Collapsibility in Infinite-Domain Quantified Constraint Satisfaction

Manuel Bodirsky[1] and Hubie Chen[2]

[1] Institut für Informatik
Humboldt-Universität zu Berlin
Berlin, Germany
bodirsky@informatik.hu-berlin.de
[2] Departament de Tecnologia
Universitat Pompeu Fabra
Barcelona, Spain
hubie.chen@upf.edu

Abstract. In this article, we study the quantified constraint satisfaction problem (QCSP) over infinite domains. We develop a technique called collapsibility that allows one to give strong complexity upper bounds on the QCSP. This technique makes use of both logical and universal-algebraic ideas. We give applications illustrating the use of our technique.

1 Introduction

The *constraint satisfaction problem* (CSP) is the problem of deciding the truth of a primitive positive sentence

$$\exists v_1 \ldots \exists v_n (R(v_{i_1}, \ldots, v_{i_k}) \wedge \ldots)$$

over a relational signature, relative to a given relational structure over the same signature. Informally, the goal in an instance of the CSP is to decide if there exists an assignment to a set of variables simultaneously satisfying a collection of constraints. Many search problems in computer science can be naturally formulated as CSPs, such as boolean satisfiability problems, graph homomorphism problems, and the problem of solving a system of equations (over some algebraic structure). The CSP can be equivalently formulated as the relational homomorphism problem [14], or the conjunctive-query containment problem [18].

The ubiquity of the CSP in conjunction with its general intractability has given rise to an impressive research program seeking to identify restricted cases of the CSP that are polynomial-time tractable. In particular, much attention has been focused on identifying those relational structures Γ such that $\mathsf{CSP}(\Gamma)$–the CSP where the relational structure is fixed to be Γ–is polynomial-time tractable. In a problem $\mathsf{CSP}(\Gamma)$, we call Γ the *constraint language*, and use the term *domain* to refer to the universe of Γ. Many recent results have studied the problems $\mathsf{CSP}(\Gamma)$ for finite-domain constraint languages Γ, see for example [8, 9, 7, 6, 13] and the references therein. However, it has been recognized that many natural combinatorial problems from areas such as graph theory and temporal reasoning can be expressed as problems of the form $\mathsf{CSP}(\Gamma)$ only if infinite-domain Γ are permitted [1]. This has motivated the study of constraint satisfaction problems $\mathsf{CSP}(\Gamma)$ on infinite domains [1, 2, 4].

A recent subject of inquiry that builds upon CSP research is the *quantified constraint satisfaction problem (QCSP)*, which is the generalization of the CSP where both existential and universal quantification is allowed, as opposed to just existential quantification. As is well-known, the extra expressiveness of the QCSP comes with an increase in complexity: the finite-domain QCSP is PSPACE-complete, in contrast to the finite-domain CSP, which is NP-complete. Recent work on the QCSP includes that of Börner, Bulatov, Krokhin, and Jeavons [5], Chen [11, 10, 12], Gottlob, Greco, and Scarcello [15], and Pan and Vardi [20].

In this paper, we consider infinite-domain quantified constraint satisfaction. Our contribution is to introduce, in the infinite-domain setting, a technique called *collapsibility* that allows us to give complexity upper bounds on problems of the form $\mathsf{QCSP}(\Gamma)$, such as NP upper bounds, that are *dramatically* lower than the "obvious" upper bound of PSPACE that typically applies. On a high level, collapsibility allows one to show that, for certain contraint languages Γ, an arbitrary instance of $\mathsf{QCSP}(\Gamma)$ can be reduced to the conjunction of instances of $\mathsf{QCSP}(\Gamma)$ that are simpler in that they have only a constant number of (or no) universally quantified variables; typically, such a conjunction can be cast as an instance of $\mathsf{CSP}(\Gamma')$ for some constraint langauge Γ' with $\mathsf{CSP}(\Gamma')$ in NP, and hence the reduction yields a proof that $\mathsf{QCSP}(\Gamma)$ is in NP.

To develop our collapsibility technique, we make use of a universal-algebraic approach to studying the complexity of constraint languages; this approach associates a set of operations called *polymorphisms* to each constraint language, and uses this set of operations to derive information about complexity. While the present work takes inspiration from technology that was developed in the finite-domain setting [11, 10] for similar purposes, there are a number of differences between the infinite and finite settings that necessitate the use of more involved and intricate argumentation in the infinite setting. One is that, while there is a canonical choice for the aforementioned simpler instances in the finite setting, in the infinite setting there is no such canonical choice and indeed often an expansion of the constraint language is required to achieve a reduction from the QCSP to the CSP. Another is that, in the infinite setting, any assignment or partial assignment f to variables induces, via the automorphism group of Γ, an *orbit* of assignments $\{\sigma(f) : \sigma \text{ is an automorphism of } \Gamma\}$. The property of an assignment satisfying constraints over Γ is orbit-invariant, but in the presence of universal quantification, one needs to make inferences about the orbit of an assignment in a careful way (see Lemma 3 and its applications).

2 Preliminaries

When A and B are sets, we use $[A \to B]$ to denote the set of functions mapping from A to B. When $f : A \to B$ is a function and A' is a subset of A, we use $f|_{A'}$

to denote the restriction of f to A'. We extend this notation to sets of functions: when $F \subseteq [A \to B]$ and A' is a subset of A, we use $F|_{A'}$ to denote the set $\{f|_{A'} : f \in F\}$. When $f : A \to B$ is a function, we use the notation $f[a' \to b']$ to denote the extension of f mapping a' to b'. We will use $[k]$ to denote the first k positive integers, $\{1, \ldots, k\}$.

Relational structures. A *relational language* τ is a (in this paper always finite) set of *relation symbols* R_i, each of which has an associated finite *arity* k_i. A *(relational) structure* Γ over the *(relational) language* τ (also called τ-*structure*) is a set D_Γ (the *domain* or *universe*) together with a relation $R_i \subseteq D_\Gamma^{k_i}$ for each relation symbol R_i from τ. For simplicity, we use the same symbol for a relation symbol and the corresponding relation. If necessary, we write R^Γ to indicate that we are talking about the relation R belonging to the structure Γ. For a τ-structure Γ and $R \in \tau$ it will also be convenient to say that $R(u_1, \ldots, u_k)$ *holds in* Γ iff $(u_1, \ldots, u_k) \in R$. If we add relations to a given structure Γ we call the resulting structure Γ' an *expansion* of Γ, and Γ is called a *reduct* of Γ'.

