

Constraint satisfaction parameterized by solution size

Andrei A. Bulatov^{1*} and Dániel Marx^{2**}

¹ Simon Fraser University

abulatov@cs.sfu.ca

² Humboldt-Universität zu Berlin

dmarx@cs.bme.hu

Abstract. In the constraint satisfaction problem (CSP) corresponding to a constraint language (i.e., a set of relations) Γ , the goal is to find an assignment of values to variables so that a given set of constraints specified by relations from Γ is satisfied. In this paper we study the fixed-parameter tractability of constraint satisfaction problems parameterized by the size of the solution in the following sense: one of the possible values, say 0, is “free,” and the number of variables allowed to take other, “expensive,” values is restricted. A size constraint requires that exactly k variables take nonzero values. We also study a more refined version of this restriction: a global cardinality constraint prescribes how many variables have to be assigned each particular value. We study the parameterized complexity of these types of CSPs where the parameter is the required number k of nonzero variables. As special cases, we can obtain natural and well-studied parameterized problems such as INDEPENDENT SET, VERTEX COVER, d -HITTING SET, BICLIQUE, etc. In the case of constraint languages closed under substitution of constants, we give a complete characterization of the fixed-parameter tractable cases of CSPs with size constraints, and we show that all the remaining problems are W[1]-hard. For CSPs with cardinality constraints, we obtain a similar classification, but for some of the problems we are only able to show that they are BICLIQUE-hard. The exact parameterized complexity of the BICLIQUE problem is a notorious open problem, although it is believed to be W[1]-hard.

1 Introduction

In a constraint satisfaction problem (CSP) we are given a set of variables, and the goal is to find an assignment of the variables subject to specified constraints. A constraint is usually expressed as a requirement that combinations of values of a certain (usually small) set of variables belong to a certain relation. In the theoretical study of CSPs, one of the key research directions has been the complexity of the CSP when there are restrictions on the type of allowed relations [9,3,2]. This research direction has been started by the seminal Schaefer’s Dichotomy Theorem [17], which showed that every Boolean CSP (i.e., CSP with 0-1 variables) restricted in this way is either solvable in polynomial time or is NP-complete. An outstanding open question is the so called *Dichotomy conjecture* of Feder and Vardi [7] which suggests that the dichotomy remains true for

* Research supported by an NSERC Discovery grant.

** Research supported by the Alexander von Humboldt Foundation and OTKA grant 67651.

CSPs over any fixed finite domain. The significance of a dichotomy result is that it is very likely to provide a comprehensive understanding of the algorithmic nature of the problem. Indeed, in order to obtain the tractability part of such a conjecture one needs to identify all the algorithmic ideas relevant for the problem.

Parameterized complexity [6,8] investigates the complexity of problems in finer details than classical complexity. Instead of expressing the running time of an algorithm as a function of the input size n only, the running time is expressed as a function of n and a well-defined parameter k of the input instance (such as the size of the solution k we are looking for). For many problems and parameters, there is a polynomial-time algorithm for every fixed value of k , i.e., the problem can be solved in time $n^{f(k)}$. In this case, it makes sense to ask if the combinatorial explosion can be limited to the parameter k by improving the running time to $f(k) \cdot n^{O(1)}$. Problems having algorithms with running time of this form are called *fixed-parameter tractable (FPT)*; it turns out that many well-known NP-hard problems, such as k -VERTEX COVER, k -PATH, and k -DISJOINT TRIANGLES are FPT. On the other hand, the theory of W[1]-hardness suggests that certain problems (e.g., k -CLIQUE, k -DOMINATING SET) are unlikely to be FPT.

The canonical complete problems of the W-hierarchy are (circuit) satisfiability problems where the solution is required to contain exactly k ones. This leads us to the study of Boolean CSP problems with the goal of finding a solution with exactly k ones. The first attempt to study the parameterized complexity of Boolean CSP was made in [14]. If we consider 0 as a “cheap” value available in abundance, while 1 is “costly” and of limited supply then the natural parameter is the number of 1’s in a solution. Boolean CSP asking for a solution that assigns exactly k ones is known as the k -ONES problem [5,11]. Clearly, the problem is polynomial-time solvable for every fixed k (by brute force), but it is not at all obvious whether it is FPT. For example, it is possible to express k -VERTEX COVER (which is FPT) and k -INDEPENDENT SET (which is W[1]-hard) as a Boolean CSP. Therefore, characterizing the parameterized complexity of k -ONES requires understanding a class of problems that includes, among many other things, the most basic graph problems. It turned out that the parameterized complexity of the k -ONES problem depends on a new combinatorial property called *weak separability* [14]. Assuming that the constraints are restricted to a finite set Γ of Boolean relations, if every relation in Γ is weakly separable, then the problem is FPT; if Γ contains even one relation violating weak separability, then the problem is W[1]-hard.

There have been further parameterized complexity studies of Boolean CSP [12,18,13], but CSP’s with larger domains were not studied. In most cases, we expect that results for larger domains are much more complex than for the Boolean case, and usually require significant new ideas. The goal of the present paper is to generalize the results of [14] to non-Boolean domains. First, we have to define what the proper generalization of k -ONES is if the variables are not Boolean. One natural generalization assumes that there is a distinguished “cheap” value 0 and requires that in a solution there are exactly k nonzero variables. We will call this version of the CSP a *constraint satisfaction problem with size constraints* and denote by OCSF. Another generalization of k -ONES specifies the number $\pi(d)$ of variables assigned each nonzero value d : A mapping $\pi : D \setminus \{0\} \rightarrow \mathbb{N}$ is given, and it is required that for each nonzero value d , exactly $\pi(d)$ variables are assigned value d . In the CSP and AI literature, requirements of this

form are called *global cardinality constraints* [1,15]. We will call this problem the *constraint satisfaction problem with cardinality constraints* and denote it by CCSP. In both versions, the parameter is the number of nonzero values required, that is, k for OCSP, and $\sum_{d \in D \setminus \{0\}} \pi(d)$ for CCSP. The usual (non-parametrized) complexity of CCSP over arbitrary domain was characterized in [4]. We investigate both versions; as we shall see, there are unexpected differences between the complexity of the two variants.

