Quantified Constraint Satisfaction and 2-Semilattice Polymorphisms

Hubie Chen

Department of Computer Science
Cornell University
Ithaca, NY 14853, USA
hubes@cs.cornell.edu

Abstract. The quantified constraint satisfaction problem (QCSP) is a natural and useful generalization of the constraint satisfaction problem (CSP) in which both universal and existential quantification of variables is permitted. Because the CSP and QCSP are in general intractable, much effort has been directed towards identifying restricted cases of these problems that are tractable in polynomial time. In this paper, we investigate restricted cases of the QCSP having 2-semilattice polymorphisms. We prove a complete classification of 2-semilattice polymorphisms, demonstrating that each gives rise to a case of the QCSP that is either tractable in polynomial time, or coNP-hard.

1 Introduction

The constraint satisfaction problem (CSP) is widely acknowledged as a convenient framework for modelling search problems. An instance of the CSP consists of a set of variables, a domain, and a set of constraints; each constraint consists of a tuple of variables paired with a relation (over the domain) which contains permitted values for the variable tuple. The question is to decide whether or not there is an assignment mapping each variable to a domain element that satisfies all of the constraints.

All of the variables in a CSP can be thought of as being implicitly existentially quantified. A natural generalization of the CSP is the quantified constraint satisfaction problem (QCSP), where variables may be both existentially and universally quantified. Whereas the CSP concerns deciding the existence of a *static* object, a satisfying assignment, the QCSP concerns deciding the existence of a *dynamic* object: a strategy telling how to set the existentially quantified variables in reaction to an arbitrary setting of the universally quantified variables, so that the constraints are satisfied. The generality of the QCSP framework permits the modelling of a variety of artificial intelligence problems that cannot be expressed using the CSP, for instance, problems from the areas of planning, game playing, and non-monotonic reasoning. Of course, the relatively higher expressiveness of the QCSP comes at the price of higher complexity: whereas the CSP is in general NP-complete, the QCSP is in general complete for the complexity class PSPACE, which is believed to be much larger than NP.

The general intractability of the CSP and QCSP motivates the search for cases of these problems that are tractable in polynomial time. A particularly useful way to restrict the CSP and QCSP in order to obtain tractable cases is to restrict the types of relations that may appear in constraints. Formally, this is done by defining a *constraint language* to be a set of relations and then defining, for each constraint language Γ, the problem $\mathsf{CSP}(\Gamma)$ ($\mathsf{QCSP}(\Gamma)$) to be the restricted version of the CSP (QCSP) where only relations from the set Γ may be present. This form of restriction can capture and place into a unified framework many particular cases of the CSP that have been independently investigated, such as Horn Satisfiability and 2-Satisfiability, as well as their corresponding QCSP generalizations, Quantified Horn Satisfiability and Quantified 2-Satisfiability.

The class of problems $\mathsf{CSP}(\Gamma)$ was first considered by Schaefer; he proved a now classic classification theorem which states that for every constraint language Γ over a two-element domain, the problem $\mathsf{CSP}(\Gamma)$ is either in P or is NP-complete [25]. The non-trivial tractable cases of $\mathsf{CSP}(\Gamma)$ given by this result are the Horn Satisfiability, 2-Satisfiability, and XOR-Satisfiability problems. Over the past decade, many more complexity classification theorems in the spirit of Schaefer's have been established for different variants and generalizations of the CSP (see for example the book [15]), including a classification theorem for the problems $\mathsf{QCSP}(\Gamma)$ in domain size two [15, 16]. This classification theorem demonstrates that, when Γ is a constraint language over a two-element domain, the only tractable problems of the form $\mathsf{QCSP}(\Gamma)$ are Quantified Horn Satisfiability [12], Quantified 2-Satisfiability [1], and Quantified XOR-Satisfiability [15], reflecting exactly the non-trivial tractable constraint languages provided by Schaefer's theorem; for all other constraint languages Γ, the problem $\mathsf{QCSP}(\Gamma)$ is PSPACE-complete.

In recent years, much effort has been directed towards the program of classifying the complexity of $\mathsf{CSP}(\Gamma)$ for all constraint languages Γ over a finite domain of *arbitrary* size. While this appears to be a particularly challenging research problem, impressive progress has been made, including the papers [18, 21, 19, 20, 17, 23, 22, 8, 4–6, 9, 3]. Many of these papers make use of an intimate connection between CSP complexity and universal algebra that has been developed [21, 19, 8]. The central notion used to establish this connection is that of *polymorphism*; a constraint language has an operation as polymorphism, roughly speaking, if each relation of the constraint language satisfies a certain closure property defined in terms of the operation. There are many results in the literature which demonstrate that if a constraint language Γ has a polymorphism of a certain type, then the problem $\mathsf{CSP}(\Gamma)$ is tractable.

Very recently, the study of QCSP complexity based on constraint languages in domains of arbitrary size was initiated [2, 13, 14]. It has been shown that the same polymorphism-based algebraic approach used to study CSP complexity can also be used to study QCSP complexity [2]. In the papers [2, 13, 14], general sufficient conditions for QCSP tractability have been identified, which, as with many of the CSP tractability results, demonstrate that the presence of a certain type of polymorphism guarantees tractability of a constraint language.

