QCSP Monsters and the Demise of the Chen Conjecture

Dmitriy Zhuk
zhuk@intsys.msu.ru
Department of Mechanics and Mathematics, Lomonosov Moscow State University
Russia

Barnaby Martin
barnaby.d.martin@durham.ac.uk
Department of Computer Science, Durham University
United Kingdom

ABSTRACT

We give a surprising classification for the computational complexity of the Quantified Constraint Satisfaction Problem over a constraint language Γ, QCSP(Γ), where Γ is a finite language over 3 elements which contains all constants. In particular, such problems are either in P, NP-complete, co-NP-complete or PSpace-complete. Our classification refutes the hitherto widely-believed Chen Conjecture.

Additionally, we show that already on a 4-element domain there exists a constraint language Γ such that QCSP(Γ) is DP-complete (from Boolean Hierarchy), and on a 10-element domain there exists a constraint language giving the complexity class Θ_2^P.

Meanwhile, we prove the Chen Conjecture for finite conservative languages Γ. If the polymorphism clone of such Γ has the polynomially generated powers (PGP) property then QCSP(Γ) is in NP. Otherwise, the polymorphism clone of Γ has the exponentially generated powers (EGP) property and QCSP(Γ) is PSpace-complete.

CCS CONCEPTS

• **Theory of computation → Problems, reductions and completeness**; **Complexity theory and logic**; *Logic and verification.*

KEYWORDS

quantified constraints, constraint satisfaction, universal algebra, computational complexity

ACM Reference Format:
Dmitriy Zhuk and Barnaby Martin. 2020. QCSP Monsters and the Demise of the Chen Conjecture. In *Proceedings of the 52nd Annual ACM SIGACT Symposium on Theory of Computing (STOC '20), June 22–26, 2020, Chicago, IL, USA.* ACM, New York, NY, USA, 14 pages. https://doi.org/10.1145/3357713.3384232

1 INTRODUCTION

The *Quantified Constraint Satisfaction Problem* QCSP(Γ) is the generalization of the *Constraint Satisfaction Problem* CSP(Γ) which, given the latter in its logical form, augments its native existential quantification with universal quantification. That is, QCSP(Γ) is the problem to evaluate a sentence of the form $\forall x_1 \exists y_1 \ldots \forall x_n \exists y_n\ \Phi$, where Φ is a conjunction of relations from the *constraint language* Γ, all over the same finite domain D. Since the resolution of the Feder-Vardi "Dichotomy" Conjecture, classifying the complexity of CSP(Γ), for all finite Γ, between P and NP-complete [6, 24], a desire has been building for a classification for QCSP(Γ). Indeed, since the classification of the *Valued CSPs* was reduced to that for CSPs [18], the QCSP remains the last of the older variants of the CSP to have been systematically studied but not classified. More recently, other interesting open classification questions have appeared such as that for *Promise CSPs* [5] and finitely-bounded, homogeneous infinite-domain CSPs [1].

While CSP(Γ) remains in NP for any finite Γ, QCSP(Γ) can be PSpace-complete, as witnessed by *Quantified 3-Satisfiability* or *Quantified Graph 3-Colouring* (see [4]). It is well-known that the complexity classification for QCSPs embeds the classification for CSPs: if $\Gamma + 1$ is Γ with the addition of a new isolated element not appearing in any relations, then CSP(Γ) and QCSP($\Gamma + 1$) are polynomially equivalent. Thus, and similarly to the Valued CSPs, the CSP classification will play a part in the QCSP classification. It is now clear that QCSP(Γ) can achieve each of the complexities P, NP-complete and PSpace-complete. It has thus far been believed these were the only possibilities (see [4, 11, 12, 14, 22] and indeed all previous papers on the topic).

A key role in classifying many CSP variants has been played by Universal Algebra. We say that a k-ary operation f *preserves* an m-ary relation R, whenever $(x_1^1, \ldots, x_1^m), \ldots, (x_k^1, \ldots, x_k^m)$ in R, then also $(f(x_1^1, \ldots, x_k^1), \ldots, f(x_1^m, \ldots, x_k^m))$ in R. The relation R is called *an invariant* of f, and the operation f is called *a polymorphism* of R. An operation f is *a polymorphism* of Γ if it preserves every relation from Γ. The *polymorphism clone* Pol(Γ) is the set of all polymorphisms of Γ. Similarly, a relation R is *an invariant* of a set of functions F if it is preserved by every operation from F. By Inv(F) we denote the set of all invariants of F. We call an operation f *idempotent* if $f(x, \ldots, x) = x$, for all x. An idempotent operation f is a *weak near-unanimity* (WNU) operation if $f(y, x, x, \ldots, x) = f(x, y, x, \ldots, x) = \cdots = f(x, x, \ldots, x, y)$. We recall the following form of the Feder-Vardi Conjecture.

Theorem 1 (CSP Dichotomy [6, 24]). *Let Γ be a finite constraint language with all constants. If Γ admits some WNU polymorphism, then* CSP(Γ) *is in P. Otherwise,* CSP(Γ) *is NP-complete.*

For the CSP one may assume, without loss of generality, that Γ contains all constants (one can imagine these appearing in various forms, one possibility being all unary relations $x = c$, for $c \in D$). This is equivalent to the assumption that all operations f of Pol(Γ) are idempotent. We can achieve this by moving to an equivalent constraint language known as the *core.* The situation is more complicated for the QCSP and it is not known that a similar trick may be accomplished (see [15]). However, all prior conjectures for the QCSP have been made in this safer environment where we may

STOC '20, June 22–26, 2020, Chicago, IL, USA

ACM ISBN 978-1-4503-6979-4/20/06…$15.00
https://doi.org/10.1145/3357713.3384232

assume idempotency and almost all classifications apply only to this situation. A rare exception to this is the paper [16] where the non-idempotent case is described as the *terra incognita*. We will henceforth assume Γ contains all constants.

For the purpose of pedagogy it is useful to look at the Π_2 restriction of $\mathrm{QCSP}(\Gamma)$, denoted $\mathrm{QCSP}^2(\Gamma)$, in which the input is of the form $\forall x_1 \ldots \forall x_n \exists y_1 \ldots \exists y_m\ \Phi$. In order to solve this restriction of the problem it suffices to look at (the conjunction of) $|D|^n$ instances of $\mathrm{CSP}(\Gamma)$. It is not hard to show (see [13]) that, if D^n can be generated under $\mathrm{Pol}(\Gamma)$ from some subset $\Sigma \subseteq D^n$, then one need only consult (the conjunction over) of $|\Sigma|$ instances of $\mathrm{CSP}(\Gamma)$. Suppose there is a polynomial p such that for each n there is a subset $\Sigma \subseteq D^n$ of size at most $p(n)$ so that D^n can be generated under $\mathrm{Pol}(\Gamma)$ from Σ, then we say $\mathrm{Pol}(\Gamma)$ has the *polynomially generated powers* (PGP) property. Under the additional assumption that there is a polynomial algorithm that computes these Σ, we would have a reduction to $\mathrm{CSP}(\Gamma)$. It turns out that if the nature of the PGP property is sufficiently benign a similar reduction can be made for the full $\mathrm{QCSP}(\Gamma)$ to the CSP with constants [8, 13]. Another behaviour that might arise with $\mathrm{Pol}(\Gamma)$ is that there is an exponential function f so that the smallest generating sets under $\mathrm{Pol}(\Gamma)$ for $\Sigma \subseteq D^n$ require size at least $f(n)$. We describe this as the the *exponentially generated powers* (EGP) property. The outstanding conjecture in the area of QCSPs is the merger of Conjectures 6 and 7 in [14] which we have dubbed in [9] the *Chen Conjecture*.

Conjecture 1 (Chen Conjecture)**.** *Let* Γ *be a finite constraint language with all constants. If* $\mathrm{Pol}(\Gamma)$ *has PGP, then* $\mathrm{QCSP}(\Gamma)$ *is in NP; otherwise* $\mathrm{QCSP}(\Gamma)$ *is PSpace-complete.*

In [14], Conjecture 6 gives the NP membership and Conjecture 7 the PSpace-completeness. In light of the proofs of the Feder-Vardi Conjecture, the Chen Conjecture implies the trichotomy of idempotent QCSP among P, NP-complete and PSpace-complete. Chen does not state that the PSpace-complete cases arise only from EGP, but this would surely have been on his mind (and he knew there was a dichotomy between PGP and EGP already for 3-element idempotent algebras [13]). Since [25], it has been known for any finite domain that only the cases PGP and EGP arise (even in the non-idempotent case), and that PGP is always witnessed in the form of *switchability*. It follows that we know that the PGP cases are in NP [8, 13].

Theorem 2 ([9])**.** *Let* Γ *be a finite constraint language with all constants such that* $\mathrm{Pol}(\Gamma)$ *has PGP. Then* $\mathrm{QCSP}(\Gamma)$ *reduces to a polynomial number of instances of* $\mathrm{CSP}(\Gamma)$ *and is in NP.*

Using the CSP classification we can then separate the PGP cases into those that are in P and those that are NP-complete.

A tantalizing characterization of idempotent $\mathrm{Pol}(\Gamma)$ that are EGP is given in [25], where it is shown that Γ must allow the *primitive positive* (pp) definition (of the form $\exists x_1 \ldots \exists x_n\ \Phi$) of relations τ_n with the following special form.

Definition 1. *Let the domain* D *be so that* $\alpha \cup \beta = D$ *yet neither of* α *nor* β *equals* D*. Let* $S = \alpha^3 \cup \beta^3$ *and* τ_n *be the* $3n$*-ary relation given by* $\bigvee_{i \in [n]} S(x_i, y_i, z_i)$*.*

The complement to S represents the Not-All-Equal relation and the relations τ_n allow for the encoding of the complement of *Not-All-Equal 3-Satisfiability* (where $\alpha \setminus \beta$ is 0 and $\beta \setminus \alpha$ is 1). Thus, if one has polynomially computable (in n) pp-definitions of τ_n, then it is clear that $\mathrm{QCSP}(\Gamma)$ is co-NP-hard [9]. In light of this observation, it seemed that only a small step remained to proving the actual Chen Conjecture, at least with co-NP-hard in place of PSpace-complete.

In this paper we refute the Chen Conjecture in a strong way while giving a long-desired classification for $\mathrm{QCSP}(\Gamma)$ where Γ is a finite 3-element constraint language with constants. Not only do we find Γ so that $\mathrm{QCSP}(\Gamma)$ is co-NP-complete, but also we find Γ so that $\mathrm{Pol}(\Gamma)$ has EGP yet $\mathrm{QCSP}(\Gamma)$ is in P. In these latter cases we can further prove that all pp-definitions of τ_n in Γ are of size exponential in n. Additionally, we show that on a 4-element domain there exists a constraint language Γ such that $\mathrm{QCSP}(\Gamma)$ is DP-complete (from the Boolean Hierarchy), and on a 10-element domain there exists a constraint language giving the complexity class Θ_2^P. Our main result for QCSP can be given as follows.

Theorem 3. *Let* Γ *be a finite constraint language on* 3 *elements which includes all constants. Then* $\mathrm{QCSP}(\Gamma)$ *is either in P, NP-complete, co-NP-complete or PSpace-complete.*

Meanwhile, we prove the Chen Conjecture is true for the class of finite conservative languages (these are those that have available all unary relations). One might see this as a maximal natural class on which the Chen Conjecture holds. Another form of "conservative QCSP", in which relativization of the universal quantifier is permitted, has been given by Bodirsky and Chen [2]. They uncovered a dichotomy between P and PSpace-complete, whereas the QCSP for finite conservative languages bequeaths the following trichotomy.

Theorem 4 (Conservative QCSP)**.** *Let* Γ *be a finite constraint language with all unary relations. If* $\mathrm{Pol}(\Gamma)$ *has PGP, then* $\mathrm{QCSP}(\Gamma)$ *is in NP. If* Γ *further admits a WNU polymorphism, then* $\mathrm{QCSP}(\Gamma)$ *is in P, else it is NP-complete. Otherwise,* $\mathrm{Pol}(\Gamma)$ *has EGP and* $\mathrm{QCSP}(\Gamma)$ *is PSpace-complete.*

It is hard to exaggerate how surprising our discovery of multitudinous complexities above P for the QCSP is. In Table 1 from [21], all syntactic fragments of first-order logic built from subsets of $\{\forall, \exists, \wedge, \vee, \neg, =\}$ are considered. It is now known that they all give model-checking problems with simple, structured complexity-theoretical classifications (the classifications are simple but not necessarily the proofs), except the QCSP ($\{\forall, \exists, \wedge\}$, with or without $=$), and its dual ($\{\forall, \exists, \vee\}$, with or without $\neq$), whose complexity classification is in any case a mirror of that for the QCSP. This holds for complexity classes of P and above (the classification of CSP complexities within P is quite rich).

1.1 Related Work

In [9], we have proved a variant of the Chen Conjecture using infinite relational languages encoded in quantifier-free logic with constants and equality. An algebra consists of a finite domain and a set of operations on that domain. A polymorphism clone is an excellent example of an algebra which additionally satisfies certain properties of closure.

Theorem 5 (Revised Chen Conjecture [9])**.** *Let* $\mathbb{A}$ *be an idempotent algebra on a finite domain* A *where we encode relations in* $\mathrm{Inv}(\mathbb{A})$ *in quantifier-free logic with constants and equality. If* $\mathbb{A}$ *satisfies PGP, then* $\mathrm{QCSP}(\mathrm{Inv}(\mathbb{A}))$ *is in NP. Otherwise,* $\mathrm{QCSP}(\mathrm{Inv}(\mathbb{A}))$ *is co-NP-hard.*

In this theorem it was known that co-NP-hardness could not be improved to PSpace-completeness, because QCSP(Inv($\mathbb{A}$)) is co-NP-complete when, e.g., $\mathbb{A} = \mathrm{Pol}(\{0,1,2\}; 0,1,2,\tau_1,\tau_2,\ldots\})$ where $\alpha = \{0,2\}$ and $\beta = \{1,2\}$. However, Inv($\mathbb{A}$) is not finitely related. It was not thought possible that there could be finite Γ such that QCSP(Γ) is co-NP-complete. If we take the tuple-listing encoding of relations instead of quantifier-free logic with constants and equality, Theorem 5 is known to fail [9].

The systematic complexity-theoretic study of QCSPs dates to the early versions of [4] (the earliest is a technical report from 2002). By the time of the journal version [4], the significance of the semilattice-without-unit $s = s_c$ (definition at opening of Section 2.1) had become apparent in a series of papers of Chen [10, 12, 13]. Although CSP(Inv($\{s\}$)) is in P it is proved in [4] that QCSP(Inv($\{s\}$)) is PSpace-complete (even for some finite sublanguage of Inv($\{s\}$)). We were unable to use the proof from that paper to expand the PSpace-complete classification in the 3-element case, but we have expanded it nonetheless.

Finally, the study of which sequences of relations R_i, of arity i, have polynomial-sized (in i) pp-definitions in a finite constraint language Γ, has been addressed in [19]. Of course, this question for our relations τ_i plays a central role in this paper.