Homomorphisms. Let Γ and Γ' be τ-structures. A *homomorphism* from Γ to Γ' is a function f from D_Γ to $D_{\Gamma'}$ such that for each n-ary relation symbol R in τ and each n-tuple $(a_1, \ldots, a_n)$, if $(a_1, \ldots, a_n) \in R^\Gamma$, then $(f(a_1), \ldots, f(a_n)) \in R^{\Gamma'}$. In this case we say that the map f *preserves* the relation R. Isomorphisms from Γ to Γ are called *automorphisms*, and homomorphisms from Γ to Γ are called *endomorphisms*. The set of all automorphisms of a structure Γ is a group, and the set of all endomorphisms of a structure Γ is a monoid with respect to composition. When referring to an automorphism of Γ, we sometimes use the term Γ-*automorphism* to make clear the relational structure. An *orbit of k-tuples in* Γ is a set of k-tuples of the form $\{(a(s_1), \ldots, a(s_k)) :$ a is an automorphism of $\Gamma\}$ for some tuple $(s_1, \ldots, s_k)$.

Polymorphisms. Let D be a countable set, and O be the set of *finitary operations* on D, i.e., functions from D^k to D for finite k. We say that a k-ary operation $f \in O$ *preserves* an m-ary relation $R \subseteq D^m$ if whenever $R(x_1^i, \ldots, x_m^i)$ holds for all $1 \leq i \leq k$ in Γ, then $R\big(f(x_1^1, \ldots, x_1^k), \ldots, f(x_m^1, \ldots, x_m^k)\big)$ holds in Γ. If f preserves all relations of a relational τ-structure Γ, we say that f is a polymorphism of Γ. In other words, f is a homomorphism from $\Gamma^k = \Gamma \times \ldots \times \Gamma$ to Γ, where $\Gamma_1 \times \Gamma_2$ is the *(categorical- or cross-) product* of the two relational τ-structures Γ_1 and Γ_2. Hence, the unary polymorphisms of Γ are the endomorphisms of Γ.

Quantified constraint satisfaction. We define a τ-formula to be a *quantified constraint formula* if it has the form $Q_1 v_1 \ldots Q_n v_n (\psi_1 \wedge \ldots \wedge \psi_m)$, where each Q_i is a quantifier from $\{\forall, \exists\}$, and each ψ_i is an atomic τ-formula that can contain variables from $\{v_1, \ldots, v_n\}$.

The quantified constraint satisfaction problem over a τ-structure Γ, denoted by $\mathsf{QCSP}(\Gamma)$, is the problem of deciding, given a quantified constraint formula over τ, whether or not the formula is true under Γ. Note that both the universal and existential quantification is understood to take place over the entire universe of Γ. We use D throughout the paper to denote the universe of a constraint language Γ under discussion. The constraint satisfaction problem over a τ-structure Γ, denoted by $\mathsf{CSP}(\Gamma)$, is the restriction of $\mathsf{QCSP}(\Gamma)$ to instances only including existential quantifiers.

A *constraint language* is simply a relational structure; we typically refer to a relational structure Γ as a constraint language when we are interested in the computational problem $\mathsf{QCSP}(\Gamma)$ or $\mathsf{CSP}(\Gamma)$. We also refer to Γ as a *template*.

We will illustrate the use of our technique on examples drawn from the following two classes of constraint languages.

Equality constraint languages. An *equality-definable relation* is a relation (on an infinite domain) that can be defined by a boolean combination of atoms of the form $x = y$. An *equality constraint language* is a relational structure having an countably infinite universe D and such that all of its relations are equality-definable relations over D.

When Γ is an equality constraint language with domain D, any permutation of D is an automorphism of Γ, that is, the automorphism group of Γ is the full symmetric group on D. Observe that, if a tuple $t = (t_1, \ldots, t_k)$ is an element of an equality-definable relation $R \subseteq D^k$, then all tuples of the form $(\pi(t_1), \ldots, \pi(t_k))$, where π is a permutation on D, are also contained in R. In studying equality constraint languages, it is therefore natural for us to associate to each tuple $(t_1, \ldots, t_k)$ the equivalence relation ρ on $\{1, \ldots, k\}$ where $i = j$ if and only if $t_i = t_j$. This is because, by our previous observation, a tuple $t = (t_1, \ldots, t_k)$ is in an equality-definable relation R if and only if all k-arity tuples inducing the same equivalence relation as t are in R. We may therefore view an equality-definable relation of arity k as the union of equivalence relations on $\{1, \ldots, k\}$.

It is known that for an equality constraint language Γ, $\mathsf{CSP}(\Gamma)$ is polynomial-time tractable if Γ has a constant unary polymorphism or an injective binary polymorphism, and is NP-complete otherwise [4]. It is also known (and not difficult to verify) that for every equality constraint language Γ, the problem $\mathsf{QCSP}(\Gamma)$ is in PSPACE [3]. In general, the quantified constraint satisfaction problem for equality constraint languages is PSPACE-complete [3]; this is closely related to a result of [21].

Temporal constraint languages. A *temporal relation* is a relation on the domain $\mathbb{Q}$ (the rational numbers) that can be defined by a boolean combination of expressions of the form $x < y$. A *temporal constraint language* is a relational structure having $\mathbb{Q}$ as universe and such that all of its relations are temporal relations. As with equality constraint languages, it is known and not difficult to verify that for every temporal constraint language Γ, the problem $\mathsf{CSP}(\Gamma)$ is in NP, the problem $\mathsf{QCSP}(\Gamma)$ is in PSPACE, and there are temporal constraint languages Γ such that $\mathsf{QCSP}(\Gamma)$ is PSPACE-complete. Temporal constraint languages are well-studied structures in model theory (e.g., they are all ω-categorical; see [16]).

3 Collapsibility

In this section, we present our collapsibility technology. We begin by introducing some notation and terminology.

When Φ is a quantified constraint formula, let V^{Φ} denote the variables of Φ, let E^{Φ} denote the existentially quantified variables of Φ, and let U^{Φ} denote the universally quantified variables of Φ. When $u \in V^{\Phi}$ is a variable of Φ, we use $V^{\Phi}_{<u}$ to denote the variables coming strictly before u in the quantifier prefix of Φ, and we use $V^{\Phi}_{\leq u}$ to denote the variables coming before u (including u) in the quantifier prefix of Φ. When S is a subset of V^{Φ}, we say that S is an *initial segment* of Φ if $S = \emptyset$ or $S = V^{\Phi}_{\leq u}$ for a variable $u \in V^{\Phi}$.

Let us intuitively think of an instance of the QCSP as a game between two players: a *universal player* that sets the universally quantified variables, and an *existential player* that sets the existentially quantified variables. The existential player wants to satisfy all of the constraints. We may formalize the notion of a strategy for the existential player in the following way.