In this paper, we continue the study of QCSP complexity by investigating constraint languages that have a *2-semilattice operation* as polymorphism. A 2-semilattice operation is a binary operation $\star$ that satisfies the semilattice identities restricted to two variables, namely, the identities $x \star x = x$ (idempotence), $x \star y = y \star x$ (commutativity), and $(x \star x) \star y = x \star (x \star y)$ (restricted associativity). 2-semilattices constitute a natural generalization of semilattices, one of the initial classes of polymorphisms shown to guarantee CSP tractability [21], and have been shown to imply CSP tractability via a consistency-based algorithm [3]. We prove a full classification of 2-semilattice polymorphisms for QCSPs, showing that some such polymorphisms guarantee tractability, while others do not. We would like to highlight three reasons as to why we believe our study of 2-semilattice polymorphisms in the QCSP setting is interesting.

First, as pointed out previously in [3], 2-semilattice polymorphisms play an important role in the investigation of *maximal constraint languages*, which are constraint languages that can express any relation when augmented with any relation not expressible by the language. Because a constraint language that can express all relations is intractable, maximal constraint languages are the largest constraint languages that could possibly be tractable (in either the CSP or QCSP setting); hence, studying maximal constraint languages allows one to obtain the most general tractability results possible. (It is worth noting here that all of the tractability results identified by Schaefer's theorem apply to maximal constraint languages.) It follows from a theorem of Rosenberg [24] that maximal constraint languages can be classified into five types; for four of these types of constraint languages, QCSP tractability or QCSP intractability can be derived using established results. For the remaining type–constraint languages having a non-projection binary idempotent operation as polymorphism–a complexity classification has not yet been established. The present work constitutes a step towards understanding this remaining type. We mention that in the CSP setting, the tractability of 2-semilattices has been leveraged to give complete complexity classifications of maximal constraint languages for domains of size three and four [7, 3].

Second, our tractability proofs make use of and validate new machinery for proving QCSP tractability that was developed in [14]. In particular, a key idea from [14] that we make use of here is that of *collapsibility*; roughly speaking, a problem $\mathsf{QCSP}(\Gamma)$ is *j-collapsible* if any problem instance can be reduced to deciding the truth of a conjunction of QCSPs, each of which has a constant number of universal quantifiers and is derived from the original instance by collapsing together universally quantified variables. In [14], it was demonstrated that many constraint languages Γ are such that (1) $\mathsf{CSP}(\Gamma)$ is tractable and (2) the problem $\mathsf{QCSP}(\Gamma)$ is j-collapsible; it was also demonstrated that these two properties can be used together to derive the tractability of $\mathsf{QCSP}(\Gamma)$. We give another class of constraint languages having these properties, by showing that QCSP problems having tractable 2-semilattice polymorphisms are 1-collapsible.

This provides further evidence that collapsibility is a fruitful and useful tool for studying QCSP complexity. Moreover, we believe our proof of 1-collapsibility to be the most non-trivial collapsibility proof to date.

Third, although all 2-semilattice polymorphisms are tractable in the CSP setting, 2-semilattice polymorphisms intriguingly yield two modes of behavior in the QCSP setting: some 2-semilattice polymorphisms guarantee tractability, while others do not. This is surprising in light of the fact that for all other types of polymorphisms (of non-trivial constraint languages) that have been investigated, polymorphisms that guarantee CSP tractability also guarantee QCSP tractability. In fact, our results imply the first and only known examples of non-trivial constraint languages that are CSP tractable, but QCSP intractable.[1] The existence of such constraint languages implies that the boundary between tractability and intractability in the QCSP context is genuinely different from the corresponding boundary in the CSP context.

The contents of this paper are as follows. We present the basic terminology and concepts to be used throughout the paper in a preliminaries section (Section 2). We prove a classification theorem which shows that every 2-semilattice polymorphism gives rise to a case of the QCSP that is either tractable in polynomial time, or is coNP-hard; and, we derive some consequences of this theorem (Section 3). We then demonstrate, for the QCSPs having a tractable 2-semilattice polymorphism, a result significantly stronger than mere polynomial time tractability: we show that such QCSPs are 1-collapsible, the strongest possible statement one can show concerning collapsibility (Section 4).

2 Preliminaries

We use $[n]$ to denote the set containing the first n positive integers, that is, $\{1, \ldots, n\}$.

2.1 Quantified constraint satisfaction

Quantified formulas. A *domain* is a nonempty set of finite size. A *tuple* (over domain D) is an element of D^k for some $k \geq 1$, and is said to have arity k. The ith coordinate of a tuple $\overline{t}$ is denoted by t_i. A *relation* (over domain D) is a subset of D^k for some $k \geq 1$, and is said to have arity k.

A *constraint* is an expression of the form $R(\overline{v})$, where R is a relation and $\overline{v}$ is a tuple of variables such that R and $\overline{v}$ have the same arity. A *constraint network* is a finite set of constraints, all of which have relation over the same domain, and is said to be over the variable set V if all of its constraints have variables from V. Throughout, we let D denote the domain of our constraints and constraint networks.

[1] Here, by a non-trivial constraint language, we mean a constraint language that includes each constant as a relation.