1.2 Structure of the paper

The paper is organized as follows. In Section 2 we formulate the main results of the paper. We start with the classification of the complexity of QCSP(Γ) for constraint languages Γ on a 3-element domain containing all constants. Then we show how we can combine two constraint languages in one constraint language and explain how this idea gives exotic complexity classes such as DP = NP $\wedge$ co-NP.

In Section 3 we give necessary further definitions, then in Section 4 prove Chen's Conjecture for the conservative case. In Section 5 we prove that the combination of two constraint languages can actually give new complexity classes.

In Sections 6 to 9, we give examples of our new complexity results on a 3-element domain. In Section 6 we give a Γ so that QCSP(Γ) is co-NP-complete. In Section 7 we give a new Γ so that QCSP(Γ) is PSpace-complete. Finally, in Sections 8 and 9, we give two examples of new Γ so that QCSP(Γ) is in P yet Pol(Γ) has EGP.

Owing to space restrictions, the proof of our main result (Theorem 6) is omitted. It can be found in the full version of this paper [26].

2 MAIN RESULTS

In this section we formulate two main results of the paper: classification of the complexity of QCSP(Γ) for all constraint languages Γ on a 3-element domain containing all constants, and a theorem showing how we can combine constraint languages to obtain exotic complexity classes.

2.1 QCSP on a 3-element domain

Let a and c be constants of our domain $\{0,1,2\}$.

$$f_{a,c}(x,y,z) = \begin{cases} x, & \text{if } x = y \text{ or } y = z = a \\ c, & \text{otherwise.} \end{cases}$$

$$s_{a,c}(x,y) = \begin{cases} x, & \text{if } x = y \text{ or } y = a \\ c, & \text{otherwise.} \end{cases}$$

$$g_{a,c}(x,y) = \begin{cases} x, & \text{if } x = a \text{ or } y \neq c \\ c, & \text{otherwise.} \end{cases}$$

$$s_c(x,y) = \begin{cases} x, & \text{if } x = y \\ c, & \text{otherwise.} \end{cases}$$

We get the following characterization of the complexity of QCSP(Γ) on a 3-element domain.

Theorem 6. *Suppose Γ is a finite constraint language on $\{0,1,2\}$ with constants. Then* QCSP(Γ) *is*

(1) *in P, if* Pol(Γ) *has the PGP property and has a WNU operation.*
(2) *NP-complete, if* Pol(Γ) *has the PGP property and has no a WNU operation.*
(3) *PSpace-complete, if* Pol(Γ) *has the EGP property and has no a WNU operation.*
(4) *PSpace-complete, if* Pol(Γ) *has the EGP property and* Pol(Γ) *does not contain f such that $f(x,a) = x$ and $f(x,c) = c$, where $a, c \in \{0,1,2\}$.*
(5) *in P, if* Pol(Γ) *contains $s_{a,c}$ and $g_{a,c}$ for some $a, c \in \{0,1,2\}$, $a \neq c$.*
(6) *in P, if* Pol(Γ) *contains $f_{a,c}$ for some $a, c \in \{0,1,2\}$, $a \neq c$.*
(7) *co-NP-complete otherwise.*

Note that the semilattice s_c can be derived from each of the operations $f_{a,c}$, $s_{a,c}$. As we know from [4], the problem QCSP(Inv(s_2)) is PSpace-complete. Figure 1 demonstrates how adding new operations makes the constraint language weaker and the corresponding QCSP easier. Note that all the constraint languages on Figure 1 have the EGP property.

Let us give examples in each of the classes above. For (1) we can build a constraint language Γ with a single ternary relation $x - y + z = 1$. For (2) we can take a single ternary relation

$$\{(1,0,0),(0,1,0),(0,0,1)\}$$

that doesn't involve 2. For (3) we can take the closely related single ternary relation

$$\{(x,0,0),(0,x,0),(0,0,x) : x \in \{1,2\}\}.$$

For (4) see Section 7. For (5) see Section 8. For (6) see Section 9. Finally, for (7) see Section 6.

2.2 QCSP Monsters

The following theorem shows how we can combine constraint languages to obtain QCSPs with different complexities.

Theorem 7. *Suppose Γ_1, Γ_2, and Γ_3 are finite constraint languages on sets A_1, A_2, and A_3 respectively, Γ_1 contains a constant relation $(x = a)$. Then there exist constraint languages $\Delta_1, \Delta_2, \Delta_3, \Delta_4$, on the domains of size $|A_1| + 1$, $|A_2| \cdot |A_3| + |A_2| + |A_3|$, $2 \cdot |A_2| + |A_3| + 2$, and $|A_2| \cdot |A_3| + |A_2| + |A_3| + 2$, respectively, such that* QCSP(Δ_i) *is polynomially equivalent to the following problem:*

i=1 *Given an instance of* QCSP(Γ_1) *and instance of an NP-complete problem; decide whether both of them hold, i.e.* QCSP(Γ_1) $\wedge$ *NP.*
i=2 *Given an instance of* QCSP(Γ_2) *and an instance of* QCSP(Γ_3)*; decide whether both of them hold, i.e.* QCSP(Γ_2) $\wedge$ QCSP(Γ_3).

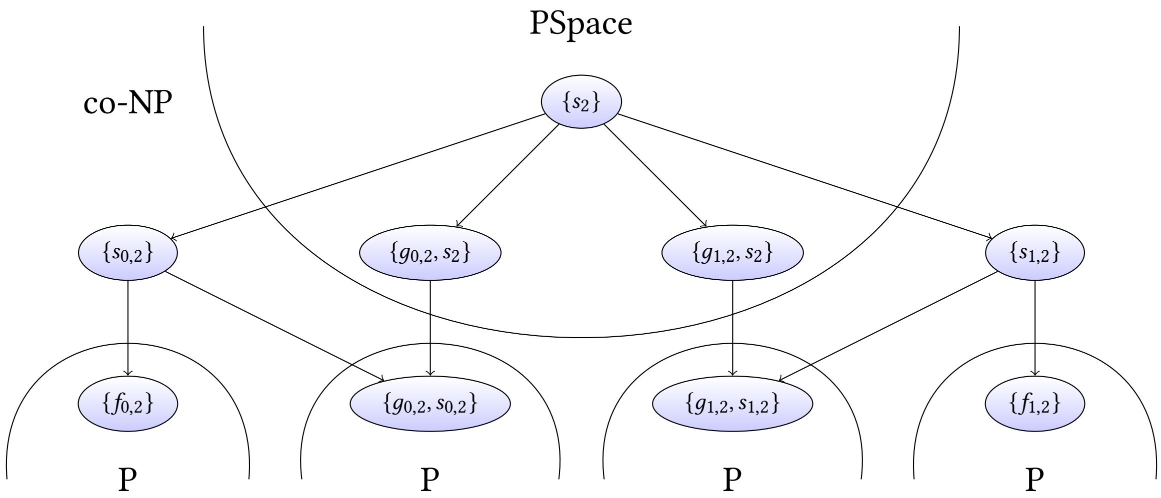

Figure 1: Constraint languages defined as invariants of sets of operations and their complexity.

i=3 *Given* $n > 0$, *instances* $I_1, \ldots, I_n$ *of* $\mathrm{QCSP}(\Gamma_2)$, *and instances* $J_1, \ldots, J_n$ *of* $\mathrm{CSP}(\Gamma_3)$*; decide whether* $(I_1 \vee J_1) \wedge \cdots \wedge (I_n \vee J_n)$ *holds, i.e.* $(\mathrm{QCSP}(\Gamma_2) \vee \mathrm{CSP}(\Gamma_3)) \wedge \cdots \wedge (\mathrm{QCSP}(\Gamma_2) \vee \mathrm{CSP}(\Gamma_3))$.

i=4 *Given* $n > 0$, *instances* $I_1, \ldots, I_n$ *of* $\mathrm{QCSP}(\Gamma_2)$, *and instances* $J_1, \ldots, J_n$ *of* $\mathrm{QCSP}(\Gamma_3)$*; decide whether* $(I_1 \vee J_1) \wedge \cdots \wedge (I_n \vee J_n)$ *holds, i.e.* $(\mathrm{QCSP}(\Gamma_2) \vee \mathrm{QCSP}(\Gamma_3)) \wedge \cdots \wedge (\mathrm{QCSP}(\Gamma_2) \vee \mathrm{QCSP}(\Gamma_3))$.

Proof. The proof for $i = 1$, $i = 2$, $i = 3$, and $i = 4$ follows from Lemmas 13, 16, 15, and 14, respectively. □

Corollary 8. *There exists a finite constraint language* Γ *on a 4-element domain such that* $\mathrm{QCSP}(\Gamma)$ *is DP-complete (where DP = NP* $\wedge$ *co-NP from Boolean hierarchy).*

Proof. By Theorem 6, there exists a constraint language Γ_1 on a 3-element domain with constants such that $\mathrm{QCSP}(\Gamma_1)$ is co-NP-complete. Applying Theorem 7 with $i = 1$ to Γ_1 we obtain a constraint language Γ on a 4-element domain such that $\mathrm{QCSP}(\Gamma)$ is polynomially equivalent to DP. □

The complexity class Θ_2^{P} (see [20] and references therein) admits various definitions, one of which is that it allows a Turing machine polynomial time with a logarithmic number of calls to an NP oracle. A condition proved equivalent to this, through Theorems 4 and 7 of [7], is as follows. In this theorem $i \leqslant p(|x|)$ indicates i is a positive integer smaller than $p(|x|)$, where x is a string of length $|x|$.

Theorem 9 ([7])**.** *Every predicate in* Θ_2^{P} *can be defined by a formula of the form* $\exists i \leqslant p(|x|)\ A(i,x) \wedge \neg B(i,x)$ *as well as by a formula of the form* $\forall i \leqslant p'(|x|)\ A'(i,x) \vee \neg B'(i,x)$ *where* A, B, A', B' *are NP-predicates and* p, p' *are polynomials.*

The second (universal) characterization will play the key role in the following observation.

Corollary 10. *There exists a finite constraint language* Γ *on a 10-element domain such that* $\mathrm{QCSP}(\Gamma)$ *is* Θ_2^{P}*-complete.*

Proof. By Theorem 6, there exists a constraint language Γ_1 on a 3-element domain with constants such that $\mathrm{QCSP}(\Gamma_1)$ is co-NP-complete. Choose a constraint language Γ_2 on a 2-element domain such that $\mathrm{CSP}(\Gamma_2)$ is NP-complete. Using item 3 of Theorem 7, we construct a constraint language Γ so that $\mathrm{QCSP}(\Gamma)$ is equivalent to the truth of $(I_1 \vee J_1) \wedge \cdots \wedge (I_n \vee J_n)$, where $I_1, \ldots, I_n$ are instances of $\mathrm{QCSP}(\Gamma_1)$ and $J_1, \ldots, J_n$ are instances of $\mathrm{CSP}(\Gamma_2)$.

To prove membership of $\mathrm{QCSP}(\Gamma)$ in Θ_2^{P}, we use the second characterization of Theorem 9 together with $A'(i,x)$ indicating that J_i is a yes-instance of $\mathrm{CSP}(\Gamma_2)$ and $\neg B'(i,x)$ indicating that I_i is a yes-instance (or $B'(i,x)$ indicating I_i is a no-instance) of $\mathrm{QCSP}(\Gamma_1)$. Thus, we want i to range over numbers from 1 to n, so in the predicates $A'(i,x)$ and $\neg B'(i,x)$ we should in particular set these to be true if i is not a number from 1 to n.

To prove that $\mathrm{QCSP}(\Gamma)$ is Θ_2^{P}-complete, we use again the second formulation of characterization of Theorem 9, but this time break the universal quantification into a conjunction of length $p'(|x|)$. □

3 PRELIMINARIES

Let $[n] = \{1, \ldots, n\}$. We identify a constraint language Γ with a set of relations over a fixed finite domain D. We may also think of this as a first-order relational structure. If Φ is a first-order formula including $x_1, \ldots, x_n$ among its free variables and not containing $y_1, \ldots, y_n$ in any capacity, then $\Phi^{x_1,\ldots,x_n}_{y_1,\ldots,y_n}$ is the result of substituting the free occurrences of $x_1, \ldots, x_n$ by $y_1, \ldots, y_n$, respectively. If I is an instance of $\mathrm{QCSP}(\Gamma)$, then $\mathrm{Var}(I)$ refers to the variables mentioned in I. If Q is a quantifier from $\{\exists, \forall\}$ then $\overline{Q}$ is its de Morgan dual, that is the unique quantifier from $\{\exists, \forall\} \setminus \{Q\}$.

We always may assume that an instance of $\mathrm{QCSP}(\Gamma)$ is of the prenex form $\forall x_1 \exists y_1 \forall x_2 \exists y_2 \ldots \forall x_n \exists y_n \Phi$, since if it is not it may readily be brought into such a form in polynomial time. Then a solution is a sequence of (Skolem) functions $f_1, \ldots, f_n$ such that

$$(x_1, f(x_1), x_2, f_2(x_1,x_2), \ldots, x_n, f_n(x_1, \ldots, x_n))$$

is a solution of Φ for all $x_1, \ldots, x_n$ (i.e. $y_i = f_i(x_1, \ldots, x_i)$). This belies a (Hintikka) game semantics for the truth of a QCSP instance in which a player called Universal plays the universal variables and a player called Existential plays the existential variables, one after another, from the outside in. The Skolem functions above give a strategy for Existential. In our proofs we may occasionally revert to a game-theoretical parlance.

An *algebra* $\mathbb{A}$ consists of domain and a set of operations defined on that domain. The most important type of algebra in this paper is a clone. Let $\mathrm{Clo}(G)$ be the *clone* generated by the set of operations G, that is the closure of G under the addition of projections and composition, where the composition of a k-ary operation f and m-ary operations $g_1, \dots, g_k$ is the m-ary operation defined by $f(g_1, \dots, g_k)$.

In general with our operators, if the argument is a singleton set, we omit the curly brackets. A *subalgebra* of $\mathbb{A}$ consists of a subset D of the domain of $\mathbb{A}$, that is preserved by all the operations of G, together with all the operations of $\mathbb{A}$ restricted to D. A *congruence* on an algebra $\mathbb{A}$ is an equivalence relation $\sim$ on its domain so that, for each k-ary operation f in $\mathbb{A}$, $f(x_1, \dots, x_k) \sim f(y_1, \dots, y_k)$ whenever $x_1 \sim y_1, \dots, x_k \sim y_k$. We can quotient $\mathbb{A}$ by $\sim$ in the obvious way to obtain a new algebra that we describe as a *homomorphic image* of $\mathbb{A}$. A *factor* of $\mathbb{A}$ is a subalgebra of a homomorphic image of $\mathbb{A}$.