A natural minor generalization of CSPs is allowing the use of constants in the input, that is, some variables in the input can be fixed to constant values, or equivalently the constant unary relation $\{(d)\}$ is allowed for every element d of the domain. It is known that the complexity of the decision CSP (corresponding to a ‘core’ structure) does not change with this generalization [3]. While there is no similar result for the versions of CSPs we study here (and thus this assumption may diminish the generality of our results), this setting is still quite general and at the same time more robust. Many technicalities can be avoided with this formulation. For example, the availability of constants ensures that the decision and search problems are equivalent: by repeatedly substituting constants and solving the decision problem, we can actually find a solution.

Is weak separability the right tractability criterion in the non-Boolean case? It is not difficult to observe that the algorithm of [14] using weak separability generalizes for non-Boolean problems³. However, it is not true that only weakly separable relations are tractable. It turns out that there are certain degeneracies and symmetries that allow us to solve the problem even for some relations that are not weakly separable. To understand these degenerate situations, the notion of multivalued morphisms (a generalization of homomorphisms) turns out to be crucial.

Results. For CSP with size constraints, we prove a dichotomy result:

Theorem 1.1. *For every finite Γ closed under substitution of constants, OCSP(Γ) is either FPT or W[1]-hard.*

The precise tractability criterion (which is quite technical) is stated in Section 4. The algorithmic part of Theorem 1.1 consists of a preprocessing to eliminate degeneracies and trivial cases, followed by the use of weak separability. In the hardness part, we take a relation R having a counterexample to weak separability, and use it to show that a known W[1]-hard problem can be simulated with this relation. In the Boolean case [14], this is fairly simple: by identifying coordinates and substituting 0’s, we can assume that the relation R is binary, and we need to prove hardness only for two concrete binary relations. For larger domains, however, this approach does not work. We cannot reduce the counterexample to a binary relation by identifying coordinates, thus a complex case analysis would be needed. Fortunately, our hardness proof is more uniform than that. We introduce gadgets that control the values that can appear on the variables. There are certain degenerate cases when these gadgets do not work. However, these degenerate cases can be conveniently described using multivalued morphisms, and these cases turn out to be exactly the cases that we can use in the algorithmic part of the proof.

In the case of cardinality constraints, we face an interesting obstacle. Consider the binary relation R containing only tuples $(0, 0)$, $(1, 0)$, and $(0, 2)$. Given a CSP instance with constraints of this form, finding a solution where the number of 1’s is exactly k

³ In fact, we give an algorithm for non-Boolean domains that is simpler than the one in [14].

and the number of 2's is exactly k is essentially equivalent to finding an independent set of a bipartite graph with k vertices in both classes, or equivalently, a complete bipartite graph (biclique) with $k + k$ vertices. The parameterized complexity of the k -BICLIQUE problem is a longstanding open question (it is conjectured to be W[1]-hard). Thus at this point, it is not possible to give a dichotomy result similar to Theorem 1.1 in the case of cardinality constraints, unless we prove that BICLIQUE is hard:

Theorem 1.2. *For every finite Γ closed under substitution of constants, $\text{CCSP}(\Gamma)$ is either FPT, or BICLIQUE-hard.*

2 Preliminaries

Constraint satisfaction problem with size and cardinality constraints. Let D be a set. We assume that D contains a distinguished element 0. Let D^n denote the set of all n -tuples of elements from D . An n -ary relation on D is a subset of D^n , and a constraint language Γ is a set of relations on D . In this paper constraint languages are assumed to be finite. We denote by $\text{dom}(\Gamma)$ the set of all values that appear in tuples of the relations in Γ . Given a constraint language Γ , an instance of the constraint satisfaction problem (CSP) is a pair $I = (V, \mathcal{C})$, where V is a set of variables, and \mathcal{C} is a set of constraints. A constraint is a pair $\langle \mathbf{s}, R \rangle$, where R is a (say, n -ary) relation from Γ , and \mathbf{s} is an n -tuple of variables. A satisfying assignment of I is a mapping $\tau : V \rightarrow D$ such that for every $\langle \mathbf{s}, R \rangle \in \mathcal{C}$ with $\mathbf{s} = (s_1 \dots, s_n)$ the image $\tau(\mathbf{s}) = (\tau(s_1), \dots, \tau(s_n))$ belongs to R . The question in the CSP is whether there is a satisfying assignment for a given instance. The CSP over constraint language Γ is denoted by $\text{CSP}(\Gamma)$.

The size of an assignment is the number of variables receiving nonzero values. A size constraint is a prescription on the size of the assignment. A cardinality constraint for a CSP instance I is a mapping $\pi : D \rightarrow \mathbb{N}$ with $\sum_{a \in D} \pi(a) = |V|$. A satisfying assignment τ of I satisfies the cardinality constraint π if the number of variables mapped to each $a \in D$ equals $\pi(a)$. We denote by $\text{CCSP}(\Gamma)$ the variant of $\text{CSP}(\Gamma)$ where the input contains both a cardinality constraint π and the size constraint $k = \sum_{a \in D \setminus \{0\}} \pi(a)$ (this constraint is used a parameter); the question is, given an instance I , an integer k , and a cardinality constraint π , whether there is a satisfying assignment of I of size k and satisfying π . So, instances of OCSP (resp., CCSP) are triples (V, \mathcal{C}, k) (resp., quadruples (V, \mathcal{C}, k, π)). A solution of an instance is a satisfying assignment satisfying the size/cardinality constraints. For both OCSP(Γ) and CCSP(Γ), we are interested in FPT with respect to the parameter k . The INDEPENDENT SET problem is representable as OCSP(R_{IS}), where $R_{IS} = \{(0, 0), (0, 1), (1, 0)\}$. Similarly, the BICLIQUE problem in which given a bipartite graph $G(A, B)$, find two $A' \subseteq A$ and $B' \subseteq B$ with $|A'| = |B'| = t$ and such that every vertex of A' is adjacent with every vertex of B' . This problem is equivalent to CCSP($\{R_{BC}\}$), where R_{BC} is a relation on $\{0, 1, 2\}$ given by $\{(0, 0), (1, 0), (0, 2)\}$.