Definition 1. *A quantified formula is an expression of the form $Q_1 v_1 \dots Q_n v_n \mathcal{C}$, where each Q_i is a quantifier from the set $\{\forall, \exists\}$, and $\mathcal{C}$ is a constraint network over the variable set $\{v_1, \dots, v_n\}$.*

The quantified formula $Q_1 v_1 \dots Q_n v_n \mathcal{C}$ is said to have $Q_1 v_1 \dots Q_n v_n$ as its *quantifier prefix*. We say that the variable v_i comes (strictly) before the variable v_j if $i \leq j$ ($i < j$). We let V_ϕ, Y_ϕ, and X_ϕ denote the variables, universally quantified variables, and existentially quantified variables of a quantified formula ϕ, respectively; we drop the ϕ subscript when it is understood from the context. We generally assume that the universally quantified variables (of a quantified formula) are denoted $y_1, \dots, y_{|Y|}$, where y_i comes strictly before y_j for $i < j$; similarly, we generally assume that the existentially quantified variables (of a quantified formula) are denoted $x_1, \dots, x_{|X|}$, where x_i comes strictly before x_j for $i < j$. When W is a non-empty subset of the variable set V_ϕ (of a quantified formula ϕ), we use $\mathsf{first}_\phi(W)$ to denote the unique variable in W coming before all of the other variables in W. For a subset W of the variable set V (of a quantified formula) and a variable v of V, we let $W[\leq v]$ ($W[< v]$) denote the set of variables in W coming (strictly) before v.

Strategies and truth. A constraint $R(v_1, \dots, v_k)$ is *satisfied* by an assignment f defined on $\{v_1, \dots, v_k\}$ if $(f(v_1), \dots, f(v_k)) \in R$. A constraint network $\mathcal{C}$ (over the variable set V) is *satisfied* by an assignment $f : V \to D$ if each constraint C in $\mathcal{C}$ is satisfied by f.

Definition 2. *A* strategy *σ is a sequence of mappings*

$$\{\sigma_i : D^{\mathsf{rank}(\sigma_i)} \to D\}_{i \in [n]}$$

where the ith mapping σ_i is a function over D having rank $\mathsf{rank}(\sigma_i) \geq 0$.

Note that when σ_i is a mapping of a strategy such that $\mathsf{rank}(\sigma_i) = 0$, we consider σ_i to be a constant, that is, an element of D.

A *strategy for the quantified formula ϕ* is a strategy $\sigma_1, \dots, \sigma_{|X_\phi|}$ where for $i \in [|X_\phi|]$, the mapping σ_i has rank $|Y_\phi[< x_i]|$. An *adversary for the quantified formula ϕ* is a function $\tau : Y_\phi \to D$. When σ is a strategy and τ is an adversary for the quantified formula ϕ, the *outcome* of σ and τ, denoted by $\mathsf{outcome}(\sigma, \tau) : V_\phi \to D$, is the assignment defined by $\mathsf{outcome}(\sigma, \tau)(x_i) = \sigma_i(\tau(y_1), \dots, \tau(y_{|Y[<x_i]|}))$ for $x_i \in X_\phi$ and $\mathsf{outcome}(\sigma, \tau)(y_i) = \tau(y_i)$ for $y_i \in Y_\phi$.

A strategy σ for the quantified formula ϕ is said to be *winning* if for all adversaries τ for ϕ, the assignment $\mathsf{outcome}(\sigma, \tau) : V_\phi \to D$ satisfies the constraint network $\mathcal{C}$ of ϕ. We consider a quantified formula ϕ to be *true* if there exists a winning strategy for ϕ. (This is one of many equivalent ways to define truth of a quantified formula.)

Problem formulation. In this paper, we focus on restricted versions of the QCSP where all relations must come from a *constraint language*. A *constraint language* is defined to be a set of relations (not necessarily of the same arity), all of which are over the same domain.

Definition 3. *Let Γ be a constraint language. The* $\mathsf{QCSP}(\Gamma)$ *problem is to decide, given as input a quantified formula ϕ with constraints having relations from Γ, whether or not ϕ is true.*

We define the $\mathsf{CSP}(\Gamma)$ problem to be the restriction of the $\mathsf{QCSP}(\Gamma)$ problem to instances where all quantifiers are existential.

Polymorphisms. A powerful algebraic theory for studying the complexity of $\mathsf{CSP}(\Gamma)$ problems was introduced in [21, 19]; it can also be applied to study the complexity of $\mathsf{QCSP}(\Gamma)$ problems [2]. (We refer the reader to those papers for more information.) An operation $\mu : D^k \to D$ is a *polymorphism* of a relation $R \subseteq D^m$ if for all tuples $\overline{t_1}, \ldots, \overline{t_k} \in R$, the tuple

$$(\mu(t_{11}, \ldots, t_{k1}), \ldots, \mu(t_{1m}, \ldots, t_{km}))$$

is in R. An operation $\mu : D^k \to D$ is a polymorphism of a constraint language Γ if μ is a polymorphism of all relations $R \in \Gamma$. It has been shown that the complexity of $\mathsf{CSP}(\Gamma)$ (and $\mathsf{QCSP}(\Gamma)$) is tightly connected to the polymorphisms of Γ; in particular, if two finite constraint languages Γ_1, Γ_2 share exactly the same polymorphisms, then $\mathsf{CSP}(\Gamma_1)$ and $\mathsf{CSP}(\Gamma_2)$ are reducible to each other via many-one polynomial time reductions (and likewise for $\mathsf{QCSP}(\Gamma_1)$ and $\mathsf{QCSP}(\Gamma_2)$). When $\mu : D^k \to D$ is an operation, we use $\mathsf{Inv}(\mu)$ to denote the set of all relations having μ as polymorphism.

Definition 4. *Let $\mu : D^k \to D$ be an operation. The* $\mathsf{QCSP}(\mu)$ *problem is to decide, given as input a quantified formula ϕ with constraints having relations from* $\mathsf{Inv}(\mu)$*, whether or not ϕ is true.*

In other words, the $\mathsf{QCSP}(\mu)$ problem is that of deciding the truth of a quantified formula where all of the relations have μ as a polymorphism.