A formula of the form $\exists y_1 \dots \exists y_n \Phi$, where Φ is a conjunction of relations from Γ is called *a positive primitive formula (pp-formula) over* Γ. If $R(x_1, \dots, x_n) = \exists y_1 \dots \exists y_n \Phi$, then we say that R is *pp-defined* by $\exists y_1 \dots \exists y_n \Phi$, and $\exists y_1 \dots \exists y_n \Phi$ is called *a pp-definition.* Note that if a relation R is pp-definable over Γ then it is preserved by any operation $f \in \mathrm{Pol}(\Gamma)$ [3, 17].

In a pp-formula we allow always, except for Section 5, the use of constants from the domain. Note that using constants is equivalent to having all unary relations $x = c$ in our constraint language On the algebraic side, this corresponds to assuming all polymorphism operations are idempotent. For a conjunctive formula Φ by $\Phi(x_1, \dots, x_n)$ we denote the n-ary relation defined by a pp-formula where all variables except $x_1, \dots, x_n$ are existentially quantified. Equivalently, $\Phi(x_1, \dots, x_n)$ is the set of all tuples $(a_1, \dots, a_n)$ such that Φ has a solution with $(x_1, \dots, x_n) = (a_1, \dots, a_n)$.

For a k-ary relation R and a set of coordinates $B \subset [k]$, define $\mathrm{pr}_B(R)$ to be the $|B|$-ary relation obtained from R by projecting onto B, or equivalently, existentially quantifying variables at positions $[k] \setminus B$.

For a tuple α by $\alpha(n)$ we denote the n-th element of α. We define relations by matrices where the columns list the tuples.

Let α and β be strict subsets of D so that $\alpha \cup \beta = D$. The most interesting cases arise when $\alpha \cap \beta = \varnothing$ but we will not insist on this at this point. An n-ary operation f is *$\alpha\beta$-projective* if there exists $i \in [n]$ so that $f(x_1, \dots, x_n) \in \alpha$, if $x_i \in \alpha$, and $f(x_1, \dots, x_n) \in \beta$, if $x_i \in \beta$. In this case, we may say that f is $\alpha\beta$-projective to coordinate i. It is now known that an idempotent algebra $\mathbb{A}$ over domain D has EGP iff there exists α and β, strict subsets of D, so that all operations of $\mathbb{A}$ are $\alpha\beta$-projective [25].

4 THE CONSERVATIVE CASE

In this section we prove Theorem 4 describing the complexity of $\mathrm{QCSP}(\Gamma)$ for conservative constraint languages Γ, i.e. languages containing all unary relations. As it was mentioned in the introduction, if $\mathrm{Pol}(\Gamma)$ has the PGP property then we can reduce $\mathrm{QCSP}(\Gamma)$ to several copies of CSP. Thus, the only open question was the complexity for the EGP case. Here we will use the following fact from [9].

Lemma 11 ([9]). *Suppose Γ is a constraint language on domain D with constants, $\mathrm{Pol}(\Gamma)$ has the EGP property. Then there exist $\alpha, \beta \subsetneq D$ such that $\alpha \cup \beta = D$ and τ_n (as in Definition 1) is pp-definable from Γ for every $n \geqslant 1$.*

It turns out that if Γ contains all unary relations then two copies of τ_k can be composed to define the relation $\tau_{2(k-1)}$ as follows. Choose $0 \in \alpha \setminus \beta$ and $1 \in \beta \setminus \alpha$, then

$$\begin{aligned}\tau_{2(k-1)}&(x_1, y_1, z_1 \dots, x_{2(k-1)}, y_{2(k-1)}, z_{2(k-1)}) = \\ &\exists w\ \tau_k(x_1, y_1, z_1 \dots, x_{k-1}, y_{k-1}, z_{k-1}, 0, 0, w) \wedge \\ &\quad \tau_k(x_k, y_k, z_k, \dots, x_{2(k-1)}, y_{2(k-1)}, z_{2(k-1)}, 1, 1, w) \wedge w \in \{0, 1\}.\end{aligned}$$

Identifying variables in τ_k we can derive τ_{k-1}, therefore τ_k is pp-definable from τ_j and unary relations whenever $k \geqslant j \geqslant 3$.

Lemma 12. *There is a polynomially (in k) computable pp-definition of τ_k from τ_3 and unary relations.*

Proof. As above we can define $\tau_{2(k-1)}$ in a recursive fashion using two copies of τ_k plus a single new existential quantifier whose variable is restricted to being on domain $\{0, 1\}$. Note that in the recursive pp-definition of τ_k over τ_3 every variable that is not quantified appears just once, each quantified variable appears three times, and most variables are not quantified. Therefore, our recursive scheme gives a polynomially computable pp-definition of τ_k. □

We are now in a position to prove Theorem 4, whose statement we recall.

Theorem 4. *Let Γ be a finite constraint language with all unary relations. If $\mathrm{Pol}(\Gamma)$ has PGP, then $\mathrm{QCSP}(\Gamma)$ is in NP. If Γ further admits a WNU polymorphism, then $\mathrm{QCSP}(\Gamma)$ is in P, else it is NP-complete. Otherwise, $\mathrm{Pol}(\Gamma)$ has EGP and $\mathrm{QCSP}(\Gamma)$ is PSpace-complete.*

Proof. Assume Γ is a finite constraint language with all unary relations. Suppose $\mathrm{Pol}(\Gamma)$ has PGP. Then we know from Theorem 2 that $\mathrm{QCSP}(\Gamma)$ reduces to a polynomial number of instances of $\mathrm{CSP}(\Gamma)$. It follows from Theorem 1 that if Γ admits a WNU then $\mathrm{QCSP}(\Gamma)$ is in P, otherwise $\mathrm{QCSP}(\Gamma)$ is NP-complete.

Suppose now $\mathrm{Pol}(\Gamma)$ has EGP. By Lemma 11 there exist α, β as in Definition 1 such that τ_3 is pp-definable from Γ. Combining this with Lemma 12 we conclude that there are polynomially (in k) computable pp-definitions of τ_k in Γ. We will reduce from the complement of *Quantified Not-All-Equal 3-Satisfiability* (QNAE3SAT) which is known to be PSpace-complete (see, e.g., [23]). From an instance $\phi := \neg\forall x_1 \exists y_1 \dots \forall x_n \exists y_n\ \Phi$ of co-QNAE3SAT, where $\Phi := \mathrm{NAE}_3(z_1^1, z_1^2, z_1^3) \wedge \dots \wedge \mathrm{NAE}_3(z_k^1, z_k^2, z_k^3)$ and $z_1^1, z_1^2, z_1^3, \dots, z_k^1, z_k^2, z_k^3 \in \{x_1, y_1, \dots, x_n, y_n\}$, we build an instance ϕ' of $\mathrm{QCSP}(\Gamma)$ as follows. Consider ϕ to be $\exists x_1 \forall y_1 \dots \exists x_n \forall y_n\ \neg\Phi$ and set

$$\begin{aligned}\phi' := \exists x_1 \forall y_1 \dots \exists x_n \forall y_n & \\ x_1, \dots, x_n \in (\alpha \setminus \beta \cup \beta \setminus \alpha)\ \wedge\ & \tau_k(z_1^1, z_1^2, z_1^3, \dots, z_k^1, z_k^2, z_k^3).\end{aligned}$$

The idea is that the set $\alpha \setminus \beta$ plays the role of 0 and $\beta \setminus \alpha$ plays the role of 1.

($\phi \in$ co-QNAE3SAT implies $\phi' \in$ QCSP(Γ).) Let the universal variables be evaluated in ϕ' and match them in ϕ according to $\alpha \setminus \beta$ being 0 and $\beta \setminus \alpha$ being 1. If a universal variable in ϕ' is evaluated in $\alpha \cap \beta$, then we can match it in ϕ w.l.o.g. to 0. Now, read a valuation of the existential variables of ϕ' from those in ϕ according to 0 becoming any fixed $d_0 \in \alpha \setminus \beta$ and 1 becoming any fixed $d_1 \in \beta \setminus \alpha$. By construction we have $\phi' \in$ QCSP(Γ).

($\phi' \in$ QCSP(Γ) implies $\phi \in$ co-QNAE3SAT.) Suppose $\phi' \in$ QCSP(Γ). We will prove $\phi \in$ co-QNAE3SAT again using the form of ϕ being $\exists x_1 \forall y_1 \ldots \exists x_n \forall y_n\ \neg\Phi$. Let the universal variables be evaluated in ϕ and match them in ϕ' according to 0 becoming any fixed $d_0 \in \alpha \setminus \beta$ and 1 becoming any fixed $d_1 \in \beta \setminus \alpha$. Now, read a valuation of the existential variables of ϕ from ϕ' according to $\alpha \setminus \beta$ being 0 and $\beta \setminus \alpha$ being 1. By construction we have $\phi \in$ co-QNAE3SAT. ◻

5 QCSP MONSTERS

This section explains the building of monsters with greater than a 3-element domain. It has no bearing on the 3-element classification.

Lemma 13. *Suppose Γ is a finite constraint language on a set A containing $x = a$. Then there exists a constraint language Γ' on a domain of size $|A|+1$ such that* QCSP(Γ') *is polynomially equivalent to* QCSP(Γ) $\wedge$ *NP, that is the following decision problem: given an instance of* QCSP(Γ) *and an instance of some NP-complete problem; decide whether both of them hold.*

Proof. Choose an element $a \in A$ and an element $a' \notin A$. Put $A' = A \cup \{a'\}$.

Put $\phi(x) = \begin{cases} x, & \text{if } x \in A \\ a, & \text{if } x = a' \end{cases}$. We assign a relation R' on the set A' to every $R \in \Gamma$ as follows: $R' = \{(a_1, \ldots, a_h) \mid (\phi(a_1), \ldots, \phi(a_h)) \in R\}$. Let $\mathrm{NAE}_3 \subseteq \{a, a'\}^3$ be the ternary relation containing all tuples on $\{a, a'\}$ except for $(a, a, a), (a', a', a')$. Let $\Gamma' = \{R' \mid R \in \Gamma\} \cup \{\mathrm{NAE}_3\}$.

Suppose I is an instance of QCSP(Γ) and J is an instance of CSP($\{\mathrm{NAE}_3\}$), which is an NP-complete problem. If we replace every relation R from Γ by the corresponding relation R', we get an instance I' that is equivalent to I. Then the instance $I' \wedge J$ can be viewed as an instance of QCSP(Γ') that is equivalent to $I \wedge J$.

Suppose I' is an instance of QCSP(Γ'). W.l.o.g. we will assume that no variable appearing in an NAE_3 relation is universally quantified, else this is a no-instance of QCSP(Γ') and can be reduced to a fixed no-instance (e.g.) J of CSP($\{\mathrm{NAE}_3\}$). Now, we define an instance I of QCSP(Γ) and an instance J of CSP($\{\mathrm{NAE}_3\}$) as follows. I is obtained from I' by replacement of all relations R' by the corresponding R and NAE_3 by $\{(a, a, a)\}$. Since Γ contains $x = a$, I is an instance of QCSP(Γ). The instance J consists of the NAE_3-part of I' which is a CSP as we already assumed it contains no universal variables. Now, to see $I' \in$ QCSP(Γ') iff $I \in$ QCSP(Γ) and $J \in$ CSP($\{\mathrm{NAE}_3\}$) it is enough to observe that QCSP(Γ) and QCSP($\Gamma' \setminus \{\mathrm{NAE}_3\}$) are equivalent on all instances. ◻

Lemma 14. *Suppose Γ_1 and Γ_2 are finite constraint languages on sets A_1 and A_2 respectively. Then there exists a constraint language Γ on a domain of size $|A_1| \cdot |A_2| + |A_1| + |A_2| + 2$ such that* QCSP(Γ) *is polynomially equivalent to* (QCSP(Γ_1) $\vee$ QCSP(Γ_2)) $\wedge \cdots \wedge$ (QCSP(Γ_1) $\vee$ QCSP(Γ_2)), *i.e. the following decision problem: given n, instances $I_1, \ldots, I_n$ of* QCSP(Γ_1)*, and instances $J_1, \ldots, J_n$ of* QCSP(Γ_2)*; decide whether $(I_1 \vee J_1) \wedge \cdots \wedge (I_n \vee J_n)$ holds.*

Proof. Assume that $A_1 \cap A_2 = \varnothing$, $a_1, a_2 \notin A_1 \cup A_2$. Let

$$\begin{aligned} A =& (A_1 \times A_2) \cup A_1 \cup A_2 \cup \{a_1, a_2\}, \\ \sigma =& (A_1 \times \{a_1\}) \cup (\{a_2\} \times A_2), \\ \sigma_1 =& \{(a, (a, b)) \mid a \in A_1, b \in A_2\} \cup \\ & (\{a_2\} \times A) \cup (A_1 \times (A_1 \cup A_2 \cup \{a_1, a_2\})), \\ \sigma_2 =& \{(b, (a, b)) \mid a \in A_1, b \in A_2\} \cup \\ & (\{a_1\} \times A) \cup (A_2 \times (A_1 \cup A_2 \cup \{a_1, a_2\})), \\ \Gamma =& \{R \cup \{(a_2, \ldots, a_2)\} \mid R \in \Gamma_1\} \cup \\ & \{R \cup \{(a_1, \ldots, a_1)\} \mid R \in \Gamma_2\} \cup \{\sigma_1, \sigma_2, \sigma\}. \end{aligned}$$

Suppose we have an instance I_1 of QCSP(Γ_1) and an instance I_2 of QCSP(Γ_2). W.l.o.g. we will assume that neither I_1 nor I_2 is empty. We will explain how to build an instance J of QCSP(Γ). Let $x_1, \ldots, x_n$ be all universally quantified variables of I_1.

We replace every atomic relation R of I_1 by $R \cup \{(a_2, \ldots, a_2)\}$ and add relational constraints $\sigma_1(x_i, y_i)$ for every $i \in \{1, \ldots, n\}$. Also we replace $\forall x_i$ by $\forall y_i \exists x_i$ for every $i \in \{1, \ldots, n\}$. The result we denote by I_1'. Similarly, but with a_1 instead of a_2 and σ_2 instead of σ_1 we define I_2'. We claim that the sentence J defined by

$$I_1' \wedge I_2' \wedge \bigwedge_{u \in \mathrm{Var}(I_1), v \in \mathrm{Var}(I_2)} \sigma(u, v)$$

(we move all the quantifiers to the left part after joining) holds if and only if I_1 holds or I_2 holds. W.l.o.g. we will henceforth assume the first variable in J is existential (if necessary we could enforce this by a dummy existential variable at the beginning of I_1).

($I_1 \in$ QCSP(Γ_1) $\vee$ $I_2 \in$ QCSP(Γ_2) implies $J \in$ QCSP(Γ).) W.l.o.g. $I_1 \in$ QCSP(Γ_1). Evaluate all variables of relations coming from Γ_2 as a_1. Evaluate all first variables in relations σ_2 as a_1. Evaluate all other variables of J according to the witnesses for I_1.

