{Pol}(\Gamma)^+$ non-negative weights that sum up to a positive number. Hence for some unary $h \in \mathrm{Pol}(\Gamma) \setminus \mathrm{Pol}(\Gamma)^+$, we have $\omega(h) > 0$. Which contradicts $h \notin \mathrm{Pol}(\Gamma)^+$.

It might happen that $y_g < 0$ for some unary operation $g \in \mathrm{Pol}(\Gamma)^+$ that is not the identity. But then there exists a unary weighted polymorphism of Γ which assigns a positive weight to g. By scaling it and adding to ω, we obtain the weighted polymorphism needed for the contradiction. □

Since $N \in \mathrm{wRelClo}(\Gamma)$ the valued constraint language $\Gamma \cup \{N\}$ is tractable if and only if Γ is tractable, and it is NP-hard if and only if Γ is NP-hard.

6.3. **Rigid cores.** Let Γ be a core valued constraint language over an n-element domain $D = \{d_1, \ldots, d_n\}$. We will use the weighted relation N to futher reduce the class of cores that we need to consider. For each $i \in \{1, \ldots, n\}$, let

$$N_i(x) = \begin{cases} N(d_1, \ldots, d_n) & \text{if } x = d_i, \\ \mathit{undefined} & \text{otherwise.} \end{cases}$$

Let Γ_c denote the valued constraint language obtainted from Γ by adding the weighted relation N and all weighted relations N_i.

Note that the only unary polymorphism of Γ_c is the identity, which also means that there is only one unary weighted polymorphism of Γ_c - the zero-valued polymorphism.

Definition 6.11. A valued constraint language Γ is a *rigid core* if there is exactly one unary polymorphism of Γ, which is the identity.

We will now prove a result which, together with the Proposition 6.5, implies that for each valued constraint language Γ, there is an equivalent language that is a rigid core.

Proposition 6.12. *For each core valued constraint language* Γ *there exists a language* Γ' *which is a rigid core, such that the valued constraint language* Γ *is tractable if and only if* Γ' *is tractable, and it is NP-hard if and only if* Γ' *is NP-hard.*

Proof. Let Γ be a core valued constraint language over a domain $D = \{d_1, \dots, d_n\}$. Assume without loss of generality that $N \in \Gamma$ and let P and Q be the positive constants in the definition of the weighted relation N. We will show a polynomial-time Turing reduction from VCSP(Γ_c) to VCSP(Γ).

Let $\mathcal{I}_c = (V_c, D, \mathcal{C}_c)$ be an instance of VCSP(Γ_c). The set of variables V in the new instance $\mathcal{I}$ is a disjoint union of V_c and $\{v_1, \dots, v_n\}$.

Replace each $((v), N_i) \in \mathcal{C}_c$ with $((v_1, \dots, v_{i-1}, v, v_{i+1}, \dots v_n), N)$, obtaining a new set of constraints $\mathcal{C}_1$, where all weighted relations are already from Γ.

Let C be the sum of weights that all weighted relations in all constraints in $\mathcal{C}_1$ assign to all tuples for which they are defined. The final set of constraints $\mathcal{C}$ additionally contains m constraints of the form $((v_1, \dots, v_n), N)$, where m is big enough to ensure that $m \cdot (Q - P) > C$.

Each optimal assignment s_c for $\mathcal{I}_c$ gives rise to an optimal assignment s for $\mathcal{I}$. It coincides with s_c on V_c and for each $i \in \{1, \dots, n\}$, we set $s(v_i) = d_i$.

In the other direction, we will show that the optimal assignments for $\mathcal{I}_c$ correspond to those optimal assignments s for $\mathcal{I}$ for which $N(s(v_1, \dots, v_n)) = P$. Take such an assignment s. The tuple $s(v_1, \dots, v_n)$ determines a unary operation g, defined by $d_i \mapsto s(v_i)$. The operation g, by the definition of the weighted relation N, belongs to the positive clone $\mathrm{Pol}(\Gamma)^+$. Hence, g^{-1} also belongs to the positive clone. Since Γ is a core, the assignment $g^{-1}(s)$ is optimal for $\mathcal{I}$. Its restriction onto V_c is an optimal assignment for $\mathcal{I}_c$.