2.2 Collapsibility

In this paper, we will make use of machinery developed in [14] (on which the material in this subsection is based) to prove tractability results for the QCSP.

Collapsings of formulas. Define a quantified formula ϕ' to be a *j-collapsing* of a quantified formula ϕ if there exists a subset Y' of Y_ϕ such that $|Y'| = \min(j, |Y_\phi|)$ and ϕ' can be obtained from ϕ by first eliminating, from the quantifier prefix of ϕ, all variables in $Y_\phi \setminus Y'$ except for $\mathsf{first}_\phi(Y_\phi \setminus Y')$; and then replacing, in the constraint network of ϕ, all variables in $Y_\phi \setminus Y'$ with $\mathsf{first}_\phi(Y_\phi \setminus Y')$. For example, the quantified formula

$$\forall y_1 \exists x_1 \forall y_2 \forall y_3 \exists x_2 \{R_1(y_1, x_1), R_2(y_3, x_2), R_3(y_2, y_3, x_2)\}$$

has the following three 1-collapsings:

$$\forall y_1 \exists x_1 \forall y_2 \exists x_2 \{R_1(y_1, x_1), R_2(y_2, x_2), R_3(y_2, y_2, x_2)\}$$

$$\forall y_1 \exists x_1 \forall y_2 \exists x_2 \{R_1(y_1, x_1), R_2(y_1, x_2), R_3(y_2, y_1, x_2)\}$$
$$\forall y_1 \exists x_1 \forall y_3 \exists x_2 \{R_1(y_1, x_1), R_2(y_3, x_2), R_3(y_1, y_3, x_2)\}.$$

Note that if a quantified formula ϕ is true, then for any $j \geq 1$ and any j-collapsing ϕ' of ϕ, the quantified formula ϕ' is true.

Define a set of functions F of the form $\tau : Y \to D$ to be a *j-adversary set* for Y if there exists a subset $Y' \subseteq Y$ of size $\min(j, |Y|)$ such that F contains exactly those functions τ having the property that $\tau(y_1) = \tau(y_2)$ for all $y_1, y_2 \in Y \setminus Y'$. When ϕ is a quantified formula and F is a set of adversaries $\tau : Y_\phi \to D$, we say that the formula ϕ is F-winnable if there exists a strategy σ for ϕ such that for all $\tau \in F$, the assignment $\mathsf{outcome}(\sigma, \tau)$ satisfies the constraint network of ϕ. As an alternative to saying that the formula ϕ is F-winnable, we will say that F is winnable when ϕ is understood from the context.

It is straightforward to verify that the j-collapsing of a quantified formula corresponding to a subset Y' is true, if and only if the formula is F-winnable, where F is the j-adversary set corresponding to Y'. Hence, we have the following proposition.

Proposition 5. *Let ϕ be a quantified formula. The j-collapsings ϕ' of ϕ are all true if and only if for each j-adversary set F for Y_ϕ, the formula ϕ is F-winnable.*

Collapsibility of problems. We say that a problem of the form $\mathsf{QCSP}(\mu)$ is *j-collapsible* if for every quantified formula ϕ that is an instance of $\mathsf{QCSP}(\mu)$, the following property holds: if all j-collapsings of ϕ are true, then ϕ is true. We say that a problem $\mathsf{QCSP}(\mu)$ has *bounded collapsibility* if there exists a $j \geq 1$ such that $\mathsf{QCSP}(\mu)$ is j-collapsible. The following notion of composability is useful for identifying problems of the form $\mathsf{QCSP}(\mu)$ that have bounded collapsibility.

When $\mu : D^k \to D$ is a function, and $F, F_1, \ldots, F_k$ are sets of functions of the form $\tau : \{y_1, \ldots, y_n\} \to D$, we say that *$F$ is μ-composable in one step from $F_1, \ldots, F_k$* if there exist strategies $\pi^1 = \{\pi^1_i : D^i \to D\}_{i \in [n]}, \ldots, \pi^k = \{\pi^k_i : D^i \to D\}_{i \in [n]}$ such that for all $\tau \in F$, the following two properties hold: (1) for all $i \in [k]$, the mapping $\tau^i : \{y_1, \ldots, y_n\} \to D$ defined by $\tau^i(y_j) = \pi^i_j(\tau(y_1), \ldots, \tau(y_j))$ (for all $j \in [n]$) is contained in F_i, and (2) it holds that $\tau(y_j) = \mu(\tau^1(y_j), \ldots, \tau^k(y_j))$ (for all $j \in [n]$). The key feature of this definition is the following lemma.

Lemma 6. *Let ϕ be a quantified formula with $\{y_1, \ldots, y_n\}$ as its universally quantified variables and with relations invariant under μ, and suppose that F is μ-composable in one step from $F_1, \ldots, F_k$. If ϕ is F_i-winnable for all $i \in [k]$, then ϕ is F-winnable.*

The condition of bounded collapsibility can be combined with CSP tractability to infer QCSP tractability results. Recall that an operation $\mu : D^k \to D$ is *idempotent* when $\mu(d, \ldots, d) = d$ for all $d \in D$.

Theorem 7. *Suppose that μ is an idempotent operation such that $\mathsf{QCSP}(\mu)$ has bounded collapsibility and $\mathsf{CSP}(\mu)$ is decidable in polynomial time. Then, $\mathsf{QCSP}(\mu)$ is decidable in polynomial time.*

3 Classification of 2-semilattices

In this section, we present a complete complexity classification of 2-semilattice operations in quantified constraint satisfaction. We first formally state the classification theorem (Theorem 8) and discuss some of its implications; then, we prove the theorem in two parts.