A *strategy* for a quantified constraint formula Φ is a sequence of partial functions $\sigma = \{\sigma_x : [V^{\Phi}_{<x} \to D] \to D\}_{x \in E^{\Phi}}$. The intuition behind this definition is that the function σ_x of a strategy describes how to set the variable x given a setting to all of the previous variables. We say that an assignment f to an initial segment of Φ is *consistent* with σ if for every existentially quantified variable x in the domain of f, it holds that $\sigma_x(f|_{V^{\Phi}_{<x}})$ is defined and is equal to $f(x)$. Intuitively, f is consistent with σ if it could have been reached in a play of the game under σ.

A *playspace* for a quantified constraint formula Φ is a set of mappings $\mathcal{A} \subseteq [V^{\Phi} \to D]$. We will often be interested in restrictions of a playspace $\mathcal{A}$ of the form $\mathcal{A}|_{V^{\Phi}_{<u}}$ or $\mathcal{A}|_{V^{\Phi}_{\leq u}}$; we will use the notation $\mathcal{A}\langle <u \rangle$ and $\mathcal{A}\langle \leq u \rangle$ for these restrictions, respectively. The quantified constraint formula Φ will always be clear from the context. Likewise, for a function f defined on a subset of V^{Φ}, we will use the notation $f\langle <u \rangle$ and $f\langle \leq u \rangle$ for the restrictions $f|_{V^{\Phi}_{<u}}$ and $f|_{V^{\Phi}_{\leq u}}$, respectively.

Intuitively, a playspace will be used to describe a restriction on the actions of the universal player: an existential strategy will be a winning strategy for a playspace as long as it can properly respond to all settings of variables that fall into the playspace. We formalize this in the following way.

Let $\mathcal{A}$ be a playspace for a quantified constraint formula Φ, and let σ be a strategy for the same formula Φ. We say that σ is a *winning* strategy for $\mathcal{A}$ if the following two conditions hold:

- for every variable $x \in E^{\Phi}$ and every assignment $f \in \mathcal{A}\langle <x \rangle$, if f is consistent with σ, then $\sigma_x(f)$ is defined and $f[x \to \sigma_x(f)] \in \mathcal{A}\langle \leq x \rangle$, and
- every assignment $f \in \mathcal{A}$ consistent with σ satisfies the constraints of Φ.

We call a playspace *winnable* if there exists a winning strategy for it.

Let us say that a playspace $\mathcal{A}$ (for a quantified constraint formula Φ) is $\forall$-*free* ($\exists$-*free*) if for every universally (existentially) quantified variable $u \in V^{\Phi}$, every domain element $d \in D$, and every function $f \in \mathcal{A}\langle <u \rangle$, the function $f[u \to d]$

is contained in $\mathcal{A}\langle \leq u \rangle$. As a simple example illustrating these notions, observe that for any quantified constraint formula Φ, the playspace $[V^{\Phi} \to D]$ is both $\forall$-free and $\exists$-free. The notion for $\forall$-freeness yields a characterization of truth for quantified constraint formulas.

Proposition 1. *Let Φ be a quantified constraint formula Φ. The following are equivalent:*

1. *Φ is true.*
2. *The $\forall$-free playspace $[V^{\Phi} \to D]$ has a winning strategy.*
3. *There exists a $\forall$-free playspace for Φ having a winning strategy.*

Having given the basic terminology for collapsibility, we now proceed to develop the technique itself. The following is an outline of the technique. What we aim to show is that for certain templates Γ, an arbitrary instance Φ of $\mathsf{QCSP}(\Gamma)$ is truth-equivalent to (the conjunction of) a collection of "simpler" QCSP instances. These simpler instances will always have the property that the truth of the original instance Φ readily implies the truth of the simpler instances; what is non-trivial is to show that the truth of all of the simpler instances implies the truth of the original instance. We will be able to establish this implication in the following way. First, we will translate the truth of the simpler instances into winnability results on playspaces (for the original instance Φ). Then, we will make use of two tools (to be developed here) that allow us to infer the winnability of larger playspaces based on the winnability of smaller playspaces and the polymorphisms of Φ. These tools will let us demonstrate the winnability of a $\forall$-free playspace, which then implies the truth of Φ by Proposition 1.

We now turn to give the two key tools which allow us to "enlargen" playspaces while still preserving winnability. To illustrate the use of these tools, we will use a running example which will fully develop a collapsibility proof.

Example 2. As a running example for this section, we consider *positive equality constraint languages*. Positive equality constraint languages are equality constraint languages where every relation is definable by a positive combination of atoms of the form $x = y$, that is, definable using such atoms and the boolean connectives $\{\vee, \wedge\}$. A simple example of a positive equality constraint language is $\Gamma = (\mathbb{N}, S)$, where S is the relation

$$S = \{(w, x, y, z) \in \mathbb{N}^4 : (w = x) \vee (y = z)\}.$$

Any equality-definable relation R, viewed as the union of equivalence relations, can be verified to have the following closure property: every equivalence relation ρ' obtainable from an equivalence relation ρ from R by combining two equivalence classes into one is also contained in R. In fact, from this observation, it is not difficult to see that a positive equality constraint language has all unary functions as polymorphisms. (Indeed, the property of having all unary functions as polymorphisms is also sufficient for an equality constraint language to be a positive equality constraint language, and hence yields an algebraic characterization of positive equality constraint languages.)

We will show that, for any positive equality constraint language Γ, the problem $\mathsf{QCSP}(\Gamma)$ reduces to $\mathsf{CSP}(\Gamma \cup \{\neq\})$. In particular, for an instance

$$\Phi = Q_1 v_1 \dots Q_n v_n \mathcal{C}$$

of $\mathsf{QCSP}(\Gamma)$, we define the *collapsing of* Φ to be the $\mathsf{CSP}(\Gamma \cup \{\neq\})$ instance

$$\Phi' = \exists v_1 \dots \exists v_n (\mathcal{C} \wedge \bigwedge \{v_i \neq v_j : i < j, Q_j = \forall\}).$$

That is, the collapsing of Φ is obtained from Φ by adding constraints asserting that each universal variable y is different from all variables coming before y, and then changing all quantifiers to existential. We will show that an instance Φ of $\mathsf{QCSP}(\Gamma)$ is true *if and only if* its collapsing is true. This gives a reduction from a problem whose most obvious complexity upper bound is PSPACE, to a problem in NP. The inclusion of this problem in NP has been previously shown by Kozen [19]; we have elected it as our running example as we believe it allows us to nicely illustrate our technique. Note that our reduction is tight in that there are known NP-hard positive equality constraint languages [3]. (The existence of such NP-hard constraint languages also implies that one cannot hope for a reduction from $\mathsf{QCSP}(\Gamma)$ to $\mathsf{CSP}(\Gamma)$ which does not "augment the template", since for positive equality constraint languages Γ, the problem $\mathsf{CSP}(\Gamma)$ is known to be polynomial-time tractable [4].)