Now suppose there is an optimal assignment s' for $\mathcal{I}$ with $N(s'(v_1, \dots, v_n)) > Q$. While there are m constraints of the form $((v_1, \dots, v_n), N)$, we have

$$Cost_{\mathcal{I}}(s') \geq m \cdot Q > m \cdot P + C.$$

But if there was any feasible assignment s_c for $\mathcal{I}_c$, the corresponding assignment s for $\mathcal{I}$ would satisfy $Cost_{\mathcal{I}}(s) < m \cdot P + C$, which gives us a contradiction, and implies that there are no feasible assignments for $\mathcal{I}_c$. □

Without loss of generality we can consider only rigid cores. Let us characterise such valued languages more precisely.

Definition 6.13. An operation f is *idempotent* if $f(x, \ldots, x) = x$ and a clone of operations is *idempotent* if all its elements are.

Clearly, if all polymorphisms of a valued constraint language Γ are idempotent, then the language is a rigid core. We will show that the converse statement is also true, and hence obtain the following:

Proposition 6.14. *A valued constraint language* Γ *is a rigid core if and only if the clone* $\mathrm{Pol}(\Gamma)$ *is idempotent.*

Proof. If g is a k-ary polymorphism of Γ, then the unary operation f defined by $f(x) = g(x, \ldots, x)$ is also a polymorphism of Γ. Since Γ is a rigid core, $f = \mathrm{id}$. Therefore $g(x, \ldots, x) = x$. □

7. Weighted varieties

In this section we introduce a notion of a weighted variety, which is a variety provided with a richer structure.

Definition 7.1. A k-ary *weighting* of an algebra $\mathbf{A}$ is a function that assigns rational weights to all k-ary basic operations of $\mathbf{A}$ in such a way, that the sum of all weights is 0. An algebra $\mathbf{A}$ together with the set of weightings is called a *weighted algebra.*

Note that in the definition above we omit the condition which says, that a weighting may assign negative weights only to projections.

Definition 7.2. A k-ary *weighting* of a signature Σ is a function that assigns rational weights to all k-ary symbols in Σ in such a way, that the sum of all weights is 0. A set of weightings of a given signature Σ is called a *weighted signature* and denoted $\mathrm{w}\Sigma$.

Consider an algebra $\mathbf{A}$ over a signature Σ and a set of weightings $\mathrm{w}\Sigma$ of Σ. This set of weightings induces a structure of a weighted algebra on $\mathbf{A}$. For any k-ary weighting $\omega \in \mathrm{w}\Sigma$ we define a k-ary weighting $\omega^{\mathbf{A}}$ of $\mathbf{A}$ by

$$\omega^{\mathbf{A}}(g) = \sum_{\{f \in \Sigma \colon f^{\mathbf{A}} = g\}} \omega(f)$$

i.e. the weight of the k-ary basic operation g of $\mathbf{A}$ is the sum the weights that ω assigns to those symbols in Σ that correspond to the operation g.

Definition 7.3. A *weighted variety* $\mathcal{W}$ is a variety $\mathcal{V}$ of algebras over a fixed signature Σ together with a set of weightings of this signature $\mathrm{w}\Sigma$.

Clearly, for each algebra $\mathbf{A}$ in the weighted variety $\mathcal{W}$ there is a corresponding weighted algebra, with the set of weightings induced by $\mathrm{w}\Sigma$. For convenience, we will say that a weighted algebra $\mathbf{A}$ belongs to a weighted variety $\mathcal{W}$.

Consider a weighted algebra $\mathbf{A}$. Take Σ to be a signature of $\mathbf{A}$, such that each symbol in Σ corresponds to a different basic operation of $\mathbf{A}$. Every weighting of $\mathbf{A}$ determines a weighting of the signature Σ. If the weighted signature $\mathrm{w}\Sigma$ consist of all those weightings, then we denote by $\mathcal{W}(\mathbf{A})$ the weighted variety $(\mathcal{V}(\mathbf{A}), \mathrm{w}\Sigma)$.