3.1 Statement of classification theorem and implications

A 2-semilattice operation is a binary operation $\star : D^2 \to D$ such that for all $x, y \in D$ it holds that $x \star x = x$ (idempotence), $x \star y = y \star x$ (commutativity), and $(x \star x) \star y = x \star (x \star y)$ (restricted associativity). Every 2-semilattice operation $\star : D^2 \to D$ induces a directed graph $\mathcal{G}^\star = (D, E)$ with edge set $E = \{(a, b) \in D \times D : a \star b = b\}$. We use $\mathcal{C}^\star$ to denote the set of strongly connected components (or components, for short) of $\mathcal{G}^\star$, and let $\leq$ be the binary relation on $\mathcal{C}^\star$ where for $C_1, C_2 \in \mathcal{C}^\star$, it holds that $C_1 \leq C_2$ if and only if there exist vertices $v_1 \in C_1$, $v_2 \in C_2$ such that there is a path (in $\mathcal{G}^\star$) from v_1 to v_2. It is straightforward to verify that $\leq$ is a partial order. We say that $C \in \mathcal{C}^\star$ is a *minimal component* if it is minimal with respect to $\leq$, that is, for all $C' \in \mathcal{C}^\star$, $C' \leq C$ implies $C' = C$.

Our classification theorem demonstrates that 2-semilattice operations give rise to two modes of behavior in QCSP complexity, depending on the structure of the graph $\mathcal{G}^\star$.

Theorem 8. *Let $\star : D^2 \to D$ be a 2-semilattice operation. If there is a unique minimal component in $\mathcal{C}^\star$, then* $\mathsf{QCSP}(\star)$ *is decidable in polynomial time. Otherwise,* $\mathsf{QCSP}(\star)$ *is coNP-hard.*

One implication of this classification theorem is a complete classification of semilattice operations in QCSP complexity. Recall that a semilattice operation is a binary operation that is associative, commutative, and idempotent. We say that a semilattice operation $\star : D^2 \to D$ has a unit element if there exists an element $u \in D$ such that for all $d \in D$, $d \star u = u \star d = d$.

Corollary 9. *Let $\star : D^2 \to D$ be a semilattice operation. If $\star$ has a unit element, then* $\mathsf{QCSP}(\star)$ *is decidable in polynomial time. Otherwise,* $\mathsf{QCSP}(\star)$ *is coNP-hard.*

Proof. When $\star$ is a semilattice operation, it is straightforward to verify that each component in $\mathcal{C}^\star$ is of size one. Hence, $\mathcal{C}^\star$ has a unique minimal component if and only if $\star$ has a unit element. □

We note that the tractability of semilattice operations with unit has been previously derived [14].

Another implication of our classification theorem is the tractability of all *commutative conservative operations*. A commutative conservative operation is a binary operation $\star : D^2 \to D$ that is commutative and conservative. (We say that $\star$ is conservative if for all $x, y \in D$ it holds that $x \star y \in \{x, y\}$.) Such operations were studied in the context of the CSP by Bulatov and Jeavons [11].

Corollary 10. *Let $\star : D^2 \to D$ be a commutative conservative operation. The problem* $\mathsf{QCSP}(\star)$ *is decidable in polynomial time.*

Proof. It is straightforward to verify that when $\star$ is a commutative conservative operation, the relation $\leq$ on $\mathcal{C}^\star$ is a total ordering, and hence has a unique minimal component. $\square$

The proof of Theorem 8 can be generalized to a multi-sorted version of the QCSP. We refer the reader to [9] for the definition and a study of the multi-sorted CSP; the definition of the multi-sorted QCSP is analogous to this definition.

Theorem 11. *Let $\star$ be a multi-sorted binary operation over a finite collection of domains $\mathcal{D}$, such that the interpretation $\star^D$ of $\star$ on any domain $D \in \mathcal{D}$ is a 2-semilattice operation. If for every domain $D \in \mathcal{D}$ there is a unique minimal component in $\mathcal{C}^{\star^D}$, then* $\mathsf{QCSP}(\star)$ *is decidable in polynomial time. Otherwise,* $\mathsf{QCSP}(\star)$ *is coNP-hard.*

3.2 Proof

We prove Theorem 8 in two parts: the tractable cases are established in Theorem 12, and the intractable cases are established in Theorem 13.

Theorem 12. *Let $\star : D^2 \to D$ be a 2-semilattice operation. If there is a unique minimal component in $\mathcal{C}^\star$, then* $\mathsf{QCSP}(\star)$ *is decidable in polynomial time.*

Proof. First, fix an element b in the minimal component of $\mathcal{C}^\star$. For every element $d \in D$, there exists a path from b to d in $\mathcal{G}^\star$. (This is because for every component C of $\mathcal{C}^\star$ it holds that $B \leq C$, where B denotes the minimal component of $\mathcal{C}^\star$.) Let k be a sufficiently large integer so that for every $d \in D$, there is a path from b to d of length less than or equal to k.

We show that $\mathsf{QCSP}(\star)$ is $(2^k - 1)$-collapsible; the result then follows from Theorem 7 along with the tractability of $\mathsf{CSP}(\star)$ (see [3]). Let ϕ be an instance of $\mathsf{QCSP}(\star)$ where all $(2^k - 1)$-collapsings of ϕ are true. Then, we have that for each $(2^k - 1)$-adversary set F for Y, the set F is winnable. For $W \subseteq Y$ and $d \in D$, define the set $F_{(W,d)}$ to be the set of adversaries $\tau : Y \to D$ such that $\tau(y) = d$ for all $y \in Y \setminus W$. We prove that for every universal variable subset $W \subseteq Y$ with $|W| \geq 2^k - 1$, the set $F_{(W,b)}$ is winnable. This suffices, as it implies that the set $F_{(Y,b)}$, which is the set of *all* adversaries $\tau : Y \to D$, is winnable.