($J \in$ QCSP(Γ) implies $I_1 \in$ QCSP(Γ_1) $\vee I_2 \in$ QCSP(Γ_2).) Consider the witnessing of $J \in$ QCSP(Γ) where universal variables are played only on elements of the form (a, b) where $a \in A_1$, $b \in A_2$. The first variable x of J is existential and indeed is associated with I_1. This must be evaluated in A_1 or as a_2. If x is evaluated in A_1 then the σ constraints force all variables associated with I_2 to now be a_1 and thus all variables associated with I_1 to be in A_1. We can now witness $I_1 \in$ QCSP(Γ_1) where the universal (a, b) corresponds to a. If x is evaluated to a_2, then the σ constraints force all variables associated with I_2 to now be in A_2 and thus all variables associated with I_1 to be in a_2. We can now witness $I_2 \in$ QCSP(Γ_2) where the universal (a, b) corresponds to b.

We can reduce a more complicated set of instances $I_1, \ldots, I_n$ of QCSP(Γ_1) and $J_1, \ldots, J_n$ of QCSP(Γ_2) to K in QCSP(Γ), in such a way that $K \in$ QCSP(Γ) iff ($I_1 \in$ QCSP(Γ_1) $\vee$ $J_1 \in$ QCSP(Γ_2)) $\wedge \ldots \wedge$ ($I_n \in$ QCSP(Γ_1) $\vee$ $J_n \in$ QCSP(Γ_2)) by taking the conjunction of our given reduction over each pair I_i and J_i.

Now, let us prove that any problem of QCSP(Γ) can be reduced to some conjunction of instances of QCSP(Γ_1) $\vee$ QCSP(Γ_2). Call an instance K of QCSP(Γ) *connected* if the Gaifman graph of the existential variables of K is connected. The relations that play a role in this graph are just those coming from Γ. The number of connected components of K will give the number of conjuncts $\text{QCSP}(\Gamma_1) \vee \text{QCSP}(\Gamma_2)$, and we will assume now w.l.og. that K is connected.

Notice that all variables in K are typed, in that any variable in a relation from Γ either takes on values ranging across: $A_1 \cup \{a_2\}$; or $A_2 \cup \{a_1\}$; or A. If a variable appears with more than one type but the types are consistent (i.e. one type is A and the other is one from $A_1 \cup \{a_2\}$ or $A_2 \cup \{a_1\}$) then this is because the variable appears in some σ_i in the second position. But now we could remove this σ_i constraint because the other existing type restriction to one of $A_1 \cup \{a_2\}$ or $A_2 \cup \{a_1\}$ means σ_i will always be satisfied. Furthermore, if some variable has inconsistent types or a fixed element constant appears in a position where it is forbidden due to type, then we know the instance is false. This would also be the case if a universal variable appears in any type other than A. We will now assume none of these situations occurs and we term such an input *reduced*.

We would like now to assume that K has no existential variables x in the second position in a σ_i. First we must argue that if K is reduced then Existential can witness the truth of K while never playing outside of $A_1 \cup A_2 \cup \{a_1, a_2\}$. Suppose Existential ever played outside of this set, then any element in the set could be chosen as a legitimate alternative. Indeed, Existential could only win by playing an element of the form (a, b) in the second position of some σ_i and in this circumstance the atom would be equally satisfied by any choice from $A_1 \cup A_2 \cup \{a_1, a_2\}$. Now we can make the assumption that K has no existential variables x in the second position in a σ_i because any choice among $A_1 \cup \{a_2\}$ or $A_2 \cup \{a_1\}$ satisfies this.

Suppose we have in K some $\sigma_1(x_1, y) \wedge \sigma_1(x_2, y)$, and y is universally quantified before x_1 and x_2, then adding the constraint $x_1 = x_2$ doesn't change the result. Let us do this and propagate out the innermost of x_1 and x_2.

Suppose we have $\forall y \exists x_1 \exists x_2 \ \sigma_1(x_1, y) \wedge \sigma_2(x_2, y)$, then this is equivalent to $\forall y_1 \forall y_2 \exists x_1 \exists x_2 \ \sigma_1(x_1, y_1) \wedge \sigma_2(x_2, y_2)$, and we will assume this latter form appears.

Finally, if $\sigma_1(x, y)$ appears in the instance K with x is quantified before y then it is equivalent to the substitution $x = a_2$. Similarly, for $\sigma_2(x, y)$ with x is quantified before y then it is equivalent to the substitution $x = a_1$.

We are now in a position to build an instance $K_1 \vee K_2$ of $\text{QCSP}(\Gamma_1) \vee \text{QCSP}(\Gamma_2)$. We can now split K into K_1 and K_2 based on the types of the existential variables using the following additional rule. If y is quantified before x (recall it must be universally quantified) then we may consider this enforces in K_1 universal quantification of x but restricted to A_1. Similarly, with $\sigma_2(x, y)$, and K_2 and A_2.

We claim $K \in \text{QCSP}(\Gamma)$ iff $K_1 \in \text{QCSP}(\Gamma_1)$ or $K_2 \in \text{QCSP}(\Gamma_2)$.

(Forward.) Assume the converse, then there exist winning strategies for Universal players for K_1 and K_2. We need to build a winning strategy for K. To do this we apply both strategies (choose different strategies for different variables) until the moment when the first existential variable (let it be x) is evaluated. Recall we assume existential variable x is either of type $A_1 \cup \{a_2\}$ or of type $A_2 \cup \{a_1\}$. W.l.o.g. let it be the former. If x is evaluated in A_1 then the Universal player of K uses the strategy of K_1, if it is evaluated as a_2 then we use the strategy for K_2. Since K is connected, if x is evaluated in A_1 then all variables of type $A_1 \cup \{a_2\}$ must be evaluated in A_1, while if x is evaluated as a_2 then all variables of type $A_2 \cup \{a_1\}$ must be evaluated from A_2 (because a_1 can not appear). Thus the strategy we built is a winning strategy for the Universal player in K.

(Backwards.) W.l.o.g. assume $K_1 \in \text{QCSP}(\Gamma_1)$. Evaluate all variables in K of type $A_2 \cup \{a_1\}$ to a_1. Evaluate all variables in K of type $A_1 \cup \{a_2\}$ in A_1 according to the winning strategy for $K_1 \in \text{QCSP}(\Gamma_1)$.

□

Similarly we can prove the following two lemmas.

Lemma 15. *Suppose Γ_1 and Γ_2 are finite constraint languages on sets A_1 and A_2 respectively. Then there exists a constraint language Γ on a domain of size $2 \cdot |A_1| + |A_2| + 2$ such that $\text{QCSP}(\Gamma)$ is polynomially equivalent to $(\text{QCSP}(\Gamma_1) \vee \text{CSP}(\Gamma_2)) \wedge \cdots \wedge (\text{QCSP}(\Gamma_1) \vee \text{CSP}(\Gamma_2))$, i.e. the following decision problem: given n, instances $I_1, \ldots, I_n$ of $\text{QCSP}(\Gamma_1)$, and instances $J_1, \ldots, J_n$ of $\text{CSP}(\Gamma_2)$; decide whether $(I_1 \vee J_1) \wedge \cdots \wedge (I_n \vee J_n)$ holds.*

Proof. It is sufficient to define a new language as follows. Let A_1' be a copy of A_1. For any $a \in A_1$ by a' we denote the corresponding element of A_1'. Let

$$\begin{aligned}
A =& A_1' \cup A_1 \cup A_2 \cup \{a_1, a_2\}, \\
\sigma =& (A_1 \times \{a_1\}) \cup (\{a_2\} \times A_2), \\
\sigma_1 =& \{(a, a') \mid a \in A_1\} \cup (\{a_2\} \times A) \cup (A_1 \times (A_1 \cup A_2 \cup \{a_1, a_2\})), \\
\Gamma =& \{R \cup \{(a_2, \ldots, a_2)\} \mid R \in \Gamma_1\} \cup \\
& \{R \cup \{(a_1, \ldots, a_1)\} \mid R \in \Gamma_2\} \cup \{\sigma_1, \sigma\}.
\end{aligned}$$

□

Lemma 16. *Suppose Γ_1 and Γ_2 are finite constraint languages on sets A_1 and A_2 respectively. Then there exists a constraint language Γ on a domain of size $|A_2| \cdot |A_3| + |A_2| + |A_3|$ such that $\text{QCSP}(\Gamma)$ is polynomially equivalent to $(\text{QCSP}(\Gamma_1) \wedge \text{QCSP}(\Gamma_2))$, i.e. the following decision problem: given an instance I of $\text{QCSP}(\Gamma_1)$ and an instance J of $\text{QCSP}(\Gamma_2)$); decide whether $I \wedge J$ holds.*

Proof. It is sufficient to define a new language as follows. Let

$$A = (A_1 \times A_2) \cup A_1 \cup A_2,$$

$$\sigma_1 = \{(a, (a, b)) \mid a \in A_1, b \in A_2\} \cup (A_1 \times (A_1 \cup A_2)),$$

$$\sigma_2 = \{(b, (a, b)) \mid a \in A_1, b \in A_2\} \cup (A_2 \times (A_1 \cup A_2)),$$

$$\Gamma = \Gamma_1 \cup \Gamma_2 \cup \{\sigma_1, \sigma_2\},$$

where we consider any relation from $\Gamma_1 \cup \Gamma_2$ as a relation on A.

□

6 CO-NP-COMPLETE LANGUAGE

In this section we define a constraint language Γ_0 on $A = \{0, 1, 2\}$ such that $\text{QCSP}(\Gamma_0)$ is co-NP-complete. Let

$$R_{and,2} = \begin{pmatrix} 0 & 0 & 1 & 1 & 2 & \cdot \\ 0 & 1 & 0 & 1 & \cdot & 2 \\ 0 & 0 & 0 & 1 & \cdot & \cdot \end{pmatrix}, R_{or,2} = \begin{pmatrix} 0 & 0 & 1 & 1 & 2 & \cdot \\ 0 & 1 & 0 & 1 & \cdot & 2 \\ 0 & 1 & 1 & 1 & \cdot & \cdot \end{pmatrix},$$

where by $\cdot$ we mean any element from $\{0, 1, 2\}$. Thus, these relations contain all the tuples starting with 2, all the tuples whose second element is 2, and their restriction to the set $\{0, 1\}$ gives row-wise the truth tables of AND and OR. Let $\Gamma_0 = \{R_{and,2}, R_{or,2}\}$.

Lemma 17. *$\mathrm{QCSP}(\Gamma_0)$ is co-NP-hard.*

Proof. We can compose relations $R_{and,2}$ and $R_{or,2}$ in the same way as we do with operations AND and OR. Thus, we can define n-ary AND and OR in the following way. For $n = 2, 3, 4, \dots$ put

$$R_{and,n+1}(x_1, \dots, x_n, x_{n+1}, y) = \exists z\ R_{and,n}(x_1, \dots, x_n, z) \wedge R_{and,2}(x_{n+1}, z, y),$$

$$R_{or,n+1}(x_1, \dots, x_n, x_{n+1}, y) = \exists z\ R_{or,n}(x_1, \dots, x_n, z) \wedge R_{or,2}(x_{n+1}, z, y).$$

Let us define a relation ξ_n for every n by

$$\begin{aligned}\xi_n(x_1, y_1, z_1, \dots, x_n, y_n, z_n) = \exists u \exists u_1 \dots \exists u_n \exists v \exists v_1 \dots \exists v_n \\ R_{and,3}(x_1, y_1, z_1, u_1) \wedge \dots \wedge R_{and,3}(x_n, y_n, z_n, u_n) \wedge \\ R_{or,3}(x_1, y_1, z_1, v_1) \wedge \dots \wedge R_{or,3}(x_n, y_n, z_n, v_n) \wedge \\ R_{and,n}(v_1, \dots, v_n, v) \wedge R_{or,n}(u_1, \dots, u_n, u) \wedge R_{and,2}(u, v, v).\end{aligned}$$

It follows from the definition that ξ_n contains all tuples with 2, and $\xi_n \cap \{0, 1\}^{3n}$ is defined by $\mathrm{AE}_3(x_1, y_1, z_1) \vee \mathrm{AE}_3(x_2, y_2, z_2) \vee \dots \vee \mathrm{AE}_3(x_n, y_n, z_n)$, where $\mathrm{AE}_3 = \{(0, 0, 0), (1, 1, 1)\}$. Now we may encode the complement of *Not-All-Equal 3-Satisfiability* using Γ. This complement can be expressed by a formula of the following form:

$$\forall y_1 \dots \forall y_t\ \mathrm{AE}_3(y_{i_1}, y_{i_2}, y_{i_3}) \vee \dots \vee \mathrm{AE}_3(y_{i_{3n-2}}, y_{i_{3n-1}}, y_{i_{3n}}),$$

where $i_1, \dots, i_{3n} \in \{1, 2, \dots, t\}$, which is equivalent to

$$\forall y_1 \dots \forall y_t\ \xi_n(y_{i_1}, y_{i_2}, \dots, y_{i_{3n}}).$$

Thus, we reduced a co-NP-complete problem to $\mathrm{QCSP}(\Gamma_0)$, which completes the proof. □

It remains to show that $\mathrm{QCSP}(\Gamma_0)$ is in co-NP. To do this we will prove that $\mathrm{QCSP}(\Gamma_0)$ can be reduced to a Π_2 instance of $\mathrm{QCSP}(\Gamma_0)$, which is a problem from the complexity class co-NP. Such restricted decision problem will be denoted by $\mathrm{QCSP}^2(\Gamma)$, that is, $\mathrm{QCSP}^2(\Gamma)$ is the decision problem where the input in a $\mathrm{QCSP}(\Gamma)$ is a Π_2 formula, that is a formula of the form $\forall x_1 \dots \forall x_n \exists y_1 \dots \exists y_s \Phi$.

We will show that such reduction is possible whenever a constraint language Γ is preserved by a 0-stable operation, where an operation f is called *0-stable* if $f(x, 0) = x$ and $f(x, 2) = 2$. Recall that s_2 is the semilattice operation such that $s_2(a, b) = 2$ whenever $a \neq b$.

Lemma 18. *Suppose a constraint language Γ is preserved by s_2 and a 0-stable operation h_0. Then an instance*

$$\forall x_1 \exists y_1 \forall x_2 \exists y_2 \dots \forall x_n \exists y_n \Phi$$

of $\mathrm{QCSP}(\Gamma)$ is equivalent to

$$\forall x_1 \forall x_2 \dots \forall x_n \exists\exists((\exists'\exists'\Phi_1) \wedge (\exists'\exists'\Phi_2) \wedge \dots \wedge (\exists'\exists'\Phi_n)),$$

where

$$\Phi_i = \Phi^{x_{i+1}, \dots, x_n, y_{i+1}, \dots, y_n}_{x'_{i+1}, \dots, x'_n, y'_{i+1}, \dots, y'_n} \wedge x'_{i+1} = 0 \wedge \dots \wedge x'_n = 0,$$

(note that $\Phi_n = \Phi$) and by $\exists\exists$ and $\exists'\exists'$ we mean that we add all necessary existential quantifiers for variables without primes and with primes, respectively.