Closures and 0-validity A constraint language is called constant closed (cc-, for short) if along with every (say, n -ary) relation R , any i , $1 \leq i \leq n$, and any $d \in D$ the relation obtained by substitution of constants $R^{i:d} = \{(a_1, \dots, a_{i-1}, a_{i+1}, \dots, a_n) \mid (a_1, \dots, a_{i-1}, d, a_{i+1}, \dots, a_n) \in R\}$, also belongs to R . Substitution of constants

d_1, \dots, d_q for coordinate positions i_1, \dots, i_q is defined in a similar way; the resulting relation is denoted by $R^{i_1, \dots, i_q; d_1, \dots, d_q}$. We call the smallest cc-language containing a constraint language Γ the *cc-closure* of Γ . It is easy to see that the cc-closure of Γ is the set of relations obtained from relations of Γ by substituting constants.

Let f be a satisfying assignment for an instance $I = (V, \mathcal{C}, k)$ of $\text{OCSP}(\Gamma)$ and $S = \{v \mid f(v) \neq 0\}$. We say that the instance $I' = (V', \mathcal{C}', k')$ is *obtained by substituting the nonzero values of f as constants* if I' is constructed as follows: $V' = V \setminus S$, and for each constraint $\langle s, R \rangle \in \mathcal{C}$ such that v_{i_1}, \dots, v_{i_r} are the variables from s contained in S and $\{v_{j_1}, \dots, v_{j_q}\} = s \setminus S$, we include in \mathcal{C}' the constraint $\langle (v_{j_1}, \dots, v_{j_q}), R^{i_1, \dots, i_r; f(v_{i_1}), \dots, f(v_{i_r})} \rangle$. The size constraint k' is set to k minus the size of f . This operation is defined similarly for a $\text{CCSP}(\Gamma)$ instance $I = (V, \mathcal{C}, k, \pi)$, but in this case the new cardinality constraint π' is given by $\pi'(d) = \pi(d) - |\{v \in V \mid f(v) = d\}|$.

A relation is said to be *0-valid* if the all-zero tuple belongs to the relation. A constraint language Γ is a *cc0-language* if every $R \in \Gamma$ is 0-valid, and if R' is a 0-valid relation obtained from R by substitution of constants, then $R' \in \Gamma$. Observe that if Γ is a cc-language and Γ_0 is the set of 0-valid relations in Γ , then Γ_0 is a cc0-language (but not necessarily a cc-language).

We say that tuple $\mathbf{t}_1 = (a_1, \dots, a_r)$ is an *extension* of tuple $\mathbf{t}_2 = (b_1, \dots, b_r)$ if they are of the same length and for every $1 \leq i \leq r$, $a_i = b_i$ if $b_i \neq 0$. Tuple \mathbf{t}_2 is then called a *subset* of \mathbf{t}_1 . A *minimal satisfying extension* of an assignment f is an extension f' of f (where f, f' are viewed as tuples) such that f' is satisfying, and f has no satisfying extension $f'' \neq f'$ such that f' is an extension of f'' .

By repeatedly branching on the unsatisfied constraints, a simple bounded search tree algorithm can enumerate all the minimal satisfying extensions of an assignment.

Lemma 2.1. *Let Γ be a finite constraint language over D . There are functions $d'_\Gamma(k)$ and $e'_\Gamma(k)$ such that for every instance of $\text{CSP}(\Gamma)$ with n variables, every assignment f has at most $d'_\Gamma(k)$ minimal satisfying extensions of size at most k and all these minimal extensions can be enumerated in time $e'_\Gamma(k)n^{O(1)}$.*

A consequence of Lemma 2.1 is that, as in [14], $\text{CCSP}(\Gamma)$ and $\text{OCSP}(\Gamma)$ can be reduced to a set of 0-valid instances. We enumerate all the minimal satisfying extensions of size at most k of the all zero assignment (where k is the size constraint) and obtain the 0-valid instances by substituting the nonzero values as constants.

Corollary 2.2. *Let Γ be a cc-language and let $\Gamma_0 \subseteq \Gamma$ be the set of all 0-valid relations. Then $\text{CCSP}(\Gamma)$ is FPT/W[1]-hard/BICLIQUE-hard if and only if $\text{CCSP}(\Gamma_0)$ is. The same holds for $\text{OCSP}(\Gamma)$ and $\text{OCSP}(\Gamma_0)$.*

A nonzero satisfying assignment f is said to be a *minimal (nonzero) satisfying assignment* if it is not a proper extension of any nonzero satisfying assignment.

Lemma 2.3. *Let Γ be a finite constraint language. There are functions $d_\Gamma(k)$ and $e_\Gamma(k)$ such that for any instance of $\text{CSP}(\Gamma)$ with n variables every variable v is nonzero in at most $d_\Gamma(k)$ minimal satisfying assignments of size at most k and all these minimal satisfying assignments can be enumerated in time $e_\Gamma(k)n^{O(1)}$.*

3 Properties of constraints

By Corollary 2.2, for proving Theorems 1.1 and 1.2 it is sufficient to consider only cc0-languages. Thus in the rest of the paper, we consider only cc0-languages.

3.1 Weak separability

In the Boolean case, the tractability of 0-valid constraints depends only on weak separability [14]. This is not true exactly this way for larger domains: as we shall see (Theorems 4.1 and 5.1), the complexity characterizations have further conditions.

Tuples $\mathbf{t}_1 = (a_1, \dots, a_r)$ and $\mathbf{t}_2 = (b_1, \dots, b_r)$ are *disjoint* if $a_i = 0$ or $b_i = 0$ for every i . The *union* of disjoint tuples \mathbf{t}_1 and \mathbf{t}_2 is $\mathbf{t}_1 + \mathbf{t}_2 = (c_1, \dots, c_r)$ where $c_i = a_i$ if $a_i \neq 0$ and $c_i = b_i$ otherwise. If (a_1, \dots, a_r) is an extension of (b_1, \dots, b_r) , then their *difference* is the tuple (c_1, \dots, c_r) where $c_i = a_i$ if $b_i = 0$ and $c_i = 0$ otherwise. A tuple \mathbf{t} is *contained* in a set $C \subseteq D$ if every nonzero entry of \mathbf{t} is in C .