It is readily seen that if an instance Φ is true, then its collapsing Φ' is true. The difficulty in justifying this reduction, then, is in showing that if a collapsing Φ' is true, then the original instance Φ is true. Our first step in showing this is to simply view the truth of Φ' as a winnability result on a playspace. Let $a : \{v_1, \dots, v_n\} \to D$ be an assignment satisfying the constraints of Φ'. Clearly, the playspace $\{a\}$ is winnable, via the strategy $\sigma = \{\sigma_x\}_{x \in E^\Phi}$ defined by $\sigma_x(a|_{V^{\Phi}<x}) = a(x)$. We will use the winnability of this playspace to derive the winnability of larger and larger playspaces, ultimately showing the winnability of the largest playspace $[V^\Phi \to D]$, and hence the truth of the formula (by Proposition 1). $\square$

The following lemma allows one to add, to a winnable playspace, tuples from the orbits induced by the tuples already in the playspace, while maintaining the property of winnability.

Lemma 3. *(Orbit Lemma) Let $\mathcal{A}$ be a winnable playspace for a quantified constraint formula Φ over template Γ. Let $y \in U^\Phi$ be a universally quantified variable. There exists a winnable playspace $\mathcal{A}'$ such that the following hold:*

- *for each $t \in \mathcal{A}\langle \leq y \rangle$ and Γ-automorphism σ that fixes every point $\{t(u) : u \in \mathcal{A}\langle < y \rangle\}$, $\sigma(t)$ is in $\mathcal{A}'\langle \leq y \rangle$. Note that here, $\sigma(t)$ is equal to t at all points except (possibly) y.*
- *$\mathcal{A} \subseteq \mathcal{A}' \subseteq \{\tau(t) : t \in \mathcal{A}, \tau$ is a Γ-automorphism$\}$.*
- *$\mathcal{A}\langle < y \rangle = \mathcal{A}'\langle < y \rangle$.*

Proof (idea). Let F be the set of all functions of the form $\sigma(t)$ satisfying the conditions of the first property, that is, t is in $\mathcal{A}\langle \leq y \rangle$ and σ is a Γ-automorphism that fixes every point $\{t(u) : u \in \mathcal{A}\langle <y \rangle\}$. For each element $f \in F \setminus \mathcal{A}\langle \leq y \rangle$, define σ_f and t_f to be such mappings so that $f = \sigma_f(t_f)$. We define $\mathcal{A}'$ to be

$$\mathcal{A} \cup \{\sigma_f(e) : f \in F \setminus \mathcal{A}\langle \leq y \rangle, e \in \mathcal{A}, e\langle \leq y \rangle = t_f\}.$$

Let $\{\rho_x\}$ be a winning strategy for $\mathcal{A}$. We assume without loss of generality that the partial functions ρ_x are only defined on functions $f \in \mathcal{A}\langle <x \rangle$ that are consistent with ρ. We need to extend the ρ_x so that they handle extensions of the functions $f \in F \setminus \mathcal{A}\langle \leq y \rangle$. When g is an extension of such a f, we define $\rho'_x(g)$ as $\sigma_f(\rho_x(\sigma_f^{-1}(g)))$. That is, we translate g back by σ_f and look at the response by ρ_x, and apply σ_f to that response to obtain our response. It is straightforward to verify that the $\{\rho'_x\}$ are a winning strategy for $\mathcal{A}'$. □

Example 4. We continue the discussion of positive equality constraint languages, our running example. We have established the winnability of a size-one playspace $\{a\}$, where for all universally quantified variables y, the value $a(y)$ is different from $a(v)$ for all variables v coming before y in the quantifier prefix. Our goal is to infer the winnability of the largest playspace $[V^{\Phi} \to D]$, using the winnability of this playspace.

Let us say that a playspace $\mathcal{A}$ is *$\neq$-free* if for every universally quantified variable $y \in V^{\Phi}$, every function $f \in \mathcal{A}\langle <y \rangle$, and every value $d \in D$ distinct from all values in $\{f(u) : u \in V^{\Phi}_{<y}\}$, the function $f[y \to d]$ is contained in $\mathcal{A}\langle \leq y \rangle$. Assuming that our original instance Φ contained at least one universally quantified variable y, our playspace $\{a\}$ is *not* $\neq$-free: there is only one extension of $a\langle <y \rangle$ in $\{a\}\langle \leq y \rangle$, namely, $a\langle \leq y \rangle$. However, using the Orbit Lemma, we can expand $\{a\}$ into a $\neq$-free playspace, as follows.

Let y_1 be the first universally quantified variable of Φ. Applying the Orbit Lemma to the playspace $\mathcal{A} = \{a\}$ and variable $y = y_1$, we obtain a playspace $\mathcal{A}_1$ that satisfies the $\neq$-freeness condition at y_1. We demonstrate this as follows. If f is a function in $\mathcal{A}_1\langle <y_1 \rangle$, we have $f \in \mathcal{A}\langle <y_1 \rangle$, since the Orbit Lemma provides $\mathcal{A}\langle <y \rangle = \mathcal{A}'\langle <y \rangle$. Let $h = f[y_1 \to d]$ be any extension of f where d is distinct from all values in the image of f. We want to show that h is contained in $\mathcal{A}_1\langle \leq y_1 \rangle$. We know that there exists an extension $f' = f[y_1 \to d']$ of f such that d' is different from all values in the image of f. (This is because $f \in \mathcal{A}\langle <y_1 \rangle = \{a\}\langle <y_1 \rangle$, and the function a assigns y_1 to a value different from all values assigned to preceding variables.) Let σ be a permutation on D (that is, a Γ-automorphism) that fixes all points in the image of f, but maps d' to d. The Orbit Lemma provides that $\sigma(f') = h$ is in $\mathcal{A}_1\langle \leq y \rangle$. Repeatedly applying the Orbit Lemma to the universally quantified variables $y_1, y_2, \ldots$ of Φ, we obtain an increasing sequence of winnable playspaces $\mathcal{A}_1, \mathcal{A}_2, \ldots$ whose last member is $\neq$-free.

Note that the Orbit Lemma provides, for each i,

$$\mathcal{A}_{i+1} \subseteq \{\tau(t) : t \in \mathcal{A}_i, \tau \text{ is a } \Gamma\text{-automorphism}\}$$

and hence, for each i,

$$\mathcal{A}_i \subseteq \{\tau(t) : t \in \mathcal{A}, \tau \text{ is a } \Gamma\text{-automorphism}\}.$$

From this, we can see that each $\mathcal{A}_i$ has the property that for any universally quantified variable y_j and for any function $f \in \mathcal{A}_i\langle <y_j \rangle$, any extension $f[y_j \to d]$ of f in $\mathcal{A}_i\langle \leq y_j \rangle$ has d distinct from all values in the image of f; this is because $\mathcal{A}$ has this property, and this property is preserved by adding, to a playspace, permutations of functions already in the playspace.