Proof. (Forwards/ downwards.) If we have a solution $(f_1, \dots, f_n)$ of the original instance then it is also a solution of the new instance with the additional assignments $y'_j = f_j(x_1, \dots, x_i, 0, \dots, 0)$ and $x'_j = 0$ in the definition of Φ_i for every j.

(Backwards/ upwards.) Consider solutions of the new instance such that $y_i = f_i(x_1, \dots, x_n)$ for every i. Let N be the minimal number such that f_N depends on x_j for some $j > N$. In fact, we would like that there is some solution such that this number does not exist as then this is also a solution of the original instance. But for now assume for contradiction that such an N does exist and we choose it to be minimal among all the solutions. Since $(f_1, \dots, f_n)$ is also a solution of Φ_N, the following tuple is a solution of Φ

$$(x_1, \dots, x_N, 0, \dots, 0, f_1(x_1, \dots, x_n), \dots, f_N(x_1, \dots, x_n), h_{N+1}(x_1, \dots, x_n), \dots, h_n(x_1, \dots, x_n))$$

for every $x_1, \dots, x_n$ and some functions $h_{N+1}, \dots, h_n$. Note that we could see this tuple (with an additional term written and another omitted) rather as

$$(x_1, \dots, x_N, 0, \dots, 0, f_1(x_1), \dots, f_{N-1}(x_1, \dots, x_{N-1}), f_N(x_1, \dots, x_n), h_{N+1}(x_1, \dots, x_n), \dots)$$

as we assume f_i depends only on $x_1, \dots, x_i$ for $i < N$. Consider all the evaluations of the variables $x_{N+1}, \dots, x_n$ to obtain 3^{n-N} solutions of Φ, then apply the semilattice operation to them to obtain one solution $\alpha(x_1, \dots, x_N)$ of the form

$$(x_1, \dots, x_N, 0, \dots, 0, f_1(x_1, \dots, x_n), \dots, f_{N-1}(x_1, \dots, x_n), e_N(x_1, \dots, x_N), \dots, e_n(x_1, \dots, x_N)).$$

Note that $e_N(x_1, \dots, x_N)$ equals c if

$$f_N(x_1, \dots, x_N, a_{N+1}, \dots, a_n) = c$$

for every $a_{N+1}, \dots, a_n$, and $e_N(x_1, \dots, x_N)$ equals 2 otherwise.

It remains to apply h_0 to the tuples

$$(x_1, \dots, x_n, f_1(x_1, \dots, x_n), \dots, f_n(x_1, \dots, x_n))$$

and $\alpha(x_1, \dots, x_N)$ to obtain a solution of the instance such that f_N doesn't depend on $x_{N+1}, \dots, x_n$, which gives us a contradiction to the minimality of N over all solutions. □

The next lemma follows from Lemma 18 and the fact that if Γ is preserved by a semilattice then $\mathrm{CSP}(\Gamma)$ can be solved in polynomial time. Nevertheless, to explain how a 0-stable operation can be used in an algorithm we give an alternative proof.

Lemma 19. *Suppose a constraint language Γ is preserved by s_2 and a 0-stable operation h_0. Then $\mathrm{QCSP}(\Gamma)$ is in co-NP.*

Proof. Suppose we have an instance $\forall x_1 \exists y_1 \forall x_2 \exists y_2 \dots \forall x_n \exists y_n \Phi$. We can use an oracle to choose an appropriate value for x_1 (let this value be a_1). Then we need to find an appropriate value for y_1, such that we can use an oracle for x_2 and continue. We want to be sure that if the instance holds then it holds after fixing y_1.

To find out how to fix y_1 we solve the instance

$$\exists y_1 \exists x_2 \exists y_2 \dots \exists x_n \exists y_n \Phi \wedge x_1 = a_1 \wedge x_2 = x_3 = \dots = x_n = 0.$$

This is a CSP instance, which can be solved in polynomial time because the semilattice preserves Γ. We check whether we have a solution with $y_1 = 0$, $y_1 = 1$, $y_1 = 2$ (we solve three instances). Let Y be the set of possible values for y_1. If $|Y| = 1$, then we fix y_1 with the only value in Y. Obviously, the fixing of y_1 cannot transform the QCSP instance that holds into the instance that does not hold. If $|Y| > 1$ then $2 \in Y$ due to the semilattice polymorphism. Let the solution of the CSP instance with $y_1 = 2$ be $(x_1, y_1, x_2, y_2, \dots, x_n, y_n) = (a_1, 2, 0, c_2, \dots, 0, c_n)$. Assume that the QCSP instance has a solution

$$(a_1, f_1(a_1), x_2, f_2(x_1, x_2), \dots, x_n, f_n(x_1, \dots, x_n)).$$

Then by applying the operation h_0 to this solution and the solution $(a_1, 2, 0, c_2, \dots, 0, c_n)$, we get a (partial) solution of the QCSP(Γ) with $y_1 = 2$.

We proceed this way through the quantifier prefix, using an oracle to choose values for $x_2, \dots, x_n$, while we solve CSP instances to choose appropriate values for $y_2, y_3, \dots, y_n$. □

Lemma 20. QCSP(Γ_0) *is co-NP-complete.*

Proof. By Lemma 17, QCSP(Γ_0) is co-NP-hard. Since Γ_0 is preserved by a 0-stable operation $g(x, y) = \begin{cases} x, & \text{if } y \in \{0, 1\} \\ 2, & \text{otherwise.} \end{cases}$ and the semilattice s_2, Lemma 19 implies that QCSP(Γ_0) is in co-NP. □

7 PSPACE-COMPLETE LANGUAGE

In this section we define a constraint language Γ such that the QCSP(Γ) is PSpace-hard, Γ has a WNU polymorphism, Pol(Γ) has the EGP property, and Pol(Γ) is $\{0, 2\}\{1, 2\}$-projective. This constraint language is interesting because it is very simple but the proof of PSpace-hardness for this concrete language reveals the main idea of the proof for any PSpace-hard constraint language on a 3-element domain.

Let τ be a ternary relation on $\{0, 1, 2\}$ consisting of all tuples (a, b, c) such that $\{a, b, c\} \neq \{0, 1\}$. Then the complement to τ is equal to NAE$_3$, where NAE$_3 = \{0, 1\}^3 \setminus \{(0, 0, 0), (1, 1, 1)\}$. Put

$$\sigma(x_1, x_2, x_3, y_1, y_2) = (y_1, y_2 \in \{0, 2\}) \wedge (\tau(x_1, x_2, x_3) \vee (y_1 = y_2)).$$

Let $\Gamma = \{\sigma, x = 0, x = 1, x = 2\}$.

The semilattice-without-unit s_2, which is a WNU, preserves Γ.

Lemma 21. Pol(Γ) *is* $\{0, 1\}\{0, 2\}$*-projective.*

Proof. The relation $\sigma_n(x_1, x_1', x_2, x_2', \dots, x_n, x_n')$ is defined by

$$\exists y_0 \exists y_1 \dots \exists y_n \left(\bigwedge_{i=1}^{n} \sigma(x_i, x_i, x_i', y_{i-1}, y_i) \wedge y_0 = 0 \wedge y_n = 2 \right).$$

Then, by Lemma 14 from [25], Pol(Γ) is $\{0, 1\}\{0, 2\}$-projective. □

Hence, by Theorem 3 from [25], Pol(Γ) has the EGP property.

Lemma 22. *The problem* QCSP(Γ) *is PSpace-hard.*

Proof. The reduction will be from the complement of (monotone) *Quantified Not-All-Equal 3-Satisfiability* (co-QNAE3SAT) which is co-PSpace-hard (see [23]) and consequently also PSpace-hard (as PSpace is closed under complement). Consider an instance of co-QNAE3SAT

$$\neg Q_1 x_1 Q_2 x_2 \dots Q_n x_n (\text{NAE}_3(z_1^1, z_1^2, z_1^3) \wedge \dots \wedge \text{NAE}_3(z_k^1, z_k^2, z_k^3))$$

where $z_1^1, z_1^2, z_1^3, \dots, z_k^1, z_k^2, z_k^3 \in \{x_1, \dots, x_n\}$, which is equivalent to

$$\overline{Q}_1 x_1 \overline{Q}_2 x_2 \dots \overline{Q}_n x_n \left(\text{AE}_3(z_1^1, z_1^2, z_1^3) \vee \dots \vee \text{AE}_3(z_k^1, z_k^2, z_k^3) \right).$$

By Φ_n we denote the inner part of the above sentence without quantifiers. By Φ_s, where $s \in \{0, 1, \dots, n\}$, we denote the formula $\overline{Q}_{s+1} x_{s+1} \dots \overline{Q}_n x_n \Phi_n$. We will define a recursive procedure giving a formula Ω_s over Γ satisfying the following properties:

(1) the only free variables of Ω_s are $x_1, \dots, x_s, y_\ell, y_m$, where $\ell < m$ (ℓ and m are different for different s);
(2) Ω_s holds if $y_\ell = y_m \in \{0, 2\}$, and holds if $x_i = 2$ for some $i \in [s]$ and x_i appears in Φ_n;
(3) Ω_s is equivalent to Φ_s if $(x_1, \dots, x_s) \in \{0, 1\}^s$ and $y_\ell \neq y_m$.

Put $\Omega_n := \exists y_1 \dots \exists y_{k-1} \bigwedge_{i=1}^{k} \sigma(z_i^1, z_i^2, z_i^3, y_{i-1}, y_i)$. If we put $y_0 = 0$ and $y_k = 2$, then to satisfy the above formula we need some tuple (z_i^1, z_i^2, z_i^3) to be from τ, which implies on $\{0, 1\}$ that this tuple is from AE$_3$. Hence Ω_n and Φ_n satisfy the above properties (1)-(3).

Let us show how to build Ω_{s-1} from Ω_s. Let ℓ and m be the minimal and maximal indices appearing in the y variables of Ω_s, respectively. Note that $\ell \leqslant 0$ and $m > 0$, and that typically ℓ decreases and m increases during our construction.

- If $\overline{Q}_s$ is the universal quantifier then put $\Omega_{s-1} = \forall x_s \Omega_s$
- If $\overline{Q}_s$ is the existential quantifier then put

$$\Omega_{s-1} = \exists y_\ell \forall x_s \exists y_m \Omega_s \wedge \sigma(x_s, 0, 0, y_{\ell-1}, y_m) \wedge \sigma(x_s, 1, 1, y_{m+1}, y_m)$$

Let us show by induction that Ω_{s-1} satisfies the properties (1)-(3) starting with $s = n$. Assume that $\overline{Q}_s$ is the universal quantifier. The properties (1) and (2) follow from the inductive assumption and the construction. The Property (3) follows from the fact that Ω_s holds on all tuples with $x_s = 2$ or x_s does not appear in Φ_n.

Assume that $\overline{Q}_s$ is the existential quantifier. The property (1) follows from the construction. Let us show the property (2). Suppose $x_i = 2$ for some $i \in [s-1]$ and x_i appears in Φ_n. By the inductive assumption Ω_s holds. To satisfy Ω_{s-1} we put any value to y_ℓ, put $y_m = y_{\ell-1}$ if $x_s = 1$, and put $y_m = y_{m+1}$ if $x_s \neq 1$. Suppose $y_{\ell-1} = y_{m+1}$, then to satisfy Ω_{s-1} we put $y_\ell = y_m = y_{\ell-1}$.

Let us show the property (3) for Ω_{s-1}. Consider $y_{\ell-1} \neq y_{m+1}$ and a tuple $(x_1, \dots, x_{s-1}) \in \{0, 1\}^{s-1}$.

(Φ_{s-1} implies Ω_{s-1}). Let Existential choose $x_s = 0$ in Φ_{s-1}. Then Existential in Ω_{s-1} puts $y_\ell = y_{\ell-1}$. If Universal chooses $x_s = 0$, then Existential plays $y_m = y_{m+1}$. Since Φ_s holds on $x_s = 0$, Ω_s holds. Since $x_s = 0$, $\sigma(x_s, 0, 0, y_{\ell-1}, y_m)$ holds. Since $y_m = y_{m+1}$, $\sigma(x_s, 1, 1, y_{m+1}, y_m)$ holds. Thus, Ω_{s-1} holds. If Universal chooses $x_s \in \{1, 2\}$, then Existential plays $y_m = y_{\ell-1}$. Since $y_m = y_\ell$, Ω_s holds. Since $y_{\ell-1} = y_m$, $\sigma(x_s, 0, 0, y_{\ell-1}, y_m)$ holds. Since $x_s \in \{1, 2\}$, $\sigma(x_s, 1, 1, y_{m+1}, y_m)$ holds. Hence, Ω_{s-1} holds.

Let Existential choose $x_s = 1$ in Φ_{s-1}. Then Existential in Ω_{s-1} puts $y_\ell = y_{m+1}$. If Universal chooses $x_s \in \{0, 2\}$, then Existential plays $y_m = y_{m+1}$. Since $y_m = y_\ell$, Ω_s holds. Since $x_s \in \{0, 2\}$, the constraint $\sigma(x_s, 0, 0, y_{\ell-1}, y_m)$ holds. Since $y_m = y_{m+1}$, the constraint $\sigma(x_s, 1, 1, y_{m+1}, y_m)$ holds. Thus, Ω_{s-1} holds. If Universal chooses $x_s = 1$, then Existential plays $y_m = y_{\ell-1}$. Since Φ_s holds on $x_s = 1$, Ω_s holds. Since $y_{\ell-1} = y_m$, $\sigma(x_s, 0, 0, y_{\ell-1}, y_m)$ holds. Since $x_s = 1$, the constraint $\sigma(x_s, 1, 1, y_{m+1}, y_m)$ holds. Hence, Ω_{s-1} holds.

(Ω_{s-1} implies Φ_{s-1}). Assume that Existential chooses $y_\ell = y_{\ell-1}$. Let Universal choose $x_s = 0$. To satisfy $\sigma(x_s, 1, 1, y_{m+1}, y_m)$ Existential has to choose $y_m = y_{m+1}$. Then for $y_{\ell-1} \neq y_{m+1}$ we have $y_\ell \neq y_m$, which by the property (3) for Ω_s implies that Φ_s holds on $x_s = 0$.

Assume that Existential chooses $y_\ell = y_{m+1}$. Let Universal choose $x_s = 1$. To satisfy $\sigma(x_s, 0, 0, y_{\ell-1}, y_m)$ Existential has to choose $y_m = y_{\ell-1}$. Then for $y_{\ell-1} \neq y_{m+1}$ we have $y_\ell \neq y_m$, which by the property (3) for Ω_s implies that Φ_s holds on $x_s = 1$.