A 0-valid relation R is said to be *weakly separable* if the following two conditions hold: (1) For every disjoint tuples $\mathbf{t}_1, \mathbf{t}_2 \in R$, we have $\mathbf{t}_1 + \mathbf{t}_2 \in R$. (2) For every disjoint tuples $\mathbf{t}_1, \mathbf{t}_2$ with $\mathbf{t}_2, \mathbf{t}_1 + \mathbf{t}_2 \in R$, we have $\mathbf{t}_1 \in R$. A constraint language Γ is said to be weakly separable if every relation from Γ is weakly separable. If constraint language Γ is not weakly separable, then we call a triple $(R, \mathbf{t}_1, \mathbf{t}_2)$, $R \in \Gamma$, witnessing that a *union counterexample* if $\mathbf{t}_1, \mathbf{t}_2$ violate condition (1), while if $\mathbf{t}_1, \mathbf{t}_2$ violate condition (2) it is called a *difference counterexample*.

The following combinatorial property is the key for solving weakly separable instances (this property does not necessarily hold for arbitrary relations):

Lemma 3.1. *Let Γ be a weakly separable finite cc0-language over D and I an instance of $\text{CCSP}(\Gamma)$ or $\text{OCSP}(\Gamma)$.*

- (1) *Any satisfying assignment of I is a union of pairwise disjoint minimal ones.*
- (2) *If there is a satisfying assignment f with $f(v) = d$ for some variable v and $d \in D$, then there is a minimal satisfying assignment f' with $f'(v) = d$.*

In light of Lemma 3.1(1), it is sufficient to enumerate every minimal assignment of size at most k (using Lemma 2.3) and then to find a disjoint minimal assignments that together satisfy the size/cardinality constraints. As the total size of the assignments we select is at most k and furthermore Lemma 2.3 implies that each variable is nonzero in at most a bounded number of these minimal assignments, the fixed-parameter tractability of finding such disjoint assignments can be shown by standard arguments.

Theorem 3.2. *Let Γ be a finite weakly separable cc0-language over D .*

1. *A solution to an instance (V, \mathcal{C}, k, π) of $\text{CCSP}(\Gamma)$ can be found in time $e_\Gamma(k)|V|^{O(1)}$.*
2. *A solution to an instance (V, \mathcal{C}, k) of $\text{OCSP}(\Gamma)$ can be found in time $k^{|D|-1}e_\Gamma(k)|V|^{O(1)}$.*

3.2 Morphisms

Homomorphisms and polymorphisms are standard tools for understanding the complexity of constraints [3,10]. We make use of the notion of multivalued morphisms, a

generalization of homomorphisms, that in a different context has appeared in the literature (see, e.g. [16]) under the guise of hyperoperation. We classify the values into 4 types according to the existence of such morphisms (Definition 3.3). This classification and the observation that these types play an essential role in the way the MVM gadgets (Section 3.4) work are the main technical ideas behind the hardness proofs.

For a subset $0 \in D' \subseteq D$ and an n -ary relation R on D , by $R|_{D'}$ we denote the relation $R \cap (D')^n$. For a language Γ , $\Gamma|_{D'}$ contains every relation $R|_{D'}$ for $R \in \Gamma$.

For a tuple $\mathbf{t} = (a_1, \dots, a_r) \in \text{dom}(\Gamma)^r$, we denote by $h(\mathbf{t})$ the tuple $(h(a_1), \dots, h(a_r))$. An *endomorphism* of Γ is a mapping $h : \text{dom}(\Gamma) \rightarrow \text{dom}(\Gamma)$ such that $h(0) = 0$ and for every $R \in \Gamma$ and $\mathbf{t} \in R$, the tuple $h(\mathbf{t})$ is also in R . Observe that the requirement $h(0) = 0$ is nonstandard, but it is natural in our setting. The mapping sending all elements of $\text{dom}(\Gamma)$ to 0 is an endomorphism of any 0-valid language. An *inner homomorphism* of Γ from D_1 to D_2 with $0 \in D_1, D_2 \subseteq \text{dom}(\Gamma)$ is a mapping $h : D_1 \rightarrow D_2$ such that $h(0) = 0$ and $h(\mathbf{t}) \in R$ holds for any r -ary relation $R \in \Gamma$ and $\mathbf{t} \in D_1^r \cap R$.

A *multivalued morphism* of Γ is a mapping $\phi : \text{dom}(\Gamma) \rightarrow 2^{\text{dom}(\Gamma)}$ such that $\phi(0) = \{0\}$ and for every $R \in \Gamma$ and $(a_1, \dots, a_r) \in R$, we have $\phi(a_1) \times \dots \times \phi(a_r) \subseteq R$. An *inner multivalued morphism* ϕ from D_1 to D_2 where $0 \in D_1, D_2 \subseteq \text{dom}(\Gamma)$ is defined to be a mapping $\phi : D_1 \rightarrow 2^{D_2}$ such that $\phi(0) = \{0\}$ and for every $R \in \Gamma$ and $(a_1, \dots, a_r) \in R|_{D_1}$, we have $\phi(a_1) \times \dots \times \phi(a_r) \subseteq R|_{D_2}$.

Observe that if $\phi : \text{dom}(\Gamma) \rightarrow 2^{\text{dom}(\Gamma)}$ is a multivalued morphism of a constraint language Γ , and $\phi' : \text{dom}(\Gamma) \rightarrow 2^{\text{dom}(\Gamma)}$ is a mapping such that $\phi'(d) \subseteq \phi(d)$ for $d \in \text{dom}(\Gamma)$, then ϕ' is a multivalued morphism. Similar statement holds for inner multivalued morphisms $\psi, \psi' : D_1 \rightarrow 2^{D_2}$.

The *product* $g \circ h$ of two endomorphisms or inner homomorphisms is defined by $(g \circ h)(x) = h(g(x))$ for every $x \in D$. If ϕ and ψ are (inner) multivalued morphisms then their product $\phi \circ \psi$ is given by $(\phi \circ \psi)(x) = \bigcup_{y \in \phi(x)} \psi(y)$.