Summarizing, we have shown that the winnability of the size-one playspace from Example 2 implies the winnability of a $\neq$-free playspace. $\square$

The next theorem allows us to, roughly speaking, use a polymorphism $g : D^k \to D$ of Φ to compose together k winnable playspaces to derive another winnable playspace.

Let $g : D^k \to D$ be an operation. Let $\mathcal{A}, \mathcal{B}_1, \ldots, \mathcal{B}_k$ be playspaces for a quantified constraint formula Φ. We say that $\mathcal{A}$ is *g-composable* from $(\mathcal{B}_1, \ldots, \mathcal{B}_k)$ if for all universally quantified variables $y \in U^\Phi$, the following holds: if $t \in \mathcal{A}\langle <y \rangle$ and $t_1 \in \mathcal{B}_1\langle <y \rangle$, $\ldots$, $t_k \in \mathcal{B}_k\langle <y \rangle$ are such that $t = g(t_1, \ldots, t_k)$ pointwise, and $d \in D$ is a value such that $t[y \to d] \in \mathcal{A}\langle \leq y \rangle$, then there exist $d_1, \ldots, d_k \in D$ such that $d = g(d_1, \ldots, d_k)$ and $t_1[y \to d_1] \in \mathcal{B}_1\langle \leq y \rangle$, $\ldots$, $t_k[y \to d_k] \in \mathcal{B}_k\langle \leq y \rangle$.

Theorem 5. *Let Φ be a quantified constraint formula, and assume that $g : D^k \to D$ is a polymorphism of all relations in Φ. Assume that $\mathcal{A}, \mathcal{B}_1, \ldots, \mathcal{B}_k$ are playspaces such that $\mathcal{A}$ is $\exists$-free and g-composable from $(\mathcal{B}_1, \ldots, \mathcal{B}_k)$. If each of the playspaces $\mathcal{B}_1, \ldots, \mathcal{B}_k$ is winnable, then $\mathcal{A}$ is winnable.*

Theorem 5 was inspired by machinery developed for finite-domain QCSPs presented in [10, Chapter 4]. Before giving the proof, we give an example application that allows us to conclude our running example.

Example 6. For a QCSP instance Φ over a positive equality constraint language Γ, we have shown, in Examples 2 and 4, the winnability of a $\neq$-free playspace $\mathcal{A}_{\neq}$ based on the truth of the collapsing Φ' of Φ; the collapsing Φ' is a CSP instance (over an equality constraint language). We now complete the justification of our reduction by showing that the winnability of this $\neq$-free playspace implies the winnability of the "full" playspace $[V^\Phi \to D]$.

Let $g : D \to D$ be a surjective unary function such that $g^{-1}(d)$ is of infinite size for every $d \in D$, that is, every point $d \in D$ in the image of g is hit by infinitely many domain points. As noted in Example 2, the function g is a polymorphism of Γ. To show the winnability of the playspace $[V^\Phi \to D]$, we show that it is g-composable from $\mathcal{A}_{\neq}$, from which its winnability follows by appeal to Theorem 5.

Why is the playspace $[V^\Phi \to D]$ g-composable from $\mathcal{A}_{\neq}$? Let $y \in U^\Phi$ be a universally quantified variable, let $t \in \mathcal{A}\langle <y \rangle$, let $t' \in \mathcal{A}_{\neq}\langle <y \rangle$ and suppose that $t = g(t')$ pointwise. It suffices to show that for any value $d \in D$, there exists $d' \in D$ such that $d = g(d')$ and $t'[y \to d'] \in \mathcal{A}_{\neq}$. This holds: one can pick d' to be any point in $g^{-1}(d) \setminus \mathsf{image}(t')$. This set is non-empty as it is the subtraction of a finite set from an infinite set, and for any such d' we have $t'[y \to d'] \in \mathcal{A}_{\neq}\langle \leq y \rangle$ by the $\neq$-freeness of $\mathcal{A}_{\neq}$. $\square$

Proof (Theorem 5). For each $i \in [k]$, let σ^i be a winning strategy for the playspace $\mathcal{B}_i$. We define a sequence of mappings $\sigma = \{\sigma_x\}_{x \in E^{\Phi}}$ that constitutes a winning strategy for $\mathcal{A}$. We consider each initial segment one by one, in order of increasing size. After the initial segment S has been considered, we will have defined mappings $\{\sigma_x\}_{x \in E^{\Phi} \cap S}$ having the following properties:

(a) if $S = V^{\Phi}|_{\leq x}$ for an existentially quantified variable x, then for any $f \in \mathcal{A}\langle <x \rangle$ consistent with σ, $\sigma_x(f)$ is defined and $f[x \to \sigma_x(f)] \in \mathcal{A}|_S$.
(b) if $f \in \mathcal{A}|_S$ is consistent with σ, then there exist $f_1 \in \mathcal{B}_1|_S$, …, $f_k \in \mathcal{B}_k|_S$ such that $f = g(f_1, \ldots, f_k)$ pointwise and f_i is consistent with σ^i for all $i \in [k]$.

This suffices, since after the initial segment $S = V^{\Phi}$ has been considered, the sequence of mappings $\{\sigma_x\}$ constitute a winning strategy. The first requirement in the definition of a winning strategy holds because property (a) holds for all possible initial segments S. The second requirement in the definition of a winning strategy holds: by property (b), any assignment $f \in \mathcal{A}|_{V^{\Phi}}$ consistent with σ is equal to g applied point-wise to assignments $f_1 \in \mathcal{B}_1|_{V^{\Phi}}$, …, $f_k \in \mathcal{B}_k|_{V^{\Phi}}$ that are consistent with $\sigma^1, \ldots, \sigma^k$, respectively; since the σ^i are winning strategies, each f_i satisfies the constraints of Φ, and since g is a polymorphism of the relations of Φ, f satisfies the constraints of Φ.

We now give the construction.

Let $S' = V^{\Phi}_{\leq u}$ be an initial segment of size $|S'| \geq 1$, and let $S = V^{\Phi}_{<u}$ be the initial segment of size $|S'|-1$. We may assume by induction that the construction has been performed for S. To perform the construction for S', we consider two cases depending on the quantifier of the variable u.

Case 1: u is an $\exists$-quantified variable. We consider each mapping $f \in \mathcal{A}|_S$. If f is not consistent with σ, then we leave $\sigma_u(f)$ undefined. If f is consistent with σ, then in order to satisfy property (a), we need to define $\sigma_u(f)$. Since property (b) holds on S, there exist the described mappings $f_1 \in \mathcal{B}_1|_S$, …, $f_k \in \mathcal{B}_k|_S$ with $f = g(f_1, \ldots, f_k)$ pointwise and with f_i consistent with σ^i for all $i \in [k]$. Since, for each $i \in [k]$, the σ^i are winning strategies, there is an extension $f'_i \in \mathcal{B}_i|_{S'}$ of f consistent with σ^i. We define $\sigma_u(f)$ as $g(f'_1(u), \ldots, f'_k(u))$. The mapping $f' = f[u \to \sigma_u(f)]$ is in $\mathcal{A}|_{S'}$ by the $\exists$-freeness of $\mathcal{A}$. Now, the mapping f' is consistent with σ, so we need to verify that property (b) holds on f'. It is straightforward to verify that the mappings $f'_1, \ldots, f'_k$ serve at witnesses.