Noting that $y_{\ell-1} \neq y_{m+1}$, and $y_\ell, y_{\ell-1}, y_{m+1} \in \{0,2\}$, we exhausted all possibilities for y_ℓ and in both cases found an appropriate evaluation of x_s, which completes the proof of the property (3) for Ω_{s-1}. Since Ω_0 is an instance of QCSP(Γ), the property (3) for Ω_0 implies that Ω_0 and Φ_0 are equivalent, and Φ_0 is the original instance of co-QNAE3SAT. □

8 NEW TRACTABLE LANGUAGE 1

In this section we will define a constraint language Γ on $A = \{0,1,2\}$ consisting of just 2 relations and constants such that Pol(Γ) has the EGP property but every pp-definition of τ_n (see Definition 1) has at least 2^n existential quantifiers. Moreover, we will show that QCSP(Γ) can be solved in polynomial time.

Let $R_{and,2} = \begin{pmatrix} 0 & 0 & 1 & 1 & 2 & \cdot \\ 0 & 1 & 0 & 1 & \cdot & 2 \\ 0 & 0 & 0 & 1 & \cdot & \cdot \end{pmatrix}, \delta = \begin{pmatrix} \cdot & 1 & 2 \\ 0 & 2 & 2 \end{pmatrix}$, where by $\cdot$ we mean any element from $\{0,1,2\}$. Let $\Gamma = \{R_{and,2}, \delta, \{0\}, \{1\}, \{2\}\}$.

Recall that here τ_n is the $3n$-ary relation defined by

$$\{(x_1, y_1, z_1, x_2, y_2, z_2, \dots, x_n, y_n, z_n) \mid \exists i \colon \{0,1\} \nsubseteq \{x_i, y_i, z_i\}\}.$$

By σ_n we denote the $2n$-ary relation defined by

$$\{(x_1, y_1, x_2, y_2, \dots, x_n, y_n) \mid \exists i \colon \{x_i, y_i\} \neq \{0,1\}\}.$$

Note τ_n can be pp-defined from σ_n but the obvious definition is of size exponential in n (see [9]). At the same time, σ_n can be pp-defined from τ_n by identification of variables.

The relation ρ of arity $2n$ omitting just one tuple 1^n0^n can be pp-defined over Γ as follows. First, as usual, we define an n-ary "and" by the following recursive formula

$$R_{and,n+1}(x_1, \dots, x_n, x_{n+1}, y) = \exists z\ R_{and,n}(x_1, \dots, x_n, z) \wedge R_{and,2}(x_{n+1}, z, y).$$

Then ρ can be defined by

$$\rho(x_1, \dots, x_n, y_1, \dots, y_n) = \exists y_1' \dots \exists y_n' \exists z \exists t\ R_{and,n}(x_1, \dots, x_n, z) \wedge \delta(y_1, y_1') \wedge \dots \wedge \delta(y_n, y_n') \wedge R_{and,n}(y_1', \dots, y_n', t) \wedge R_{and,2}(z, z, t).$$

As a conjunction of such relations with permuted variables we can define the relation σ_n but this definition will be of exponential size. Then, we know from [25] that Pol(Γ) has the EGP property, and from [9] that τ_n can be pp-defined from Γ. Below we will prove that any pp-definition of σ_n and τ_n is of exponential size, as well as the fact that QCSP(Γ) can be solved in polynomial time. In the following $<$ is the lexicographical order on $\{0,1\}^n$ built from $0 < 1$.

Lemma 23. *Suppose $R = \Phi(x_1, \dots, x_n)$, where Φ is a conjunctive formula over Γ, $\alpha \in \{0,1\}^n \setminus R$, there exists $\beta \in \{0,1\}^n \cap R$ such that $\beta < \alpha$ and there exists $\beta \in \{0,1\}^n \cap R$ such that $\beta > \alpha$. Then there exists a variable y in Φ, such that for $R' = \Phi(x_1, \dots, x_n, y)$ we have the following property*

$$\beta \in \{0,1\}^n \wedge (\beta < \alpha) \wedge \beta d \in R' \Rightarrow d = 0,$$

$$\beta \in \{0,1\}^n \wedge (\beta > \alpha) \wedge \beta d \in R' \Rightarrow d = 1.$$

Informally speaking, this lemma says that whenever we have a tuple outside of a relation there should be a variable in its pp-definition distinguishing between smaller and greater tuples of the relation.

Proof. For every variable y of Φ let C_y be the set of all elements d such that there exists $\beta \in \{0,1\}^n \cap R$, $\beta < \alpha$ and Φ has a solution with $y = d$ and $(x_1, \dots, x_n) = \beta$. Similarly, let D_y be the set of all elements d such that there exists $\beta \in \{0,1\}^n \cap R$, $\beta > \alpha$ and Φ has a solution with $y = d$ and $(x_1, \dots, x_n) = \beta$.

Then we assign a value $v(y)$ to every variable y in the following way: if $C_y = \{0\}$ then put $v(y) := 0$; otherwise, if $C_y \subseteq \{0,1\}$ and $D_y = \{1\}$ then put $v(y) := 1$; otherwise put $v(y) := 2$.

If $\alpha(i) = 0$ then $C_{x_i} = \{0\}$ and $v(x_i) = 0$. If $\alpha(i) = 1$ then $C_{x_i} \subseteq \{0,1\}$ and $D_{x_i} = \{1\}$, therefore $v(x_i) = 1$. Since $\alpha \notin R$, v cannot be a solution of Φ, therefore v breaks at least one of the relations in Φ. We consider several cases:

1. The corresponding relation is $y = a$ for some a. If $a = 0$ then $C_y = \{0\}$ and $v(y) = 0$, if $a = 1$ then $C_y = D_y = \{1\}$ and $v(y) = 1$, if $a = 2$ then $C_y = \{2\}$ and $v(y) = 2$. Thus, the evaluation v cannot break the relation $y = a$.
2. The corresponding relation is $R_{and,2}(y_1, y_2, y_3)$. Assume that $v(y_1) = 0$ and $v(y_2) \in \{0,1\}$. Then $C_{y_1} = \{0\}$ and $C_{y_2} \subseteq \{0,1\}$, which means that on all tuples $\beta < \alpha$ the value of y_3 should be equal to 0. Hence $C_{y_3} = \{0\}$ and $v(y_3) = 0$. If $v(y_1) = 2$ or $v(y_2) = 2$, then we cannot break the relation $R_{and,2}$. The only remaining case is when $v(y_1) = v(y_2) = 1$, which means that $C_{y_1}, C_{y_2} \subseteq \{0,1\}$ and $D_{y_1} = D_{y_2} = \{1\}$. This implies that $C_{y_3} \subseteq \{0,1\}$ and $D_{y_3} = \{1\}$. If $C_{y_3} = \{0\}$, then y_3 is the variable we were looking for. Otherwise, the evaluation of y_3 is 1, which agrees with the definition of $R_{and,2}$.
3. The corresponding relation is $\delta(y_1, y_2)$. If $v(y_1) = 0$ then $C_{y_1} = \{0\}$, and by the definition of δ we have $C_{y_2} = \{0\}$, which means that $v(y_2) = 0$. If $v(y_1) \neq 0$, it follows from the fact that $v(y_2)$ cannot be outside of $\{0,2\}$.

□

Note that s_2, $s_{0,2}$, and $g_{0,2}$ preserve Γ (see Section 2.1 for the definition). Put $h_{0,2}(x,y,z) = g_{0,2}(s_{0,2}(x_1, x_3), s_2(x_2, x_3))$, then

$$h_{0,2}(x,y,z) = \begin{cases} x, & \text{if } x = z = 0 \\ x, & \text{if } x = 1, y = z \\ 2, & \text{otherwise.} \end{cases}$$

The following lemma and corollary do not play a role in our main result but we include them for their intrinsic intriguingness as well as by way of a sanity check

Lemma 24. *Any pp-definition of σ_n over Γ, where $n \geqslant 3$, has at least 2^n variables.*

Proof. Let the pp-definition be given by a conjunctive formula Φ such that $\sigma_n = \Phi(x_1, \dots, x_{2n})$. By Lemma 23 for any $\alpha \in \{0,1\}^{2n} \setminus \sigma_n$ there should be a variable y such that if we define the relation $R' = \Phi(x_1, \dots, x_{2n}, y)$, then for every $\beta < \alpha$ (we consider only tuples from $\{0,1\}^{2n}$) we have $\beta d \in R' \Rightarrow d = 0$ and for every $\beta > \alpha$ we have $\beta d \in R' \Rightarrow d = 1$.

Assume that one variable y can be used for two different tuples $\alpha_1, \alpha_2 \in \{0,1\}^{2n} \setminus \sigma_n$. We consider two cases.

Case 1. Assume that there is i such that $\alpha_1(i) = \alpha_2(i)$. Without loss of generality we assume that $\alpha_1(1) = \alpha_2(1)$ and $\alpha_1(2n) \neq \alpha_2(2n)$. Let us define tuples $\beta_1, \beta_2, \beta_3 \in \sigma_n$.

Put $\beta_1(i) = \begin{cases} 1, & \text{if } i \in \{1,2\} \\ \alpha_1(i), & \text{otherwise} \end{cases}$, $\beta_2(i) = \begin{cases} 1, & \text{if } i \in \{3,4\} \\ \alpha_2(i), & \text{otherwise} \end{cases}$,
$\beta_3(i) = \begin{cases} \alpha_1(i), & \text{if } \alpha_1(i) = \alpha_2(i) \text{ or } i \leqslant 4. \\ 0, & \text{otherwise} \end{cases}$.

Let us show that $h_{0,2}(\beta_1, \beta_2, \beta_3) = \beta_1$. In fact, for the first two rows, reading down through the $2n$ rows of $h_{0,2}(\beta_1, \beta_2, \beta_3)$, we have $h_{0,2}(1,0,0) = h_{0,2}(1,1,1) = 1$. For the next two rows we have $h_{0,2}(0,1,0) = 0$ and $h_{0,2}(1,1,1) = 1$. For the remaining rows we either use $h_{0,2}(0,0,0) = 0$ and $h_{0,2}(1,1,1) = 1$, or $h_{0,2}(0,1,0) = 0$ and $h_{0,2}(1,0,0) = 1$.

Since $\beta_1 > \alpha_1, \beta_2 > \alpha_2, \beta_3 < \alpha_1$, by Lemma 23, y should be equal to 1 in any solution of Φ such that $(x_1, \ldots, x_{2n}) \in \{\beta_1, \beta_2\}$, and it should be equal to 0 in any solution of Φ such that $(x_1, \ldots, x_{2n}) = \beta_3$. Since $h_{0,2}(\beta_1, \beta_2, \beta_3) = \beta_1$ and $h_{0,2}(1,1,0) = 2$, we get a contradiction.

Case 2. Assume that $\alpha_1(i) \neq \alpha_2(i)$ for every i. Put

$$\beta_1(i) = \begin{cases} 1, & \text{if } i \in \{1,2\} \\ \alpha_1(i), & \text{otherwise} \end{cases}, \beta_2(i) = \begin{cases} 1, & \text{if } i \in \{1,2\} \\ \alpha_2(i), & \text{otherwise} \end{cases},$$

$$\beta_3(i) = \begin{cases} 1, & \text{if } i \in \{1,2\} \\ 0, & \text{otherwise} \end{cases}, \beta_4(i) = \begin{cases} 1, & \text{if } i \in \{3,4\} \\ \alpha_1(i), & \text{otherwise} \end{cases},$$

$$\beta_5(i) = \begin{cases} \alpha_1(i), & \text{if } i \in \{1,2\} \\ 0, & \text{otherwise} \end{cases}.$$

Since $h_{0,2}(1,1,1) = h_{0,2}(1,0,0) = 1$, $h_{0,2}(0,1,0) = 0$, we have $h_{0,2}(\beta_1, \beta_2, \beta_3) = \beta_1$. Since $\beta_1 > \alpha_1$ and $\beta_2 > \alpha_2$, by Lemma 23, y should be equal to 1 in any solution of Φ such that $(x_1, \ldots, x_{2n}) \in \{\beta_1, \beta_2\}$. Since $h_{0,2}(1,1,a) = 1$ only if $a = 1$, y should be equal to 1 on β_3 (in any solution of Φ such that $(x_1, \ldots, x_{2n}) = \beta_3$). Since $h_{0,2}(\beta_3, \beta_4, \beta_5) = \beta_3$, and y should be equal to 1 on β_4 and equal to 0 on β_5, we obtain $h_{0,2}(1,1,0) = 1$, which contradicts the definition of $h_{0,2}$.

Thus, for every tuple $\alpha \in \{0,1\}^{2n} \setminus \sigma_n$ there exists a unique variable y, which completes the proof. □

Since σ_n can be obtained from τ_n by identification of variables, we have the following corollary.

Corollary 25. *Any pp-definition of τ_n over Γ has at least 2^n variables.*

Below we present an algorithm that solves $\text{QCSP}^2(\Gamma)$ in polynomial time (see the pseudocode). By h we denote the operation defined on subsets of A by $h(B) = \begin{cases} 0, & \text{if } B = \{1\} \\ 1, & \text{otherwise} \end{cases}$. By SolveCSP we denote a polynomial algorithm, solving constraint satisfaction problem for a constraint language preserved by the semilattice operation s_2: it returns true if it has a solution, it returns false otherwise.

Lemma 26. *Function Solve$_1$ solves $\text{QCSP}^2(\Gamma)$ in polynomial time.*

Proof. First, let us show that the algorithm actually solves the problem. If the answer is false, then we found an evaluation of $(x_1, \ldots, x_n)$ such that the corresponding CSP has no solutions, which means that the answer is correct.

function Solve$_1$(Θ)
  Input: $\text{QCSP}^2(\Gamma)$ instance $\Theta = \forall x_1 \ldots \forall x_n \exists y_1 \ldots \exists y_s \Phi$.
  if $\neg$SolveCSP($\mathbf{x} = (0, \ldots, 0) \wedge \Phi$) **then return** false
    ▷ Here $\mathbf{x} = (x_1, \ldots, x_n)$
  if $\neg$SolveCSP($\mathbf{x} = (1, \ldots, 1) \wedge \Phi$) **then return** false
  for $j := 1, \ldots, s$ **do**
    for $i := 1, \ldots, n$ **do**
      $D_i := \varnothing$
      $\mathbf{c} := (\underbrace{1, \ldots, 1}_{i-1}, 0, 1, \ldots, 1)$
      for $a \in A$ **do**
        if SolveCSP($\mathbf{x} = \mathbf{c} \wedge y_j = a \wedge \Phi$) **then**
          $D_i := D_i \cup \{a\}$
      if $D_i = \varnothing$ **then return** false
    if $\neg$SolveCSP($\mathbf{x} = (h(D_1), \ldots, h(D_n)) \wedge \Phi$) **then**
      return false
  return true

Assume that the answer is true. Let $R(x_1, \ldots, x_n)$ be defined by the formula $\exists y_1 \ldots \exists y_s \Phi$. We need to prove that R is a full relation. Assume the converse. Using the semilattice operation s_2 we can generate A^n from $\{0,1\}^n$, hence $\{0,1\}^n \nsubseteq R$. Then let α be a minimal tuple from $\{0,1\}^n \setminus R$. Without loss of generality we assume that $\alpha = 1^k 0^{n-k}$. For every i we put $\alpha_i = 1^{i-1} 0 1^{n-i}$. Since α is minimal, all the tuples smaller than α should be in R (note that $(0, 0, \ldots, 0) \in R$). Then by Lemma 23 there should be a variable y such that for any $\beta < \alpha$ we have $\beta d \in R' \Rightarrow d = 0$, for any $\beta > \alpha$ we have $\beta d \in R' \Rightarrow d = 1$, where $R' = \Phi(x_1, \ldots, x_n, y)$. Since $D_i \neq \varnothing$, $\alpha_i \in R$ for every i, and for every $i > k$ we have $\alpha_i d \in R' \Rightarrow d = 1$.