For $x, y \in \text{dom}(\Gamma)$, we say that x *produces* y in Γ if Γ has a multivalued morphism ϕ with $\phi(x) = \{0, y\}$ and $\phi(z) = \{0\}$ for every $z \neq x$. Observe that the relation “ x produces y ” is transitive.

Definition 3.3. A value $y \in \text{dom}(\Gamma)$ is

1. *regular* if there is no multivalued morphism ϕ where $0, y \in \phi(x)$ for some $x \in \text{dom}(\Gamma)$,
2. *semi-regular* if there is a multivalued morphism ϕ where $0, y \in \phi(x)$ for some $x \in \text{dom}(\Gamma)$, but there is no $x \in \text{dom}(\Gamma)$ that produces y ,
3. *self-producing* if y produces y , and for every x that produces y , y also produces x .
4. *degenerate otherwise*.

It will sometimes be convenient to say that a value y has *type* 1, 2, 3, or 4. We need the following simple properties:

Proposition 3.4. *If there is an endomorphism h with $h(x) = y$, then the type of x cannot be larger than that of y .*

Proposition 3.5. *Every degenerate value y is produced by a nondegenerate value x .*

3.3 Components

The structure of endomorphisms and inner homomorphisms plays an important role in our study. Let Γ be a cc0-language. A *retraction* to $X \subseteq D \setminus \{0\}$ is a mapping ret_X such that $\text{ret}_X(x) = x$ for $x \in X$ and $\text{ret}_X(x) = 0$ otherwise. A nonempty subset $C \subseteq D \setminus \{0\}$ is a *component* of Γ if ret_C is an endomorphism of Γ . A component C is *minimal* if there is no component that is a proper subset of C . If a set C is not a component, then there is a relation $R \in \Gamma$ and $\mathbf{t} \in R$ such that $\mathbf{t}' = \text{ret}_C \mathbf{t} \notin R$. Observe that the intersection of two components is also a component (if it is nonempty). Hence for every nonempty $X \subseteq D \setminus \{0\}$, there is a unique inclusion-wise minimal component that contains X ; this component is called the component *generated by* X (or simply the component of X). The importance of components comes from the following result:

Lemma 3.6. *If Γ is not weakly separable, then either*

- *there is a union counterexample $(R, \mathbf{t}_1, \mathbf{t}_2)$ such that \mathbf{t}_1 (resp., \mathbf{t}_2) is contained in a component generated by a value a_1 (resp., a_2), or*
- *there is a difference counterexample $(R, \mathbf{t}_1, \mathbf{t}_2)$ such that both \mathbf{t}_1 and \mathbf{t}_2 are contained in a component generated by a value a_1 .*

3.4 Multivalued morphism gadgets

For a relation R and a tuple $\mathbf{t} \in R$, we denote by $\text{supp}(\mathbf{t})$ the set of coordinate positions of \mathbf{t} occupied by nonzero elements. Let $\text{supp}_{\mathbf{t}}(R)$ denote the relation obtained by substituting 0 into all coordinates of R except for $\text{supp}(\mathbf{t})$, i.e. if R is r -ary and $\text{supp}(\mathbf{t}) = \{1, \dots, r\} \setminus \{i_1, \dots, i_r\}$ then $\text{supp}_{\mathbf{t}}(R) = R^{i_1, \dots, i_r; 0, \dots, 0}$.

For a cc0-language Γ and some $0 \in D' \subseteq \text{dom}(\Gamma)$, a *multivalued morphism gadget* $\text{MVM}(\Gamma, D')$ consists of $|D'| - 1$ bags of vertices B_d , $d \in D' \setminus \{0\}$. The number of variables in each bag will be specified every time it is used. The gadget is equipped with the following set of constraints. For every $R \in \Gamma$ and every tuple $\mathbf{t} = (a_1, \dots, a_r) \in R_{|D'}$ (with, say, $\text{supp}(\mathbf{t}) = \{i_1, \dots, i_q\}$), we add all possible constraints $\langle \mathbf{s}, \text{supp}_{\mathbf{t}}(R) \rangle$ where $\mathbf{s} = (v_{i_1}, \dots, v_{i_q})$ such that $v_j \in B_{a_j}$ for every $j \in \{i_1, \dots, i_q\}$. The *standard assignment* of a gadget assigns a to every variable in bag B_a ; observe that it assignment satisfies every constraint of the gadget. We say that bag B_a and the variables in bag B_a *represent* a .

Proposition 3.7. *Let $0 \in D' \subseteq \text{dom}(\Gamma)$. Consider a satisfying assignment f of an $\text{MVM}(\Gamma, D')$ gadget. If $h_f : D' \rightarrow 2^{\text{dom}(\Gamma)}$ is a mapping such that $h_f(a)$ is the set of values appearing in bag B_a of the gadget and $h_f(0) = \{0\}$, then h_f is an inner multivalued morphism of Γ from D' to $\text{dom}(\Gamma)$.*

We define gadgets connecting MVM gadgets. The gadget $\text{NAND}(G_1, G_2)$ on $\text{MVM}(\Gamma, D')$ gadgets G_1, G_2 has constraints as follows. For every $R \in \Gamma$ and disjoint tuples $\mathbf{t}_1 = (a_1, \dots, a_r), \mathbf{t}_2 = (b_1, \dots, b_r)$ in $R_{|D'}$, we add a constraint $\langle \mathbf{s}, \text{supp}_{\mathbf{t}_1 + \mathbf{t}_2} R \rangle$, where $\mathbf{s} = (v_{i_1}, \dots, v_{i_q})$ with $\{i_1, \dots, i_q\} = \text{supp}(\mathbf{t}_1 + \mathbf{t}_2)$, such that v_j for $j \in \{i_1, \dots, i_q\}$ is in bag B_{a_j} of G_1 if $a_j \neq 0$ and v_j is in bag B_{b_j} of G_2 if $b_j \neq 0$.