Definition 7.4. If the set of basic operations of a weighted algebra $\mathbf{A}$ is a clone of operations and each weighting of $\mathbf{A}$ is a valid weighting of that clone, then we call $\mathbf{A}$ a *weighted c-algebra.*

For a weighted c-algebra $\mathbf{A}$, let $\mathrm{Imp}(\mathbf{A})$ denote the set of weighted relations on A, which are improved by all weightings in $\mathbf{A}$.

Let $\mathbf{A}$ be a weighted c-algebra. Consider a finite weighted algebra $\mathbf{B} \in \mathcal{W}(\mathbf{A})$. Clearly, the set of basic operations of $\mathbf{B}$ is a clone of operations. Moreover, any weighting that $\mathrm{w}\Sigma$ induces on $\mathbf{B}$ assigns negative weights only to projections on B (see Example 5.13). Therefore we obtain the following:

Proposition 7.5. *If $\boldsymbol{A}$ is a weighted c-algebra, then every $\boldsymbol{B} \in \mathcal{W}(\boldsymbol{A})$ is also a weighted c-algebra.*

Take Γ to be a valued constraint language over a domain D and consider the c-algebra $\mathbf{A} = (D, \mathrm{Pol}(\Gamma))$ together with a set of weightings $\mathrm{wPol}(\Gamma)$. We will prove that for each finite weighted c-algebra $\mathbf{B} \in \mathcal{W}(\mathbf{A})$ the valued constraint language $\mathrm{Imp}(\mathbf{B})$ is not harder then Γ. The proof consists of a sequence of lemmas.

Lemma 7.6. *Let $\boldsymbol{A}$ be a finite weighted c-algebra. For any finite $\boldsymbol{B} \in P_{fin}(\boldsymbol{A})$, we have that* $\mathrm{VCSP}(\mathrm{Imp}(\boldsymbol{B}))$ *polynomial-time reduces to* $\mathrm{VCSP}(\mathrm{Imp}(\boldsymbol{A}))$.

Proof. Let A^n be the universe of $\mathbf{B}$ and let Γ be a finite subset of $\mathrm{Imp}(\mathbf{B})$. Take $\varrho \in \Gamma$ to be a r-ary weighted relation i.e. ϱ is a partial function from $(A^n)^r$ to $\mathbb{Q}$. There is a natural way of defining a corresponding weighted relation of arity $n \cdot r$ on the set A. We denote this weighted relation by ϱ'.

Let $\omega^{\mathbf{A}}$ be a k-ary weighting from $\mathbf{A}$. Since all weightings in the weighted c-algebras from $\mathcal{W}(\mathbf{A})$ are induced by the fixed set of weightings of their signature, there is a corresponding k-ary weighting $\omega^{\mathbf{B}}$ in $\mathbf{B}$, which is a weighted polymorphism of ϱ. It is not hard to show that $\omega^{\mathbf{A}}$ is a weighted polymorphism of ϱ', as the basic operations of $\mathbf{B}$ are the operations of $\mathbf{A}$ computed coordinatewise. Hence, each weighting from $\mathbf{A}$ is a weighted polymorphism of ϱ', which means that $\varrho' \in \mathrm{Imp}(\mathbf{A})$.

For each $\varrho \in \Gamma$ we have defined a corresponding $\varrho' \in \mathrm{Imp}(\mathbf{A})$. Let $\Gamma' \subseteq \mathrm{Imp}(\mathbf{A})$ be the (finite) set of all those weighted relations.

Now take an arbitrary instance $\mathcal{I} = (V, A^n, \mathcal{C})$ of $\mathrm{VCSP}(\Gamma)$. Replace the domain A^n by A, and each variable $v_i \in V$ by a set of n variables $\{v_i^1, \ldots, v_i^n\}$, obtaining a new set of variables V'. In each constraint $(\sigma, \varrho) \in \mathcal{C}$, where ϱ is an r-ary weighted relation, replace the r-tuple σ of variables from V by the corresponding nr-tuple of variables from V', and the relation ϱ by a corresponding relation ϱ' from Γ'. The new instance $\mathcal{I}' = (V', A, \mathcal{C}')$ is an instance of $\mathrm{VCSP}(\Gamma')$. It is easy to see that there is a one-to-one correspondence between the optimal assignments for $\mathcal{I}$ and the optimal assignments for $\mathcal{I}'$. □