Case 2: u is a $\forall$-quantified variable. Clearly, property (a) is trivially satisfied for S', so we need only consider property (b). Suppose that $f' \in \mathcal{A}|_{S'}$ is consistent with σ. We want to show the existence of the described mappings $f'_1, \ldots, f'_k$. Let $f = f'|_S$. Since property (b) holds for the initial segment S, we know that there exist $f_1 \in \mathcal{B}_1\langle <u \rangle, \ldots, f_k \in \mathcal{B}_k\langle <u \rangle$ such that $f = g(f_1, \ldots, f_k)$ pointwise and f_i is consistent with σ^i for all $i \in [k]$. By the definition of g-composable, there exist extensions $f'_1, \ldots, f'_k$ of $f_1, \ldots, f_k$, respectively, to S', satisfying the conditions of property (b). □

4 Applications

In the previous section, we developed some tools for giving collapsibility proofs, and illustrated their use on positive equality constraint languages. We showed that for any positive equality constraint language Γ, the problem $\mathsf{QCSP}(\Gamma)$ is in NP. In this section, we give further applications of our technique.

4.1 Max-closed constraints

We consider temporal constraint languages that are closed under the binary operation $\max : \mathbb{Q} \times \mathbb{Q} \to \mathbb{Q}$ that returns the maximum of its two arguments. We will demonstrate the following theorem.

Theorem 7. *Let Γ be a temporal constraint language having the* max *operation as polymorphism. The problem $\mathsf{QCSP}(\Gamma)$ is in NP.*

Example 8. Consider the temporal constraint language $(\mathbb{Q}, R)$ where R is the relation $\{(x, y, z) \in \mathbb{Q}^3 : x < y \text{ or } x < z\}$. This constraint language has the max operation as polymorphism: suppose $(a, b, c), (a', b', c') \in R$. We want to show that $(\max(a, a'), \max(b, b'), \max(c, c')) \in R$. Let us assume without loss of generality that $a > a'$. We know that either $a < b$ or $a < c$. If $a < b$, then $a < \max(b, b')$ and we have $\max(a, a') < \max(b, b')$. If $a < c$, then $a < \max(c, c')$ and we have $\max(a, a') < \max(c, c')$. $\square$

We now prove this theorem. Let Φ be an instance of $\mathsf{QCSP}(\Gamma)$ for a max-closed template Γ. As in the collapsibility proof for positive equality constraint languages, we will show a reduction to a CSP. Whereas in the case of positive languages we gave a direct reduction to a CSP, here, we give a reduction to a conjunction of QCSP instances, each of which has one universally quantified variable; we argue that this ensemble can be formulated as a CSP.

Denote Φ as $Q_1 v_1 \dots Q_n v_n \mathcal{C}$. (We assume that Φ has at least one universally quantified variable, otherwise, it is an instance of $\mathsf{CSP}(\Gamma)$.) For a universally quantified variable $v_i \in U^{\Phi}$, we define the v_k*-collapsing* of Φ to be the QCSP instance

$$\Phi' = \exists v_1 \dots \exists v_{k-1} \forall v_k \exists v_{k+1} \dots \exists v_n (\mathcal{C} \wedge \bigwedge \{v_i > v_j : i < j, j \neq k, Q_j = \forall\}).$$

That is, the v_k-collapsing of Φ is obtained from Φ by adding constraints asserting that each universal variable y (other than v_k) is less than all variables coming before it, and changing all universal quantifiers to existential except for that of v_k. It is readily verifiable that if the original $\mathsf{QCSP}(\Gamma)$ instance Φ was true, then all of its y-collapsings (with $y \in U^{\Phi}$) are also true. We show that the converse holds. This suffices to place $\mathsf{QCSP}(\Gamma)$ in NP, by the following lemma.

Lemma 9. *Let $B \subseteq \mathbb{Q}^3$ be the "different-implies-between" relation defined by $B = \{(x, y, z) \in \mathbb{Q}^3 : (x \neq z) \to ((x < y < z) \vee (x > y > z))\}$. Let Γ' be the expansion of a temporal constraint language Γ with B and $<$. Given an instance*

Φ of $\mathsf{QCSP}(\Gamma)$*, there exists an instance Φ' of* $\mathsf{CSP}(\Gamma')$ *that is true if and only if Φ is true. For every constant k, the mapping $\Phi \to \Phi'$ can be computed in polynomial time on those formulas Φ with $|U^{\Phi}| \leq k$.*

Proof (idea). We demonstrate how to translate Φ to Φ' iteratively by a process that removes one universally quantified variable at a time. Let y denote the first universally quantified variable of Φ. We may assume that Φ is of the form $\exists x_1 \ldots \exists x_n \forall y \phi$. We claim that there are polynomially (in n) many formulas $\psi_1, \ldots, \psi_m$ with free variables $\{x_1, \ldots, x_n, y\}$ such that:

- for any assignment f to the variables $\{x_1, \ldots, x_n, y\}$, there exists a ψ_i such that the only extension of $f|_{\{x_1,\ldots,x_n\}}$ satisfying ψ_i is in the same orbit as f, that is, is of the form $\sigma(f)$ for an automorphism f, and
- for any assignment f to the variables $\{x_1, \ldots, x_n\}$ and for all ψ_i, there is an extension of f under which ψ_i is true.

This suffices, since Φ can then be rewritten as $\exists x_1 \ldots \exists x_n \bigwedge_i (\exists x'_i(\psi_i(x_1, \ldots, x_n, x'_i) \wedge \phi))$. The ψ_i are as follows:

- For each choice of two variables $x, x' \in \{x_1, \ldots, x_n\}$, there is a ψ_i with $\psi_i(x_1, \ldots, x_n, y)$ defined as $B(x, y, x')$.
- For each choice of variable $x \in \{x_1, \ldots, x_n\}$, there is a ψ_i with $\psi_i(x_1, \ldots, x_n, y)$ defined as $x = y$.
- There is a ψ_i with $\psi_i(x_1, \ldots, x_n, y)$ defined as $\bigwedge_{i \in [n]}(x_i < y)$.
- There is a ψ_i with $\psi_i(x_1, \ldots, x_n, y)$ defined as $\bigwedge_{i \in [n]}(x_i > y)$.