If one of G_1, G_2 has the standard assignment and the other is fully zero, then all the constraints in $\text{NAND}(G_1, G_2)$ are satisfied. On the other hand, if both G_1 and G_2 have

the standard assignment and there is a union counterexample, then $\text{NAND}(G_1, G_2)$ is not satisfied. For the reductions, we need this second conclusion not only if both G_1 and G_2 have the standard assignment, but also assignments that “behave well” in some sense. The right notion for our purposes is the following: An inner homomorphism $h : D' \rightarrow \text{dom}(\Gamma)$ is \mathbf{t} -recoverable if Γ has an endomorphism h' such that $(h \circ h')(\mathbf{t}) = \mathbf{t}$.

Lemma 3.8. *Let $0 \in D' \subseteq \text{dom}(\Gamma)$ and let there be a $\text{NAND}(G_1, G_2)$ gadget on $\text{MVM}(\Gamma, D')$ gadgets G_1, G_2 .*

(1) *If one of G_1 and G_2 has the standard assignment and the other gadget is fully zero, then all constraints of $\text{NAND}(G_1, G_2)$ are satisfied.*

(2) *If $\Gamma|_{D'}$ has a union counterexample $(R, \mathbf{t}_1, \mathbf{t}_2)$ and an assignment τ is such that for $i = 1, 2$, τ on G_i is a \mathbf{t}_i -recoverable inner homomorphism h_i , then $\text{NAND}(G_1, G_2)$ is not satisfied.*

The $\text{IMP}(G_1, G_2)$ gadget is defined similarly, but instead of $\mathbf{t}_1, \mathbf{t}_2 \in R|_{D'}$, we require $\mathbf{t}_2, \mathbf{t}_1 + \mathbf{t}_2 \in R|_{D'}$. An analog of Lemma 3.8 holds for such gadgets.

When the multivalued morphism gadgets are used in the reductions, it will be essential that the bags of the gadgets have very specific sizes. We will ensure somehow that in a solution each bag is either fully zero or fully nonzero. Our aim is to choose the sizes of the bags in such a way that if the sum of the sizes of certain bags add up to a certain integer, then this is only possible if there is exactly one bag of each size.

Fix an integer t and a set $0 \in D' \subseteq D$. It will be convenient to assume that $D' = \{0, 1, \dots, d\}$. By $\mathcal{Z}^{t, D'}$ we denote the set of integers $Z_{i,j}^{t, D'}$ for $1 \leq i \leq t$ and $1 \leq j \leq d$, given by $Z_{i,j}^{t, D'} := (4td)^{2td+(id+j)} + (4td)^{5td-(id+j)}$.

Lemma 3.9. *Let us fix t and $D' = \{0, 1, \dots, d\}$. If $A \subseteq \mathcal{Z}^{t, D'}$ and \mathcal{B} is a multiset of values from $\mathcal{Z}^{t, D'}$ with $|\sum_{S \in A} S - \sum_{S \in \mathcal{B}} S| < (4td)^{2td}$, then \mathcal{B} is a set and $\mathcal{B} = A$.*

3.5 Frequent instances

The following property plays an important role in our algorithms. We say that an instance of $\text{CCSP}(\Gamma)$ or $\text{OCSP}(\Gamma)$, with parameter k is c -frequent (for some integer c) if for every $d \in \text{dom}(\Gamma) \setminus \{0\}$ there are at least c variables that take value d in satisfying assignments of size at most k . The algorithm of Lemma 2.1 can be used to decide in fpt-time whether an instance is c -frequent. Lemma 3.10 shows that if an instance is not c -frequent, it can be reduced to c -frequent instances satisfying an additional technical requirement. This is done by eliminating values that appear on less than c variables one by one. A subset $0 \in D' \subseteq \text{dom}(\Gamma)$ is *closed* (with respect to Γ) if Γ has no inner homomorphism from D' that maps some element of D' to an element in $\text{dom}(\Gamma) \setminus D'$.

Lemma 3.10. *Let Γ be a finite cc0-language. Given an instance I of $\text{CCSP}(\Gamma)$ or $\text{OCSP}(\Gamma)$ with parameter k and an integer c , we can construct in time $f_\Gamma(k, c)n^{O(1)}$ a set of c -frequent instances such that*

1. *instance I has a solution iff at least one of the constructed instances has a solution,*
2. *each instance I_i is an instance of $\text{CCSP}(\Gamma|_{D_i})$, respectively, $\text{OCSP}(\Gamma|_{D_i})$, for some $D_i \subseteq \text{dom}(\Gamma)$ closed in Γ , and*
3. *the parameter k_i of I_i is at most k .*

4 Classification for size constraints

Unlike in the Boolean case, weak separability of Γ is not equivalent to the tractability of $\text{OCSP}(\Gamma)$: it is possible that Γ is not weakly separable, but $\text{OCSP}(\Gamma)$ is FPT. However, if there is a subset $D' \subseteq \text{dom}(\Gamma)$ of the domain such that $\Gamma|_{D'}$ is not weakly separable and D' has “no special problems” in a certain technical sense, then $\text{OCSP}(\Gamma)$ is $\text{W}[1]$ -hard. We need the following definitions. A value $d \in \text{dom}(\Gamma)$ is *weakly separable* if $\Gamma|_{\{0,d\}}$ is weakly separable. A *contraction* of Γ to D' with $0 \in D' \subseteq \text{dom}(\Gamma)$ is an endomorphism $h : \text{dom}(\Gamma) \rightarrow D'$ such that $h(d) \neq 0$ for any $d \in \text{dom}(\Gamma) \setminus \{0\}$. Contraction h is *proper* if $D' \neq \text{dom}(\Gamma)$.

The main result for the size constraints CSP is the following dichotomy theorem.

Theorem 4.1. *Let Γ be a finite cc0-language. If there are two sets $\{0\} \subseteq D_2 \subseteq D_1 \subseteq \text{dom}(\Gamma)$ such that (1) D_1 is closed in Γ , (2) $\Gamma|_{D_1}$ has a contraction h to D_2 , (3) $\Gamma|_{D_2}$ has no proper contraction, (4) $\Gamma|_{D_1}$ has no weakly separable value that is either degenerate or self-producing, and (5) $\Gamma|_{D_2}$ is not weakly separable, then $\text{OCSP}(\Gamma)$ is $\text{W}[1]$ -hard. If there are no such D_1, D_2 , then $\text{OCSP}(\Gamma)$ is FPT.*

We present an algorithm solving the FPT cases of the problem and then an important case of the hardness proof, demonstrating the concepts introduced in Section 3.