Lemma 7.7. *Let* $\boldsymbol{A}$ *be a finite weighted c-algebra. For any finite* $\boldsymbol{B} \in S(\boldsymbol{A})$*, we have that* $\mathrm{VCSP}(\mathrm{Imp}(\boldsymbol{B}))$ *polynomial-time reduces to* $\mathrm{VCSP}(\mathrm{Imp}(\boldsymbol{A}))$*.*

Proof. Notice that $\mathrm{Imp}(\mathbf{B}) \subseteq \mathrm{Imp}(\mathbf{A})$, so there is nothing to be proved. □

Lemma 7.8. *Let* $\boldsymbol{A}$ *be a finite weighted c-algebra. For any finite* $\boldsymbol{B} \in H(\boldsymbol{A})$*, we have that* $\mathrm{VCSP}(\mathrm{Imp}(\boldsymbol{B}))$ *polynomial-time reduces to* $\mathrm{VCSP}(\mathrm{Imp}(\boldsymbol{A}))$*.*

Proof. By the isomorphism theorem we can consider $\mathbf{B}$ to be a quotient algebra $\mathbf{A}/\sim$ rather than a homomorphic image of $\mathbf{A}$.

Let $A/\sim$ be the universe of $\mathbf{B}$ and let Γ be a finite subset of $\mathrm{Imp}(\mathbf{B})$. Take $\varrho \in \Gamma$ to be a r-ary weighted relation i.e. ϱ is a partial function from $(A/\sim)^r$ to $\mathbb{Q}$. We define a corresponding weighted relation ϱ' of arity r on the set A by $\varrho'(x_1, \ldots, x_r) = \varrho([x_1]_\sim, \ldots [x_r]_\sim)$.

Let $\omega^{\mathbf{A}}$ be a k-ary weighting from $\mathbf{A}$. There is a corresponding k-ary weighting $\omega^{\mathbf{B}}$ in $\mathbf{B}$, which is a weighted polymorphism of ϱ. It is not hard to show that $\omega^{\mathbf{A}}$ is a weighted polymorphism of ϱ'. Hence, each weighting from $\mathbf{A}$ is a weighted polymorphism of ϱ', which means that $\varrho' \in \mathrm{Imp}(\mathbf{A})$.

For each $\varrho \in \Gamma$ we have defined a corresponding $\varrho' \in \mathrm{Imp}(\mathbf{A})$. Let $\Gamma' \subseteq \mathrm{Imp}(\mathbf{A})$ be the (finite) set of all those weighted relations.

Now take an arbitrary instance $\mathcal{I} = (V, A/\sim, \mathcal{C})$ of $\mathrm{VCSP}(\Gamma)$. Replace the domain $A/\sim$ by A. In each constraint $(\sigma, \varrho) \in \mathcal{C}$ replace the relation ϱ by a corresponding relation ϱ' from Γ'. The new instance $\mathcal{I}' = (V, A, \mathcal{C}')$ is an instance of $\mathrm{VCSP}(\Gamma')$.

If $s' \colon V \to A$ is an optimal assignment for $\mathcal{I}'$, then $s \colon V \to A/\sim$ defined by $s(v) = [s'(v)]_\sim$ is an optimal assignment for $\mathcal{I}$. On the other hand, if $s \colon V \to A/\sim$ is an optimal assignment for $\mathcal{I}$, then any assignment $s' \colon V \to A$, such that for each $v \in V$, we have $s'(v) \in s(v)$, is optimal for $\mathcal{I}'$. □

The above lemmas together with the Proposition 5.8, give us the following:

Proposition 7.9. *For any finite weighted c-algebra* $\boldsymbol{A}$*, and any finite* $\boldsymbol{B} \in \mathcal{W}(\boldsymbol{A})$*, we have that* $\mathrm{VCSP}(\mathrm{Imp}(\boldsymbol{B}))$ *polynomial-time reduces to* $\mathrm{VCSP}(\mathrm{Imp}(\boldsymbol{A}))$*.*

Definition 7.10. A weighted c-algebra $\mathbf{A}$ is *idempotent* if all basic operations of $\mathbf{A}$ are idempotent. A weighted variety $\mathcal{W}$ is said to be *idempotent* if all weighted c-algebras that belong to $\mathcal{W}$ are idempotent.