The proof is by induction on $|W|$. When $|W| = 2^k - 1$, the claim is immediate from the assumption that that each $(2^k - 1)$-adversary set (for Y) is winnable, along with the fact that any subset of a winnable adversary set is winnable. For the induction, assume that $|W| > 2^k - 1$ and let W_0 be a subset of W having size 2^k. Let $\{w_1, \ldots, w_{2^k}\}$ denote the elements of W_0. Let F_i denote $F_{(W \setminus \{w_i\}, b)}$ (for all $i \in [2^k]$). We may assume by induction that F_i is winnable (for all $i \in [2^k]$).

We claim that there are strategies $\pi^1 = \{\pi_i^1 : D^i \to D\}_{i \in [n]}, \ldots, \pi^{2^k} = \{\pi_i^{2^k} : D^i \to D\}_{i \in [n]}$ such that for all $\tau \in F$, the following two properties hold:

(1) for all $i \in [2^k]$, the mapping $\tau^i : \{y_1, \ldots, y_n\} \to D$ defined by $\tau^i(y_j) = \pi^i_j(\tau(y_1), \ldots, \tau(y_j))$ (for all $j \in [n]$) is contained in F_i, and
(2) for all $j \in [n]$, it holds that

$$\tau(y_j) = ((\tau^1(y_j) \star \tau^2(y_j)) \star (\tau^3(y_j) \star \tau^4(y_j))) \star \cdots$$

where the right hand side of the above expression is a balanced binary tree of depth k with leaves $\tau^1(y_j), \ldots, \tau^{2^k}(y_j)$.

By appeal to Lemma 6, establishing this claim entails that $F_{(W,b)}$ is winnable. We consider two cases:

- For universal variables $y_j \in Y \setminus W_0$, we let π^i_j be the projection onto the last coordinate (for all i), so that $\tau^i(y_j) = \tau(y_j)$ and property (2) holds by the idempotence of $\star$. Note that since $\tau(y_j) = b$ when $y_j \in Y \setminus W$, we have that $\tau^i(y_j) = b$ for such y_j (for all i).
- For universal variables $y_j \in W_0$, it suffices by symmetry to show that the polynomial
$$(((b \star x_2) \star (x_3 \star x_4)) \star \cdots)$$
that is a balanced binary tree of depth k with leaves $b, x_2, \ldots, x_{2^k}$, is surjective. Identifying the variables $x_{2^i+1}, \ldots, x_{2^{i+1}}$ to be equivalent (for i ranging from 0 to $k-1$), it suffices to show that the polynomial
$$(\cdots((b \star x'_0) \star x'_1) \star \cdots \star x'_{k-1})$$
is surjective. This follows immediately from our choice of k: for any element $d \in D$, there is a path from b to d in $\mathcal{G}^\star$ of length k. (Note that every vertex in $\mathcal{G}^\star$ has a self-loop, so as long as there is a path from b to d with length less than or equal to k, there is a path from b to d with length equal to k.) □

We complete the classification of 2-semilattices by proving a complement to Theorem 12, namely, that the remaining 2-semilattices give rise to QCSPs that are intractable.

Theorem 13. *Let $\star : D^2 \to D$ be a 2-semilattice operation. If there are two or more minimal components in $\mathcal{C}^\star$, then* $\mathsf{QCSP}(\star)$ *is coNP-hard (even when restricted to $\forall\exists$-formulas).*

Proof. We show coNP-hardness by reducing from the propositional tautology problem. Let $C(y_1, \ldots, y_n)$ be an instance of this problem, where C is a circuit with input gates having labels $y_1, \ldots, y_n$. We assume without loss of generality that all non-input gates of C are either AND or NOT gates, and assign all non-input gates labels $x_1, \ldots, x_m$. The quantifier prefix of the resulting quantified formula is $\forall y_1 \ldots \forall y_n \exists x_1 \ldots \exists x_m$. Let B_0 and B_1 be distinct minimal components in $\mathcal{C}^\star$, and let b_0 and b_1 be elements of B_0 and B_1, respectively. The constraint network of the resulting quantified formula is constructed as follows.

For each AND gate x_i with inputs $v, v' \in \{y_1, \ldots, y_n\} \cup \{x_1, \ldots, x_m\}$, include the four constraints:

- $(v \in B_1) \wedge (v' \in B_1) \Rightarrow (x_i = b_1)$
- $(v \in B_0) \wedge (v' \in B_1) \Rightarrow (x_i = b_0)$
- $(v \in B_1) \wedge (v' \in B_0) \Rightarrow (x_i = b_0)$
- $(v \in B_0) \wedge (v' \in B_0) \Rightarrow (x_i = b_0)$

For each NOT gate x_i with input $v \in \{y_1, \ldots, y_n\} \cup \{x_1, \ldots, x_m\}$, include the two constraints:

- $(v \in B_0) \Rightarrow (x_i = b_1)$
- $(v \in B_1) \Rightarrow (x_i = b_0)$

For the output gate x_o, include the constraint:

- $(x_o \in B_0) \Rightarrow$ FALSE

It is fairly straightforward to verify that each of the given constraints has the $\star$ operation as polymorphism; the key fact is that (for $i \in \{0, 1\}$) multiplying any element of D by an element c outside of B_i yields an element c' outside of B_i. (If not, there is an edge from c to c' by restricted associativity of $\star$; this edge gives a contradiction to the minimality of B_i).