The algorithm. Let $I = (V, \mathcal{C}, k)$ be an instance of $\text{OCSP}(\Gamma)$. Let us use Lemma 3.10 to obtain instances I_1, \dots, I_ℓ such that I_i is a k -frequent instance of $\text{OCSP}(\Gamma|_{D^i})$ for some closed set $D^i \subseteq \text{dom}(\Gamma)$. Fix some i and let h be a contraction of $\Gamma|_{D^i}$ such that $|h(D^i)|$ is minimum possible. Set $D_1 := D^i$ and $D_2 := h(D^i)$.

The pair D_1, D_2 violates one of properties (1)–(5) in Theorem 4.1. By the way D_1 and D_2 defined, it is clear that (1) and (2) hold. If the pair violates (3), then let g be a proper contraction of $\Gamma|_{D_2}$. Then $h \circ g$ is a contraction of $\Gamma|_{D_1}$ such that $|g(h(D_1))|$ is strictly less than $|h(D_1)|$, a contradiction. If D_1, D_2 violate (4), then instance I_i always has a solution. Indeed, suppose that $d \in D_1$ is weakly separable and d is produced by $d' \in D_1$ (possibly $d = d'$). Let k_i be the parameter of I_i ; then $k_i \leq k$ by Lemma 3.10(4). Since I_i is k -frequent, the set S of variables of I_i where d' can appear in a satisfying assignment of size at most k contains at least k elements. As d' produces d , $\Gamma|_{D_1}$ has a multivalued morphism ϕ such that $\phi(d') = \{0, d\}$ and $\phi(a) = \{0\}$ for $a \in D_1 \setminus \{d'\}$. Therefore, for every $v \in S$, the assignment $\delta_{v,d}$ with $\delta_{v,d}(v) = d$ and 0 everywhere else is a satisfying assignment of I_i . As d is weakly separable in $\Gamma|_{D_1}$, the disjoint union of k_i such assignments $\delta_{v,d}$ is a solution to I_i . Finally, if (5) is violated, then instance I_i of $\text{OCSP}(\Gamma|_{D_1})$ has a solution if and only if it has a solution restricted to D_2 , and the latter can be decided using Lemma 3.2 (as $\Gamma|_{D_2}$ is weakly separable).

Hardness. We say that a set p_1, \dots, p_ℓ of endomorphisms of Γ is a *partition set* if, for every $d \in D' \setminus \{0\}$, $p_i(d) \neq 0$ for exactly one i . The *sum* of the partition set is the mapping h defined such that $h(d)$ is the unique nonzero value in $p_1(d), \dots, p_\ell(d)$. The partition set is *good* if the sum of these pairwise disjoint endomorphisms is also an endomorphism; otherwise, the partition set is *bad*.

Lemma 4.2. *If every value is regular in $\Gamma|_{D_2}$, there is no bad partition set in $\Gamma|_{D_2}$, and there is a union counterexample in $\Gamma|_{D_2}$, then $\text{OCSP}(\Gamma)$ is $\text{W}[1]$ -hard.*

Proof. Assume $D_2 = \{0, 1, \dots, p\}$. The reduction is from MULTICOLORED INDEPENDENT SET, the following W[1]-hard problem: Given a graph G with vertices $v_{i,j}$ ($1 \leq i \leq t, 1 \leq j \leq n$), find an independent set of size t of the form $\{v_{1,y_1}, \dots, v_{t,y_t}\}$. For each $v_{i,j}$, we introduce a gadget $\text{MVM}(\Gamma, D_2)$ denoted by $G_{i,j}$. The bag of $G_{i,j}$ corresponding to value $d \in D_2 \setminus \{0\}$ has size $Z_{i,d}^{t,D_2}$. The size constraint is $k := \sum_{i=1}^t \sum_{d \in D_2 \setminus \{0\}} Z_{i,d}^{t,D_2}$. If $v_{i,j}$ and $v_{i',j'}$ are adjacent, then we add the gadget $\text{NAND}(G_{i,j}, G_{i',j'})$. Also, for every $1 \leq i \leq t, 1 \leq j < j' \leq n$, we add the $\text{NAND}(G_{i,j}, G_{i,j'})$ gadget.

Suppose that there is a solution C of size exactly t for the MULTICOLORED INDEPENDENT SET instance. If vertex $v_{i,j}$ is in C , then set the standard assignment on gadget $G_{i,j}$, otherwise set the zero assignment. It is clear that this results in an assignment satisfying the size constraint. By Lemma 3.8(1), the constraints of the $\text{MVM}(\Gamma, D_2)$ gadgets as well as the $\text{NAND}(G_{i,j}, G_{i,j'})$ gadgets are satisfied.

For the other direction, suppose that there is a solution τ satisfying the size constraint. First we observe that τ contains values only from D_1 . Indeed, if $c \notin D_1$ appears in bag B_d of a gadget $G_{i,j}$, then τ on $G_{i,j}$ is an inner homomorphism g of Γ from D_2 with $g(d) = c$. Now $h \circ g$ maps a value of D_1 to c , contradicting the assumption that D_1 is a closed set. By applying h on a solution, it can be assumed that only values from D_2 are used. Thus τ on the $\text{MVM}(\Gamma, D_2)$ gadgets provides multivalued morphisms of $\Gamma|_{D_2}$. Since every value is regular in $\Gamma|_{D_2}$, each bag is either fully zero or fully nonzero. The sizes of the nonzero bags add up exactly to the size constraint k . Thus by Lemma 3.9, there is exactly one nonzero bag with size $Z_{i,d}^{t,D_2}$ for every $1 \leq i \leq t$ and $d \in D_2 \setminus \{0\}$.