Proposition 7.11. *A weighted variety* $\mathcal{W}(\boldsymbol{A})$ *generated by an idempotent weighted c-algebra* $\boldsymbol{A}$ *is idempotent.*

Proof. Let Σ be the signature of $\mathbf{A}$. Take any weighted c-algebra $\mathbf{B}$ that belongs to $\mathcal{W}(\mathbf{A})$ and a basic operation g of $\mathbf{B}$. There is a symbol $f \in \Sigma$ for which $f^{\mathbf{B}} = g$. Since the weighted c-algebra $\mathbf{A}$ is idempotent, it satisfies an identity $f(x, \ldots, x) \approx x$. This means that $\mathbf{B}$ also satisfies this identity. Hence, the operation $f^{\mathbf{B}} = g$ is idempotent. □

Take a valued constraint language Γ over the domain D to be a rigid core. Consider the c-algebra $\mathbf{A} = (D, \mathrm{Pol}(\Gamma))$ together with the set of weightings $\mathrm{wPol}(\Gamma)$. This weighted c-algebra is idempotent. Hence, the weighted variety generated by it is idempotent.

References

[1] G. Birkhoff, On the structure of abstract algebras, *Proc. Camb. Philos. Soc. 31*, pages 433-454, 1935.

[2] A. Bulatov and P. Jeavons, Algebraic structures in combinatorial problems, *Technical Report MATH-AL-4-2001*, Technische Universitaät Dresden, Dresden, Germany, 2001.

[3] A. Bulatov, P. Jeavons and A. Krokhin, Classifying the complexity of constraints using finite algebras, *SIAM Journal of Computing*, 34(3), pages 720-742, 2005.

[4] A. Bulatov, A. Krokhin and P. Jeavons, Constraint satisfaction problems and finite algebras, *Automata, languages and programming (Geneva, 2000)*, volume 1853 of *Lecture Notes in Comput. Sci.*, pages 272-282, Springer, Berlin, 2000.

[5] D. Cohen, M. Cooper, P. Creed, P. Jeavons and S. Živný, An algebraic theory of complexity for discrete optimisation, *CoRR*, Technical Report abs/1207.6692v1, 2012.

[6] D. Cohen, M. Cooper and P. Jeavons, An algebraic characterisation of complexity for valued constraints, *Proceedings of CP'06*, volume 4204 of *LNCS*, pages 107-121, 2006.

[7] D. Cohen, M. Cooper, P. Jeavons and A. Krokhin, The complexity of soft constraint satisfaction, *Artificial Intelligence*, 170(11), pages 983-1016, 2006.

[8] V. Dalmau and A. Krokhin, Robust satisfiability for CSPs: hardness and algorithmic results, *ACM Transactions on Computation Theory*, 2013, to appear.

[9] T. Feder and M. Vardi, The computational structure of monotone monadic snp and constraint satisfaction: A study through datalog and group theory, *SIAM Journal on Computing*, 28(1), pages 57-104, 1998.

[10] A. Huber, A. Krokhin and R. Powell, Skew bisubmodularity and Valued CSPs, *Proceedings of SODA'13*, pages 1296-1305, 2013.

[11] P. Jeavons, D. Cohen and M. Gyssens, Closure properties of constraints. *Journal of the ACM*, 44(4), pages 527-548, 1997.

[12] V. Kolmogorov and S. Živný, The complexity of conservative valued CSPs, *Proceedings of SODA'12*, pages 750-759, 2012.

[13] A. Tarski, A remark on functionally free algebras, *Ann. of Math.*, 47, pages 163-165, 1946.

[14] J. Thapper and S. Živný, The complexity of finite-valued CSPs, *Proceedings of STOC'13*, pages 695-704, 2013.

Institute of Mathematics of the Polish Academy of Sciences, Warsaw, Poland
E-mail address: joanna.ochremiak@gmail.com