We verify the reduction to be correct as follows.

Suppose that the original circuit was a tautology. Let $f : \{y_1, \ldots, y_n\} \rightarrow D$ be any assignment to the $\forall$-variables of the quantified formula. Define $f' : \{y_1, \ldots, y_n\} \rightarrow (B_0 \cup B_1)$ by $f'(y_i) = f(y_i)$ if $f(y_i) \in B_0 \cup B_1$, and as an arbitrary element of $B_0 \cup B_1$ otherwise. The AND and NOT gate constraints force each x_i to have either the value b_0 or b_1 under f'; it can be verified that the assignment taking x_i to its forced value and y_i to $f(y_i)$ satisfies all of the constraints.

Suppose that the original circuit was not a tautology. Let $g : \{y_1, \ldots, y_n\} \rightarrow \{0, 1\}$ be an assignment making the circuit C false. Let $g' : \{y_1, \ldots, y_n\} \rightarrow D$ be an assignment to the $\forall$-variables of the quantified formula such that $g'(y_i) \in B_{g(y_i)}$ (for all $i \in [n]$). Under the assignment g', the only assignment to the $\exists$-variables x_i satisfying the AND and NOT gate constraints is the mapping taking x_i to b_0 if the gate with label x_i has value 0 under g, and b_1 if the gate with label x_i has value 1 under g. Hence, if all of these constraints are satisfied, then the output gate constraint must be falsified. We conclude that no assignment to the $\exists$-variables x_i satisfies all of the constraints under the assignment g', and so the quantified formula is false. $\square$

4 Tractable 2-semilattices are 1-collapsible

In the previous section, it was shown that for certain 2-semilattice operations $\star$, the problem $\mathsf{QCSP}(\star)$ is polynomial-time tractable. For these operations, it was proved that $\mathsf{QCSP}(\star)$ is j-collapsible for *some* constant j. However, the given proof demonstrated j-collapsibility for constants j that could be arbitrarily large, depending on the operation $\star$. In this section, we refine this result by proving the strongest possible statement concerning the collapsibility of $\mathsf{QCSP}(\star)$: the problem $\mathsf{QCSP}(\star)$ is 1-collapsible whenever $\mathsf{QCSP}(\star)$ is tractable.

Theorem 14. *Let $\star : D^2 \to D$ be a 2-semilattice operation. If there is a unique minimal component in $\mathcal{C}^\star$, then* $\mathsf{QCSP}(\star)$ *is 1-collapsible.*

In the proof of this theorem, we write adversaries (and sets of adversaries) using tuple notation. We also use the notation $F \lhd F_1 \star F_2$ to denote that F is $\star$-composable in one step from F_1 and F_2.

Proof. Fix d_0 to be any element of the unique minimal component of $\mathcal{C}^\star$. For every element $d \in D$, there exists a path from d_0 to d in $\mathcal{G}^\star$. Hence, it is possible to select sufficiently large integers K and L and elements $\{d_i^j\}_{i\in[L],j\in[K]}$ so that:

– each of the sets

$$P^1 = \{d_0^1, d_1^1, \ldots, d_L^1\}$$

$$\vdots$$

$$P^K = \{d_0^K, d_1^K, \ldots, d_L^K\}$$

is a path in the sense that, for all $j \in [K]$, all of the pairs

$$(d_0^j, d_1^j), (d_1^j, d_2^j), \ldots, (d_{L-1}^j, d_L^j)$$

are edges in $\mathcal{G}^\star$; and,

– all elements of D lie on one of these paths, that is, $D = \cup_{j=1}^K P^j$.

Here, we use $d_0^1, \ldots, d_0^K$ as alternative notation for d_0, that is,

$$d_0 = d_0^1 = \cdots = d_0^K.$$

For $i = 0, \ldots, L$ we define

$$D_i \stackrel{\text{def}}{=} \{d_i^1, \ldots, d_i^k\},$$

and we define

$$E_i \stackrel{\text{def}}{=} D_0 \cup \cdots \cup D_i.$$

Notice that $E_L = D$.

To prove 1-collapsibility, we fix a problem instance, and assume that each 1-adversary set is winnable. In particular, we assume that each of the sets

$$D \times \{d_0\} \times \cdots \times \{d_0\}$$

$$\vdots$$

$$\{d_0\} \times \cdots \times \{d_0\} \times D$$

is winnable. Our goal is to prove that $D \times \cdots \times D$ is winnable.

We prove by induction that for all $i = 0, \ldots, L$ the set $E_i \times \cdots \times E_i$ is winnable. The base case $i = 0$ holds by assumption. For the induction, assume that $E_i \times \cdots \times E_i$ is winnable (where $i \geq 0$); we show that $E_{i+1} \times \cdots \times E_{i+1}$ is winnable.