Take a union counterexample $(R, \mathbf{t}_1, \mathbf{t}_2)$ in $\Gamma|_{D_2}$; by Lemma 3.6, we can assume that $\mathbf{t}_1, \mathbf{t}_2$ are in the components of $\Gamma|_{D_2}$ generated by some $a_1, a_2 \in D_2$, respectively. We show that for every $1 \leq i \leq t$, there are values y_i^1 and y_i^2 such that every endomorphism of $\Gamma|_{D_2}$ given by G_{i,y_i^1} (resp., G_{i,y_i^2}) is \mathbf{t}_1 -recoverable (resp., \mathbf{t}_2 -recoverable). For a fixed i , let g_1, \dots, g_n be arbitrary endomorphisms of $\Gamma|_{D_2}$ given by $G_{i,1}, \dots, G_{i,n}$, respectively. Since the sizes of nonzero bags are all different, these endomorphisms are pairwise disjoint and they form a partition set. As there is no bad partition set in $\Gamma|_{D_2}$, their sum g is an endomorphism of $\Gamma|_{D_2}$. Since $\Gamma|_{D_2}$ has no proper contractions, g has to be a permutation and hence g^s is the identity for some $s \geq 1$. There is a unique $1 \leq y_i^1 \leq n$ such that $g_{y_i^1}(a_1) \neq 0$. The homomorphism $g_{y_i^1} \circ g^{s-1}$ maps every $a \in D_2$ either to 0 or a ; i.e., $g_{y_i^1} \circ g^{s-1} = \text{ret}_S$ for some set $S \subseteq D_2$ containing a_1 . Hence S is a component containing a_1 and S contains every value of \mathbf{t}_1 . It follows that $g_{y_i^1}$ given by G_{i,y_i^1} is \mathbf{t}_1 -recoverable. A similar argument works for y_j^2 , thus the required values y_i^1, y_i^2 exist. Let us observe that it is not possible that $y_i^1 \neq y_i^2$: by Lemma 3.8(2) the constraints of $\text{NAND}(G_{i,y_i^1}, G_{i,y_i^2})$ are not satisfied in this case. Let C contain vertex $v_{i,j}$ if $j = y_i^1 = y_i^2$. It follows that C is a multicolored independent set: if vertices $v_{i,j}, v_{i',j'}$ are adjacent, then some constraint of $\text{NAND}(G_{i,j}, G_{i',j'}) = \text{NAND}(G_{i,y_j^1}, G_{i',y_j^2})$ is not satisfied. \square

5 Classification for cardinality constraints

The characterization of the complexity of $\text{CCSP}(\Gamma)$ requires a new definition, which was not relevant for $\text{OCSP}(\Gamma)$. The *core* of Γ is the component generated by the set of all nondegenerate values in $\text{dom}(\Gamma)$. We say that Γ is a core if the core of Γ is $\text{dom}(\Gamma)$.

Theorem 5.1. *Let Γ be a cc0-language. If there is a $0 \in D' \subseteq \text{dom}(\Gamma)$ s.t. $\Gamma|_{D'}$ is a core and not weakly separable, then $\text{CCSP}(\Gamma)$ is BICLIQUE-hard, and FPT otherwise.*

A significant difference between the hardness proofs of $\text{OCSP}(\Gamma)$ and $\text{CCSP}(\Gamma)$ is that in $\text{OCSP}(\Gamma)$, we can assume that no proper contraction exists and this can be used to show that certain endomorphisms have to be permutations (see Lemma 4.2). For $\text{CCSP}(\Gamma)$, we cannot make this assumption, thus we need a delicate argument, making use of the cardinality constraint, to achieve a similar effect.

References

1. Bessière, C., Hebrard, E., Hnich, B., Walsh, T.: The complexity of global constraints. In: Wallace, M. (ed.) AAAI. LNCS, vol. 3258, pp. 112–117. Springer (2004)
2. Bulatov, A.: Tractable conservative constraint satisfaction problems. In: LICS. pp. 321–330. IEEE Computer Society (2003)
3. Bulatov, A.A., Jeavons, P., Krokhin, A.A.: Classifying the complexity of constraints using finite algebras. SIAM J. Comput. 34(3), 720–742 (2005)
4. Bulatov, A.A., Marx, D.: The complexity of global cardinality constraints. In: LICS. pp. 419–428. IEEE Computer Society (2009)
5. Creignou, N., Schnoor, H., Schnoor, I.: Non-uniform boolean constraint satisfaction problems with cardinality constraint. CSL. LNCS, vol. 5213, pp. 109–123. Springer (2008)
6. Downey, R.G., Fellows, M.R.: Parameterized Complexity (1999)
7. Feder, T., Vardi, M.Y.: Monotone monadic snp and constraint satisfaction. In: STOC. pp. 612–622 (1993)
8. Flum, J., Grohe, M.: Parameterized Complexity Theory. Springer, Berlin (2006)
9. Jeavons, P., Cohen, D., Gyssens, M.: Closure properties of constraints. J. ACM 44, 527–548 (1997)
10. Jeavons, P., Cohen, D., Gyssens, M.: How to determine the expressive power of constraints. Constraints 4, 113–131 (1999)
11. Khanna, S., Sudan, M., Trevisan, L., Williamson, D.P.: The approximability of constraint satisfaction problems. SIAM J. Comput. 30(6), 1863–1920 (2001)
12. Kratsch, S., Wahlström, M.: Preprocessing of min ones problems: A dichotomy. In: ICALP (1). LNCS, vol. 6198, pp. 653–665. Springer (2010)
13. Krokhin, A.A., Marx, D.: On the hardness of losing weight. In: ICALP. LNCS, vol. 5125, pp. 662–673. Springer (2008)
14. Marx, D.: Parameterized complexity of constraint satisfaction problems. Computational Complexity 14
15. Régin, J.C., Gomes, C.P.: The cardinality matrix constraint. In: CP. LNCS, vol. 3258, pp. 572–587. Springer (2004)
16. Rosenberg, I.: Multiple-valued hyperstructures. In: ISMVL. pp. 326–333 (1998)
17. Schaefer, T.J.: The complexity of satisfiability problems. In: STOC. pp. 216–226 (1978)
18. Szeider, S.: The parameterized complexity of k-flip local search for sat and max sat. SAT. LNCS, vol. 5584, pp. 276–283. Springer (2009)