Let $\beta = 0 1^{k-1} 0^{n-k}$. Note that $\beta < \alpha$ and therefore, $\beta \in R$. Assume that D_1 calculated for the variable y is equal to $\{1\}$, then $\alpha_1 d \in R' \Rightarrow d = 1$. Put $\gamma_0 = s_{0,2}(\beta, \alpha_1) = 0 1^{k-1} 2^{n-k}$. Since $s_{0,2}$ preserves R' and $s_{0,2}(0,1) = 2$, we have $\gamma_0 2 \in R'$. Put $\gamma_i = g_{0,2}(\alpha_{k+i}, \gamma_{i-1})$ for $i = 1, 2, \ldots, n-k$. Since $g_{0,2}(1,2) = 2$, we have $\gamma_i 2 \in R'$ for every i. Note that $\gamma_{n-k} = \alpha_n \in \{0,1\}^n$. We can check that $g_{0,2}(\alpha_1 1, \gamma_{n-k} 2) = \alpha_1 2$, which contradicts the fact that $D_1 = \{1\}$. In this way we can show that $D_1, \ldots, D_k$ calculated for y are not equal to $\{1\}$. We also know that the corresponding $D_{k+1}, \ldots, D_n$ are equal to $\{1\}$. Hence, the tuple $(h(D_1), \ldots, h(D_n)) = \alpha$ was checked in the algorithm, which contradicts the fact that $\alpha \notin R$.

It remains to show that the algorithm works in polynomial time. It follows from the fact that in the algorithm we just solve $3 \cdot s \cdot n + s + 2$ CSP instances over a language preserved by the semilattice operation s_2. □

Corollary 27. $\text{QCSP}(\Gamma)$ *is in P.*

Proof. Since $s_{0,2}$ is a 0-stable operation preserving Γ, Lemma 18 implies that $\text{QCSP}(\Gamma)$ can be polynomially reduced to $\text{QCSP}^2(\Gamma)$, and $\text{QCSP}^2(\Gamma)$ can be solved by the function Solve$_1$. □

9 NEW TRACTABLE LANGUAGE 2

In this section we define another constraint language Γ' on $A = \{0,1,2\}$ such that $\mathrm{Pol}(\Gamma')$ has the EGP property but every pp-definition of τ_n (see Definition 1) has at least 2^n existential quantifiers. Moreover, we will show that $\mathrm{QCSP}(\Gamma')$ can be solved in polynomial time.

Let $R'_{and,2} = \begin{pmatrix} 0 & \cdot & 1 & 1 & 2 & 2 \\ \cdot & 0 & 1 & 2 & 1 & 2 \\ 0 & 0 & 1 & \cdot & \cdot & \cdot \end{pmatrix}$, $\delta = \begin{pmatrix} 0 & 1 & 2 \\ 1 & \cdot & \cdot \end{pmatrix}$, where by $\cdot$ we denote any element from $\{0,1,2\}$.

Let $\Gamma' = \{R'_{and,2}, \delta, \{0\}, \{1\}, \{2\}\}$.

Again, recall that here τ_n the $3n$-ary relation defined by

$$\{(x_1, y_1, z_1, x_2, y_2, z_2, \dots, x_n, y_n, z_n) \mid \exists i\colon \{0,1\} \nsubseteq \{x_i, y_i, z_i\}\}.$$

By σ_n we denote the $2n$-ary relation defined by

$$\{(x_1, y_1, x_2, y_2, \dots, x_n, y_n) \mid \exists i\colon \{x_i, y_i\} \neq \{0,1\}\}.$$

Note τ_n can be pp-defined from σ_n but the obvious definition is of size exponential in n (see [9]). At the same time, σ_n can be pp-defined from τ_n by identification of variables. It follows from the following lemma that $\mathrm{Pol}(\Gamma')$ has the EGP property.

Lemma 28. *σ_n can be pp-defined over Γ'.*

Proof. Recursively we define

$$R'_{and,n+1}(x_1, \dots, x_n, x_{n+1}, y) = \exists z\, R'_{and,n}(x_1, \dots, x_n, z) \wedge R'_{and,2}(x_{n+1}, z, y),$$

$$\omega_n(x_1, y_1, x_2, y_2, \dots, x_n, y_n) = \exists u_1 \dots \exists u_n \exists z R'_{and,2n}(x_1, \dots, x_n, u_1, \dots, u_n, z) \wedge \delta'(y_1, u_1) \wedge \dots \wedge \delta'(y_n, u_n) \wedge z = 0.$$

The relation ω_n contains all the tuples but $(1,0,1,0,\dots,1,0)$. Then the relation σ_n can be represented as a conjunction of 2^n relations such that each of them is obtained from ω_n by a permutation of variables. □

We can check that Γ' is preserved by $f_{0,2}$ (see Section 2.1 for the definition). Below we will show that $\mathrm{QCSP}(\Gamma)$ is solvable in polynomial time for any constraint language $\Gamma \subseteq \mathrm{Inv}(f_{0,2})$. Note that $s_{0,2}(x,y) = f_{0,2}(x,y,y)$ and $s_2(x,y) = s_{0,2}(x, s_{0,2}(y,x))$.

Suppose $R = \Phi(x_1, \dots, x_n)$, where Φ is a conjunctive formula over a constraint language $\Gamma \subseteq \mathrm{Inv}(f_{0,2})$. For a variable y of Φ we define a partial operation $F_y(x_1, \dots, x_n)$ on $\{0,1\}$ as follows. If $\alpha \in R$ and every solution of Φ with $(x_1, \dots, x_n) = \alpha$ has $y = c$, where $c \in \{0,1\}$, then $F_y(\alpha) = c$. Otherwise we say that $F_y(\alpha)$ is not defined. We say that $\alpha \in R \cap \{0,1\}^n$ is *a minimal 1-set for a variable y* if $F_y(\alpha) = 1$, and $F_y(\beta) = 0$ for every $\beta < \alpha$ (every time we use $<$ we mean that both tuples are on $\{0,1\}$).

The following lemma proves that F_y is monotonic.

Lemma 29. *Suppose $\alpha \leqslant \beta$, $F_y(\alpha)$ and $F_y(\beta)$ are defined. Then $F_y(\alpha) \leqslant F_y(\beta)$.*

Proof. Assume the contrary, then $F_y(\alpha) = 1$ and $F_y(\beta) = 0$. We have $s_{0,2}(\beta 0, \alpha 1) = \beta 2$, which means that there exists a solution of Φ with $(x_1, \dots, x_n) = \beta$ and $y = 2$, hence $F_y(\beta)$ is not defined. Contradiction. □

Lemma 30. *There is at most one minimal 1-set for every variable y.*

Proof. Assume the contrary. Let α_1 and α_2 be two minimal 1-sets for y. It follows from the definition that α_1 and α_2 should be incomparable. Let $\alpha = \alpha_1 \wedge \alpha_2$ (by $\wedge$ we denote the conjunction on $\{0,1\}$). Then $f_{0,2}(\alpha_1 1, \alpha 0, \alpha_2 1) = \alpha_1 2$, which contradicts the fact that F_y is defined on α_1. □

Lemma 31. *Suppose $\alpha \in \{0,1\}^n \setminus R$, α contains at least two 1s, and $\beta \in R$ for every $\beta < \alpha$. Then there exists a constraint $\rho(z_1, \dots, z_l)$ in Φ and $B \subseteq \{1, \dots, l\}$ such that $\alpha = \bigvee_{i \in B} \alpha_i$, where α_i is the minimal 1-set for the variable z_i (by $\vee$ we denote the disjunction on $\{0,1\}$).*

Proof. First, to every variable y of Φ we assign a value $v(y)$ in the following way. If $F_y(\beta) = 0$ for every $\beta < \alpha$ then we put $v(y) := 0$. Otherwise, if $F_y(\beta) \in \{0,1\}$ for every $\beta < \alpha$ then we put $v(y) := 1$. Otherwise, put $v(y) := 2$.

If $\alpha(i) = 0$ then $F_{x_i}(\beta) = 0$ for every $\beta < \alpha$, which means that $v(x_i) = 0$. If $\alpha(i) = 1$ then $F_{x_i}(\beta) \in \{0,1\}$ for every $\beta < \alpha$. Since α has at least two 1, for some $\beta < \alpha$ we have $F_{x_i}(\beta) = 1$, which means that $v(x_i) = 1$. Thus we assigned the tuple α to $(x_1, \dots, x_n)$.

Since $\alpha \notin R$ the evaluation v cannot be a solution of Φ, therefore it breaks at least one constraint from Φ. Let us add to Φ all projections of all constraints we have in Φ. Thus, for every constraint $C = \rho(z_1, \dots, z_l)$ we add $\mathrm{pr}_S C$, where $S \subseteq \{z_1, \dots, z_l\}$. Obviously, when we do this, we do not change the solution set of Φ and stay in $\mathrm{Inv}(f_{0,2})$.

Choose a constraint of the minimal arity $\rho(z_1, \dots, z_l)$ that does not hold in the evaluation v, that is, $(v(z_1), \dots, v(z_l)) \notin \rho$. Let $(a_1, \dots, a_l) = (v(z_1), \dots, v(z_l))$. Since ρ is a constraint of the minimal arity, the evaluation v holds for every proper projection of $\rho(z_1, \dots, z_l)$, which means that for every i there exists b_i such that $(a_1, \dots, a_{i-1}, b_i, a_{i+1}, \dots, a_l) \in \rho$.

Assume that $(a_1, \dots, a_l)$ has two 2, that is $a_i = a_j = 2$ for $i \neq j$. Then the semilattice s_2 applied to $(a_1, \dots, a_{i-1}, b_i, a_{i+1}, \dots, a_l)$ and $(a_1, \dots, a_{j-1}, b_j, a_{j+1}, \dots, a_l)$ gives $(a_1, \dots, a_l)$, which contradicts the fact that s_2 preserves ρ.

Assume that $a_i = 2$ for some i. W.l.o.g. we assume that $a_l = 2$. By the definition, there should be a tuple $\beta < \alpha$ such that $F_{z_l}(\beta)$ is not defined. Put $c_i = F_{z_i}(\beta)$ for every $i < l$, and $c_l = 2$. By the definition of $F_{z_l}(\beta)$, there should be a solution of Φ with $(x_1, \dots, x_n) = \beta$ and $z_l = 2$, or two solutions of Φ with $(x_1, \dots, x_n) = \beta$ and $z_l = 0, 1$. Since s_2 preserves Γ, in both cases we have a solution of Φ with $(x_1, \dots, x_n) = \beta$ and $z_l = 2$. Note that $(z_1, \dots, z_l) = (c_1, \dots, c_l)$ in this solution, therefore $(c_1, \dots, c_l) \in \rho$. By the definition, $c_i \leqslant a_i$ for every $i < l$. We apply $s_{0,2}$ to the tuples $(a_1, \dots, a_{l-1}, b_l)$ and $(c_1, \dots, c_l)$ to obtain the tuple $(a_1, \dots, a_l)$, which is not from ρ. This contradicts the fact that $s_{0,2}$ preserves ρ.

Assume that $a_i \neq 2$ for every i. W.l.o.g. we assume that $a_1 = \dots = a_k = 1$ and $a_{k+1} = \dots = a_l = 0$. If $k = 0$ and $(a_1, \dots, a_l) = (0, \dots, 0)$, then we consider a solution of Φ corresponding to $(x_1, \dots, x_n) = (0, \dots, 0)$. By the definition of F_{z_i} we have $(z_1, \dots, z_l) = (0, \dots, 0)$ in this solution. Hence, $(0, \dots, 0) \in \rho$, which contradicts our assumption. Assume that $k \geqslant 1$. For each $i \in [k]$ we define a tuple α_i as follows. Since F_{z_i} is defined on any tuple $\beta < \alpha$ and $F_{z_i}(\beta) = 1$ for some $\beta < \alpha$, there exists a minimal 1-set $\alpha_i \leqslant \beta$ for z_i. Assume that $\alpha' := \alpha_1 \vee \dots \vee \alpha_k < \alpha$. Consider a solution of Φ with $(x_1, \dots, x_n) = \alpha'$. Since $F_{z_i}(\alpha')$ is defined, $F_{z_i}(\alpha_i) = 1$ and

F_{z_i} is monotonic, we have $F_{z_i}(\alpha') = 1$ for every $i \in [k]$. Therefore, $(z_1, \ldots, z_l) = (a_1, \ldots, a_l)$ in this solution, which means that $(a_1, \ldots, a_l) \in \rho$ and contradicts the assumption.

Thus, $\alpha' \not\leqslant \alpha$. Since $\alpha_i \leqslant \alpha$ for every i, we obtain $\alpha' \leqslant \alpha$, and therefore $\alpha' = \alpha$, which completes the proof. □

Lemma 32. *Suppose α is a minimal 1-set for y, $i \in \{1, 2, \ldots, n\}$, and $2^{i-1}02^{n-i} \in R$. Then $\alpha(i) = 1$ if and only if $F_y(2^{i-1}02^{n-i}) = 0$.*

Proof. Assume that $\alpha(i) = 0$ and $F_y(2^{i-1}02^{n-i}) = 0$. We have $s_2(2^{i-1}02^{n-i}0, \alpha 1) = 2^{i-1}02^{n-i}2$, which means that Φ has a solution with $(x_1, \ldots, x_n) = 2^{i-1}02^{n-i}$ and $y = 2$. This contradicts the fact that $F_y(2^{i-1}02^{n-i}) = 0$.

Assume that $\alpha(i) = 1$ and $F_y(2^{i-1}02^{n-i})$ is not defined or equal to 1. Then Φ has a solution with $(x_1, \ldots, x_n) = 2^{i-1}02^{n-i}$ and $y = c$, where $c \neq 0$. Let $\beta < \alpha$ be the tuple that differs from α only in the i-th coordinate. Since $f_{0,2}(\alpha, \beta, 2^{i-1}02^{n-i}) = \alpha$ and $f_{0,2}(1, 0, c) = 2$, Φ should have a solution with $(x_1, \ldots, x_n) = \alpha$ and $y = 2$, which contradicts the definition of a minimal 1-set. □

The following lemma and corollary do not play a role in our main result, but we present them for further curiosity and another sanity check.