We first show that $E_{i+1} \times E_i \times \cdots \times E_i$ is winnable. We have

$$(E_{i+1} \times E_0 \times \cdots \times E_0) \lhd (E_i \times E_i \times \cdots \times E_i) \star (D \times \{d_0\} \times \cdots \times \{d_0\}).$$

Then, we have

$$(E_{i+1} \times E_1 \times \cdots \times E_1) \lhd (E_i \times E_i \times \cdots \times E_i) \star (E_{i+1} \times E_0 \times \cdots \times E_0)$$

and

$$(E_{i+1} \times E_2 \times \cdots \times E_2) \lhd (E_i \times E_i \times \cdots \times E_i) \star (E_{i+1} \times E_1 \times \cdots \times E_1).$$

Continuing in this manner, we can show that $E_{i+1} \times E_i \times \cdots \times E_i$ is winnable. By symmetric arguments, we can show the winnability of the sets

$$(E_i \times E_{i+1} \times E_i \times \cdots \times E_i)$$

$$\vdots$$

$$(E_i \times E_i \times \cdots \times E_i \times E_{i+1}).$$

We have the winnability of $E_{i+1} \times E_{i+1} \times E_i \times \cdots E_i$ by

$$(E_{i+1} \times E_{i+1} \times E_i \times \cdots E_i) \lhd (E_{i+1} \times E_i \times E_i \times \cdots \times E_i) \star (E_i \times E_{i+1} \times E_i \times \cdots \times E_i)$$

and we have the winnability of $E_{i+1}^3 \times E_i \times \cdots \times E_i$ by

$$(E_{i+1}^3 \times E_i \times \cdots \times E_i) \lhd (E_{i+1}^2 \times E_i \times E_i \times \cdots \times E_i) \star (E_i^2 \times E_{i+1} \times E_i \times \cdots \times E_i).$$

Proceeding in this fashion, we have the winnability of $E_{i+1} \times \cdots \times E_{i+1}$, as desired. $\square$

References

1. Bengt Aspvall, Michael F. Plass, and Robert Endre Tarjan. A Linear-Time Algorithm for Testing the Truth of Certain Quantified Boolean Formulas. Inf. Process. Lett. 8(3): 121-123 (1979).
2. F. Börner, A. Bulatov, A. Krokhin, and P. Jeavons. Quantified Constraints: Algorithms and Complexity. Computer Science Logic 2003.
3. A. Bulatov. Combinatorial problems raised from 2-semilattices. Manuscript.
4. Andrei A. Bulatov. A Dichotomy Theorem for Constraints on a Three-Element Set. FOCS 2002.
5. A. Bulatov. Malt'sev constraints are tractable. Technical report PRG-RR-02-05, Oxford University, 2002.
6. Andrei A. Bulatov. Tractable conservative Constraint Satisfaction Problems. LICS 2003.
7. Andrei A. Bulatov and Andrei A. Krokhin and Peter Jeavons. The complexity of maximal constraint languages. STOC 2001.
8. Andrei A. Bulatov, Andrei A. Krokhin, and Peter Jeavons. Constraint Satisfaction Problems and Finite Algebras. ICALP 2000.

9. A. Bulatov and P. Jeavons. An Algebraic Approach to Multi-sorted Constraints Proceedings of 9th International Conference on Principles and Practice of Constraint Programming, 2003.
10. A. Bulatov, and P. Jeavons. Algebraic structures in combinatorial problems. Technical report MATH-AL-4-2001, Technische Universitat Dresden, 2001.
11. A. Bulatov, and P. Jeavons. Tractable constraints closed under a binary operation. Technical report PRG-TR-12-00, Oxford University, 2000.
12. Hans Kleine Büning, Marek Karpinski, and Andreas Flögel. Resolution for Quantified Boolean Formulas. Information and Computation 117(1): 12-18 (1995).
13. Hubie Chen. Quantified Constraint Satisfaction Problems: Closure Properties, Complexity, and Algorithms. Cornell technical report, 2003.
14. Hubie Chen. Collapsibility and Consistency in Quantified Constraint Satisfaction. AAAI 2004.
15. Nadia Creignou, Sanjeev Khanna, and Madhu Sudan. Complexity Classifications of Boolean Constraint Satisfaction Problems. SIAM Monographs on Discrete Mathematics and Applications 7, 2001.
16. Victor Dalmau. Some Dichotomy Theorems on Constant-free Quantified Boolean Formulas. Technical report LSI-97-43-R, Llenguatges i Sistemes Informàtics - Universitat Politècnica de Catalunya, 1997.
17. Victor Dalmau and Justin Pearson. Set Functions and Width 1. Constraint Programming '99.
18. Tomás Feder and Moshe Y. Vardi. The Computational Structure of Monotone Monadic SNP and Constraint Satisfaction: A Study through Datalog and Group Theory. SIAM J. Comput. 28(1): 57-104, 1998.
19. Peter Jeavons. On the Algebraic Structure of Combinatorial Problems. Theor. Comput. Sci. 200(1-2): 185-204, 1998.
20. P.G.Jeavons, D.A.Cohen and M.Cooper. Constraints, Consistency and Closure. Artificial Intelligence, 1998, 101(1-2), pages 251-265.
21. Peter Jeavons, David A. Cohen, and Marc Gyssens. Closure properties of constraints. J. ACM 44(4): 527-548 (1997).
22. Phokion G. Kolaitis, Moshe Y. Vardi. Conjunctive-Query Containment and Constraint Satisfaction. J. Comput. Syst. Sci. 61(2): 302-332 (2000)
23. Phokion G. Kolaitis and Moshe Y. Vardi. A Game-Theoretic Approach to Constraint Satisfaction. AAAI 2000.
24. I.G. Rosenberg. Minimal Clones I: the five types. Lectures in Universal Algebra (Proc. Conf. Szeged 1983), Colloq. Math. Soc. Janos Bolyai. 1986.
25. T. Schaefer. The complexity of satisfiability problems. Proceedings of the 10th Annual Symposium on Theory of Computing, ACM, 1978.