The total number of formulas ψ_i is $\binom{n}{2} + n + 2$ which is polynomial in n. $\square$

Lemma 9 can be viewed as a strong version of the well-known quantifier elimination property for temporal constraint languages.

We want to show that if all y-collapsings of an instance Φ are true, then Φ itself is true. How will we do this? We will first translate the truth of each y-collapsing into a winnability result on a playspace $\mathcal{A}_y$ for Φ. We will then show that each of these playspaces $\mathcal{A}_y$ can be expanded into a playspace $\mathcal{A}'_y$ that obeys a "freeness" condition but is still winnable. We then compose together the playspaces $\mathcal{A}'_y$ using Theorem 5 to derive the winnability of the full playspace $[V^{\Phi} \to \mathbb{Q}]$.

We translate the truth of the y-collapsings of Φ into winnability results on playspaces, as follows. Let us say that a playspace $\mathcal{A}$ (for Φ) is *$\forall$-free at $z \in U^{\Phi}$* if for every assignment $f \in \mathcal{A}\langle<z\rangle$, and every $d \in D$, the function $f[z \to d]$ is contained in $\mathcal{A}\langle\leq z\rangle$. For each $y \in U^{\Phi}$, it is readily verified that the truth of the y-collapsing of Φ implies the winnability of a playspace $\mathcal{A}_y$ that is $\forall$-free at y, and where for all $t \in \mathcal{A}_y$ it holds that $t(a) > t(b)$ if $b \in U^{\Phi} \setminus \{y\}$, $a \in V^{\Phi}$, and a comes before b in the quantifier prefix.

Let $S \subseteq U^{\Phi}$ be a set of universally quantified variables. We define a playspace $\mathcal{A}$ for Φ to be *$(S, <)$-free* if:

- for every variable $y \in S$, $\mathcal{A}$ is $\forall$-free at y, and

– for every variable $y \in U^{\Phi} \setminus S$, and every assignment $f \in \mathcal{A}\langle <y \rangle$, there exists an interval $(-\infty, d_y]$ such that for every $d \in (-\infty, d_y]$, the function $f[y \to d]$ is contained in $\mathcal{A}\langle \leq y \rangle$.

Our playspaces $\mathcal{A}_y$ are not $(\{y\}, <)$-free, but via repeated application of the Orbit Lemma, from each playspace $\mathcal{A}_y$ we may obtain a winnable playspace $\mathcal{A}'_y$ that is $(\{y\}, <)$-free.

We prove by induction, on the size of S, that there is a winnable playspace $\mathcal{A}'_S$ that is $(S, <)$-free (for all $S \subseteq U^{\Phi}$). This suffices to show the winnability of a $(U^{\Phi}, <)$-free playspace, which is $\forall$-free, implying the truth of Φ by Proposition 1. Suppose $k \geq 1$. By induction, we assume that we have constructed our $\mathcal{A}'_S$ for $|S| \leq k$. Let $S' \subseteq U^{\Phi}$ be of size $|S'| = k + 1$. We want to show the winnability of a $(S', <)$-free playspace. Pick any element $s_0 \in S'$ and set $S = S' \setminus \{s_0\}$. Suppose that $\mathcal{A}'_{s_0}$ is $(\{s_0\}, <)$-free with respect to $\{d_y\}_{y \in U^{\Phi} \setminus \{s_0\}}$, and that $\mathcal{A}'_S$ is $(S, <)$-free with respect to $\{e_y\}_{y \in U^{\Phi} \setminus S}$. We show the winnability of the playspace $\mathcal{A}'_{S'}$ that is $(S', <)$-free with respect to $\{\min(d_y, e_y)\}_{y \in U^{\Phi} \setminus S'}$, and also $\exists$-free. In particular, we prove that $\mathcal{A}_{S'}$ is max-composable from $(\mathcal{A}'_{s_0}, \mathcal{A}'_S)$. The winnability of $\mathcal{A}_{S'}$ then follows from Theorem 5. Let $y \in U^{\Phi}$ and consider $t \in \mathcal{A}'_{S'}\langle <y \rangle$, $t_{s_0} \in \mathcal{A}_{s_0}\langle <y \rangle$, $t_S \in \mathcal{A}_S\langle <y \rangle$ such that $t = \max(t_{s_0}, t_S)$ pointwise. Let $d \in \mathbb{Q}$ be a value such that $t[y \to d] \in \mathcal{A}'_{S'}$. We want to find values d_1, d_2 such that $d = \max(d_1, d_2)$ and $t_{s_0}[y \to d_1] \in \mathcal{A}_{s_0}\langle \leq y \rangle$, and $t_S[y \to d_2] \in \mathcal{A}_S\langle \leq y \rangle$. We split into cases.

– $y = s_0$: we select $d_1 = d$ and d_2 to be a value such that $d_2 \leq d$ and $d_2 \leq d_y$. The first inequality guarantees that $d = \max(d_1, d_2)$ and the second guarantees that $t_{s_0}[y \to d_1] \in \mathcal{A}_{s_0}\langle \leq y \rangle$.
– $y \in S$: we select $d_2 = d$ and d_1 to be a value such that $d_1 \leq d$ and $d_1 \leq e_y$. This case is similar to the previous one, except we use the $\forall$-freeness of $\mathcal{A}_S$ at y, whereas in the previous case, we used the $\forall$-freeness of $\mathcal{A}_{s_0}$.
– $y \in U^{\Phi} \setminus S'$: we select $d_1 = d$ and $d_2 = d$.

4.2 Near-unanimity operations

A *near-unanimity operation* is an operation $f : D^k \to D$ of arity $k \geq 3$ satisfying the identities $f(y, x, \ldots, x) = f(x, y, x, \ldots, x) = \cdots = f(x, \ldots, x, y)$ for all $x, y \in D$. Near-unanimity operations have been studied in the finite case in [17, 10]. We show that, for a constraint language Γ having a near-unanimity operation as polymorphism, the problem $\mathsf{QCSP}(\Gamma)$ essentially reduces to the problem $\mathsf{CSP}(\Gamma)$.

Theorem 10. *Suppose that Γ is a constraint language having a near-unanimity operation $g : D^k \to D$ as polymorphism. There exists a polynomial-time computable mapping that, given an instance Φ of $\mathsf{QCSP}(\Gamma)$, outputs a set S of instances of $\mathsf{QCSP}(\Gamma)$ such that:*

– *each instance in S has at most $k-1$ universally quantified variables, and*
– *all instances in S are true if and only if the original instance Φ is true.*

Proof. Let Φ be an instance of $\mathsf{QCSP}(\Gamma)$. In this proof, we define a j-collapsing of Φ to be an instance of $\mathsf{QCSP}(\Gamma)$ obtained from Φ by selecting a subset $S \subseteq U^\Phi$ of universally quantified variables of size $|S| = j$, and changing the quantifiers of the variables $U^\Phi \setminus S$ to existential, and adding constraints $\{y = y' : y, y' \in U^\Phi \setminus S\}$ equating all of the variables in $U^\Phi \setminus S$. (Note that these equalities can subsequently be eliminated by renaming and removing variables.)