Lemma 33. *Suppose $\Gamma \subseteq \mathrm{Inv}(f_{0,2})$, all relations in Γ are of arity at most k. Then any pp-definition of σ_n over Γ, where $n \geqslant 2$, has at least $2^n/2^k$ constraints.*

Proof. Suppose $\sigma_n = \Phi(x_1, \ldots, x_n)$, where Φ is a conjunctive formula over Γ. There exist 2^n tuples from $A^{2n} \setminus \sigma_n$ and each of them has at least two 1s. By Lemma 31, for each $\alpha \in A^{2n} \setminus \sigma_n$ there should be a constraint $\rho(z_1, \ldots, z_l)$ such that $\alpha = \bigvee_{i \in B} \alpha_i$ for some $B \subseteq \{1, 2, \ldots, l\}$. Since every constraint of Φ is of arity at most k, there are at most 2^k options to choose B. Therefore, one constraint of Φ can cover at most 2^k tuples from $A^{2n} \setminus \sigma_n$. Thus, Φ has at least $2^n/2^k$ constraints. □

Thus, for a fixed (finite) Γ we need exponentially many constraints to define σ_n. Since σ_n can be obtained from τ_n by identification of variables, we have the following corollary.

Corollary 34. *Suppose $\Gamma \subseteq \mathrm{Inv}(f_{0,2})$, all relations in Γ are of arity at most k. Then any pp-definition of τ_n over Γ, where $n \geqslant 2$, has at least $2^n/2^k$ constraints.*

Below we present an algorithm that solves $\mathrm{QCSP}^2(\Gamma)$ in polynomial time for $\Gamma \subseteq \mathrm{Inv}(f_{0,2})$ (see the pseudocode of the function $\textsc{Solve}_2$). Again, by $\textsc{SolveCSP}$ we denote a polynomial algorithm, solving constraint satisfaction problem for a constraint language preserved by a semilattice operation: it returns true if it has a solution, it returns false otherwise.

Lemma 35. *Function $\textsc{Solve}_2$ solves $\mathrm{QCSP}^2(\Gamma)$ in polynomial time for a finite constraint language $\Gamma \subseteq \mathrm{Inv}(f_{0,2})$.*

Proof. First, let us show that the algorithm actually solves the problem. If the answer is false, then we found an evaluation of $(x_1, \ldots, x_n)$ such that the corresponding CSP has no solutions, which means that the answer is correct.

Assume that the answer is true. Let $R(x_1, \ldots, x_n)$ be defined by the formula $\exists y_1 \ldots \exists y_s \Phi$. We need to prove that R is a full relation.

1: **function** $\textsc{Solve}_2(\Theta)$
2: **Input:** $\mathrm{QCSP}^2(\Gamma)$ instance $\Theta = \forall x_1 \ldots \forall x_n \exists y_1 \ldots \exists y_s \Phi$.
3: **if** $\neg\textsc{SolveCSP}(\mathbf{x} = (0, \ldots, 0) \wedge \Phi)$ **then return** false
▷ Here $\mathbf{x} = (x_1, \ldots, x_n)$
4: **for** $i := 1, \ldots, n$ **do** ▷ Check all tuples with just one 1
5: $\mathbf{c} := (\underbrace{0, \ldots, 0}_{i-1}, 1, 0, \ldots, 0)$
6: **if** $\neg\textsc{SolveCSP}(\mathbf{x} = \mathbf{c} \wedge \Phi)$ **then return** false
7: **for** $j := 1, \ldots, s$ **do**
▷ Calculate the minimal 1-set for every y_j
8: $\alpha_j := (0, \ldots, 0)$
9: **for** $i := 1, \ldots, n$ **do**
10: $D_i := \varnothing$
11: $\mathbf{c} := (\underbrace{2, \ldots, 2}_{i-1}, 0, 2, \ldots, 2)$
12: **for** $a \in A$ **do**
13: **if** $\textsc{SolveCSP}(\mathbf{x} = \mathbf{c} \wedge y_j = a \wedge \Phi)$ **then**
14: $D_i := D_i \cup \{a\}$
15: **if** $D_i = \varnothing$ **then return** false
16: **if** $D_i = \{0\}$ **then**
17: $\alpha_j := \alpha_j \vee (\underbrace{0, \ldots, 0}_{i-1}, 1, 0, \ldots, 0)$
18: **for** a constraint $\rho(z_1, \ldots, z_l)$ of Φ **do**
▷ Check all constraints
19: **for** $V \subseteq \{1, 2, \ldots, l\}$ **do**
▷ Check all subsets of variables
20: $\beta := (0, \ldots, 0)$
21: **for** $j \in V$ **do**
22: **if** $z_j = x_i$ for some i **then**
23: $\beta := \beta \vee (\underbrace{0, \ldots, 0}_{i-1}, 1, 0, \ldots, 0)$
▷ Add the minimal 1-set for x_i
24: **if** $z_j = y_i$ for some i **then**
25: $\beta := \beta \vee \alpha_i$
▷ Add the minimal 1-set for y_i
26: **if** $\neg\textsc{SolveCSP}(\mathbf{x} = \beta \wedge \Phi)$ **then return** false
27: **return** true

Assume the converse. Using the semilattice operation s_2 we can generate A^n from $\{0, 1\}^n$, hence $\{0, 1\}^n \not\subseteq R$. Then let α be a minimal tuple from $\{0, 1\}^n \setminus R$. Since we checked that $(0, 0, \ldots, 0)$ and all tuples having just one 1 are from R, α contains at least two 1. Then, by Lemma 31, there should be a constraint $\rho(z_1, \ldots, z_l)$ of Φ and a subset $V \subseteq \{1, 2, \ldots, l\}$ such that α is a disjunction of the minimal 1-sets of z_i for $i \in V$. Thus, it is sufficient to find the minimal 1-set corresponding to each variable and check all the disjunctions.

By Lemma 32, if α_j is a minimal 1-set for a variable y_j then it was correctly found in lines 7-17 of the algorithm. Note that if y_j does not have a minimal 1-set then we do not care what we found. Then, in lines 18-25 we check all constraints of Φ, check all subsets of variables V, and calculate the corresponding disjunction. In line 26 we check whether Φ has a solution with $(x_1, \ldots, x_n) = \alpha$. Thus, Lemma 31 guarantees that $\{0, 1\}^n \subseteq R$, and therefore $A^n \subseteq R$.

It remains to show that the algorithm works in polynomial time. In the algorithm we just solve at most $1 + n + s \cdot n \cdot 3 + m \cdot 2^r$ CSP instances over a language preserved by the semilattice operation s_2, where m is the number of constraints in Φ and r is the maximal arity of constraints in Φ. Since Γ is finite, r is a constant, hence the algorithm is polynomial. □

Corollary 36. QCSP(Γ) *is in P for every finite* $\Gamma \subseteq \mathrm{Inv}(f_{0,2})$.

Proof. Since $s_{0,2}$ is a 0-stable operation preserving Γ, Lemma 18 implies that QCSP(Γ) can be polynomially reduced to QCSP^2(Γ), and QCSP^2(Γ) can be solved by the function Solve$_2$. □

10 CONCLUSION

Our demonstration of QCSP monsters suggests that a complete complexity classification of QCSP(Γ) under polynomial reductions is likely to be exceedingly challenging. Indeed, suppose P $\neq$ NP, how many equivalence classes of problems QCSP(Γ) are there up to polynomial equivalence? In this paper we showed that there are at least six of them. Are there any more? Are there infinitely many? We don't know the answer.

Meanwhile, the most sensible approach to complexity classification for QCSP(Γ) might be to try to find those that are in P, in contradistinction to those that are NP-hard under polynomial Turing reductions (which would thus capture also the co-NP-hardness). Similarly, someone could ask about a general criteria for the QCSP to be PSpace-hard or to be a member of a concrete complexity class, which is also a very intriguing question.

As the next step, it seems very natural to work on a classification for constraint languages on a three-element domain without constants, where the reduction to CSP doesn't work and brand new ideas are required.

ACKNOWLEDGMENTS

The first proof that there is a concrete finite constraint language Γ such that QCSP(Γ) is co-NP-complete is due to Miroslav Olšák. The observation that QCSP(Γ) as in Corollary 10 is Θ_2^{P}-complete was communicated to us by Emil Jeřábek. We are grateful for discussions with Victor Lagerkvist about polynomial-sized pp-definitions, with particular reference to our relations τ_i. We would like also to thank Antoine Mottet and Libor Barto for fruitful discussions, as well as discerning comments from several anonymous reviews.

The first author has received funding from the European Research Council (ERC) under the European Unions Horizon 2020 research and innovation programme (grant agreement No 771005) and from Russian Foundation for Basic Research (grant 19-01-00200).

REFERENCES

[1] Libor Barto and Michael Pinsker. 2016. The algebraic dichotomy conjecture for infinite domain Constraint Satisfaction Problems. In *Proceedings of the 31st Annual ACM/IEEE Symposium on Logic in Computer Science, LICS '16, New York, NY, USA, July 5-8, 2016.* 615–622. https://doi.org/10.1145/2933575.2934544

[2] Manuel Bodirsky and Hubie Chen. 2009. Relatively quantified constraint satisfaction. *Constraints* 14, 1 (2009), 3–15. https://doi.org/10.1007/s10601-008-9054-z

[3] V. G. Bodnarchuk, L. A. Kaluzhnin, V. N. Kotov, and B. A. Romov. 1969. Galois theory for Post algebras parts I and II. *Cybernetics* 5 (1969), 243–252, 531–539.

[4] Ferdinand Börner, Andrei A. Bulatov, Hubie Chen, Peter Jeavons, and Andrei A. Krokhin. 2009. The complexity of constraint satisfaction games and QCSP. *Inf. Comput.* 207, 9 (2009), 923–944.

[5] Joshua Brakensiek and Venkatesan Guruswami. 2018. Promise Constraint Satisfaction: Structure Theory and a Symmetric Boolean Dichotomy. In *Proceedings of the Twenty-Ninth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2018, New Orleans, LA, USA, January 7-10, 2018.* 1782–1801. https://doi.org/10.1137/1.9781611975031.117

[6] Andrei A. Bulatov. 2017. A dichotomy theorem for nonuniform CSPs. In *2017 IEEE 58th Annual Symposium on Foundations of Computer Science (FOCS).* 319–330.

[7] Samuel R. Buss and Louise Hay. 1991. On Truth-Table Reducibility to SAT. *Inf. Comput.* 91, 1 (1991), 86–102. https://doi.org/10.1016/0890-5401(91)90075-D

[8] Catarina Carvalho, Florent R. Madelaine, and Barnaby Martin. 2015. From complexity to algebra and back: digraph classes, collapsibility and the PGP. In *30th Annual IEEE Symposium on Logic in Computer Science (LICS).*

[9] Catarina Carvalho, Barnaby Martin, and Dmitriy Zhuk. 2017. The Complexity of Quantified Constraints Using the Algebraic Formulation. In *42nd International Symposium on Mathematical Foundations of Computer Science, MFCS 2017, August 21-25, 2017 - Aalborg, Denmark.* 27:1–27:14. https://doi.org/10.4230/LIPIcs.MFCS.2017.27

[10] Hubie Chen. 2004. Quantified Constraint Satisfaction and 2-Semilattice Polymorphisms. In *Principles and Practice of Constraint Programming - CP 2004, 10th International Conference, CP 2004, Toronto, Canada, September 27 - October 1, 2004, Proceedings.* 168–181. https://doi.org/10.1007/978-3-540-30201-8_15

[11] Hubie Chen. 2006. A rendezvous of logic, complexity, and algebra. *SIGACT News* 37, 4 (2006), 85–114. https://doi.org/10.1145/1189056.1189076

[12] Hubie Chen. 2008. The Complexity of Quantified Constraint Satisfaction: Collapsibility, Sink Algebras, and the Three-Element Case. *SIAM J. Comput.* 37, 5 (2008), 1674–1701. https://doi.org/10.1137/060668572

[13] Hubie Chen. 2011. Quantified constraint satisfaction and the polynomially generated powers property. *Algebra universalis* 65, 3 (2011), 213–241. https://doi.org/10.1007/s00012-011-0125-4 An extended abstract appeared in ICALP B 2008.

[14] Hubie Chen. 2012. Meditations on Quantified Constraint Satisfaction. In *Logic and Program Semantics - Essays Dedicated to Dexter Kozen on the Occasion of His 60th Birthday.* 35–49.

[15] Hubie Chen, Florent R. Madelaine, and Barnaby Martin. 2015. Quantified Constraints and Containment Problems. *Logical Methods in Computer Science* 11, 3 (2015). https://doi.org/10.2168/LMCS-11(3:9)2015

[16] Hubie Chen and Peter Mayr. 2016. Quantified Constraint Satisfaction on Monoids. In *25th EACSL Annual Conference on Computer Science Logic, CSL 2016, August 29 - September 1, 2016, Marseille, France.* 15:1–15:14. https://doi.org/10.4230/LIPIcs.CSL.2016.15

[17] David Geiger. 1968. Closed systems of functions and predicates. *Pacific journal of mathematics* 27, 1 (1968), 95–100.

[18] Vladimir Kolmogorov, Andrei A. Krokhin, and Michal Rolínek. 2017. The Complexity of General-Valued CSPs. *SIAM J. Comput.* 46, 3 (2017), 1087–1110. https://doi.org/10.1137/16M1091836

[19] Victor Lagerkvist and Magnus Wahlström. 2017. The power of primitive positive definitions with polynomially many variables. *J. Log. Comput.* 27, 5 (2017), 1465–1488. https://doi.org/10.1093/logcom/exw005

[20] Thomas Lukasiewicz and Enrico Malizia. 2017. A novel characterization of the complexity class $\Theta_{\mathrm{k}}^{\mathrm{P}}$ based on counting and comparison. *Theor. Comput. Sci.* 694 (2017), 21–33. https://doi.org/10.1016/j.tcs.2017.06.023

[21] Florent R. Madelaine and Barnaby Martin. 2018. On the Complexity of the Model Checking Problem. *SIAM J. Comput.* 47, 3 (2018), 769–797. https://doi.org/10.1137/140965715

[22] Barnaby Martin. 2017. Quantified Constraints in Twenty Seventeen. In *The Constraint Satisfaction Problem: Complexity and Approximability*, Andrei Krokhin and Stanislav Zivny (Eds.). Dagstuhl Follow-Ups, Vol. 7. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, Dagstuhl, Germany, 327–346. https://doi.org/10.4230/DFU.Vol7.15301.327

[23] Christos H. Papadimitriou. 1994. *Computational Complexity.* Addison-Wesley.

[24] D. Zhuk. 2017. A Proof of CSP Dichotomy Conjecture. In *2017 IEEE 58th Annual Symposium on Foundations of Computer Science (FOCS).* 331–342. https://doi.org/10.1109/FOCS.2017.38

[25] Dmitriy Zhuk. 2019. The size of generating sets of powers. *Journal of Combinatorial Theory, Series A* 167 (2019), 91–103.

[26] Dmitriy Zhuk and Barnaby Martin. 2019. QCSP monsters and the demise of the Chen Conjecture. *CoRR* abs/1907.00239 (2019). arXiv:1907.00239 http://arxiv.org/abs/1907.00239