Clearly, if Φ is true, all of its j-collapsings are true. We show that if the j-collapsings of Φ are true for all $j \leq k-1$, then Φ is true. This is obvious if Φ has $k-1$ or fewer universally quantified variables, so we assume that it has k or more universally quantified variables. It is straightforward to verify that the truth of the j-collapsing of Φ arising from the subset $S \subseteq U^\Phi$ (with $|S| = j$) implies the winnability of a playspace $\mathcal{A}_S$ (for Φ) that is $\forall$-free at all $y \in S$ and such that $f(y') = a$ for all $y \in U^\Phi \setminus S$ for a fixed constant a. (See the proof of Theorem 7 for the definition of $\forall$-*free at* y.)

We prove that for all subsets $S \subseteq U^\Phi$, there is a winnable playspace $\mathcal{A}_S$ (for Φ) that is $\forall$-free at all $y \in S$. This suffices, since then $\mathcal{A}_{U^\Phi}$ is a winnable playspace that is $\forall$-free. We prove this by induction on $|S|$. We have pointed out that this is true when $|S| \leq k-1$, so assume that $|S| \geq k$. Select k distinct elements $s_1, \ldots, s_k \in S$. For each $i \in [k]$, define $S_i = S \setminus \{s_i\}$. We claim that the playspace $\mathcal{A}_S$ that consists of all functions $f : V^\Phi \to D$ such that $f(y) = a$ for all $y \in U^\Phi \setminus S$, is g-composable from $(\mathcal{A}_{S_1}, \ldots, \mathcal{A}_{S_k})$. The result then follows from Theorem 5. We verify this as follows. Let $y \in U^\Phi$ and suppose that $t \in \mathcal{A}_S\langle {<}y \rangle$ and $t_i \in \mathcal{A}_{S_i}\langle {<}y \rangle$ for all $i \in [k]$ are such that $t = g(t_1, \ldots, t_k)$ pointwise, and $d \in D$ is a value such that $t[y \to d] \in \mathcal{A}_S\langle {\leq}y \rangle$. We want to give values $d_1, \ldots, d_k \in D$ such that $d = g(d_1, \ldots, d_k)$ and $t_i[y \to d_i] \in \mathcal{A}_{S_i}\langle {\leq}y \rangle$. We split into cases.

- If $y = s_i$ for some $i \in [k]$, we set $d_i = a$ and $d_j = d$ for all other j, that is, $j \in [k] \setminus \{i\}$.
- If $y \in S \setminus \{s_1, \ldots, s_k\}$, we set $d_i = d$ for all $i \in [k]$.
- If $y \in U^\Phi \setminus S$, we set $d_i = a$ for all $i \in [k]$.

□

The following is an example application of Theorem 10. Define the operation $\mathsf{median} : \mathbb{Q}^3 \to \mathbb{Q}$ to be the operation that returns the median of its three arguments. The operation median is a near-unanimity operation of arity 3.

Theorem 11. *Suppose that Γ is a temporal constraint language having the* median *operation as polymorphism. The problem* $\mathsf{QCSP}(\Gamma)$ *is in* NP.

Proof. We use the reduction of Theorem 10 along with Lemma 9 to obtain a reduction to $\mathsf{CSP}(\Gamma')$ for a temporal constraint language Γ'. □

References

1. Manuel Bodirsky. Constraint satisfaction with infinite domains. Ph.D. thesis, Humboldt-Universität zu Berlin, 2004.

2. Manuel Bodirsky. The core of a countably categorical structure. In Volker Diekert and Bruno Durand, editors, *Proceedings of the 22nd Annual Symposium on Theoretical Aspects of Computer Science (STACS'05), Stuttgart (Germany)*, LNCS 3404, pages 100–110, Springer-Verlag Berlin Heidelberg, 2005.
3. Manuel Bodirsky and Hubie Chen. Quantified equality constraints. Manuscript, 2006.
4. Manuel Bodirsky and Jan Kára. The complexity of equality constraint languages. In *Proceedings of the International Computer Science Symposium in Russia (CSR'06)*, LNCS 3967, pages 114–126, 2006.
5. F. Boerner, A. Bulatov, A. Krokhin, and P. Jeavons. Quantified constraints: Algorithms and complexity. In *Proceedings of CSL'03*, LNCS 2803, pages 58–70, 2003.
6. A. Bulatov and V. Dalmau. A simple algorithm for Mal'tsev constraints. *SIAM J. Comp. (to appear).*
7. A. Bulatov, A. Krokhin, and P. G. Jeavons. Classifying the complexity of constraints using finite algebras. *SIAM Journal on Computing*, 34:720–742, 2005.
8. Andrei Bulatov, Andrei Krokhin, and Peter Jeavons. The complexity of maximal constraint languages. In *Proceedings of STOC'01*, pages 667–674, 2001.
9. Andrei A. Bulatov. A dichotomy theorem for constraints on a three-element set. In *FOCS'02*, pages 649–658, 2002.
10. Hubie Chen. The computational complexity of quantified constraint satisfaction. Ph.D. thesis, Cornell University, August 2004.
11. Hubie Chen. Collapsibility and consistency in quantified constraint satisfaction. In *AAAI*, pages 155–160, 2004.
12. Hubie Chen. Quantified constraint satisfaction, maximal constraint languages, and symmetric polymorphisms. In *STACS*, pages 315–326, 2005.
13. Victor Dalmau. Generalized majority-minority operations are tractable. In *LICS*, pages 438–447, 2005.
14. 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*, 28:57–104, 1999.
15. Georg Gottlob, Gianluigi Greco, and Francesco Scarcello. The complexity of quantified constraint satisfaction problems under structural restrictions. In *IJCAI 2005*, 2005.
16. Wilfrid Hodges. *A shorter model theory.* Cambridge University Press, 1997.
17. Peter Jeavons, David Cohen, and Martin Cooper. Constraints, consistency and closure. *AI*, 101(1-2):251–265, 1998.
18. Ph. G. Kolaitis and M. Y. Vardi. Conjunctive-query containment and constraint satisfaction. In *Proceedings of PODS'98*, pages 205–213, 1998.
19. Dexter Kozen. Positive first-order logic is NP-complete. *IBM Journal of Research and Development*, 25(4):327–332, 1981.
20. Guoqiang Pan and Moshe Vardi. Fixed-parameter hierarchies inside PSPACE. In *LICS 2006.* To appear.
21. Larry J. Stockmeyer and Albert R. Meyer. Word problems requiring exponential time: Preliminary report. In *STOC*, pages 1–9, 1973.