

QCSP monsters and the demise of the Chen Conjecture

Dmitriy Zhuk*

Barnaby Martin

July 28, 2022

Abstract

We give a surprising classification for the computational complexity of the Quantified Constraint Satisfaction Problem over a constraint language Γ , $\text{QCSP}(\Gamma)$, 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 $\text{QCSP}(\Gamma)$ 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 $\text{QCSP}(\Gamma)$ is in NP. Otherwise, the polymorphism clone of Γ has the exponentially generated powers (EGP) property and $\text{QCSP}(\Gamma)$ is PSpace-complete.

1 Introduction

The *Quantified Constraint Satisfaction Problem* $\text{QCSP}(\Gamma)$ is the generalization of the *Constraint Satisfaction Problem* $\text{CSP}(\Gamma)$ which, given the latter in its logical form, augments its native existential quantification with universal quantification. That is, $\text{QCSP}(\Gamma)$ is the problem to evaluate a sentence of the form $\forall x_1 \exists y_1 \dots \forall x_n \exists y_n \Phi$, 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 $\text{CSP}(\Gamma)$, for all finite Γ , between P and NP-complete [8, 29], a desire has been building for a classification for $\text{QCSP}(\Gamma)$. Indeed, since the classification of the *Valued CSPs* was reduced to that for CSPs [20], 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* [6] and finitely-bounded, homogeneous infinite-domain CSPs [2]. The exact dates of these classification questions are not exactly cast in stone but the question for QCSPs dates at least to the announcement of the classification for the 2-element case in [28].

While $\text{CSP}(\Gamma)$ remains in NP for any finite Γ , $\text{QCSP}(\Gamma)$ can be PSpace-complete, as witnessed by *Quantified 3-Satisfiability* or *Quantified Graph 3-Colouring* (see [5]). 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 $\text{CSP}(\Gamma)$ and $\text{QCSP}(\Gamma + 1)$ are polynomially equivalent. Thus, and similarly to the Valued

*The 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).

CSPs, the CSP classification will play a part in the QCSP classification. It is now clear that $\text{QCSP}(\Gamma)$ can achieve each of the complexities P, NP-complete and PSpace-complete. It has thus far been believed these were the only possibilities (see [5, 14, 13, 16, 25] 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 $(x_1^1, \dots, x_k^1), \dots, (x_1^m, \dots, x_k^m)$ in R , then also $(f(x_1^1, \dots, x_k^1), \dots, f(x_1^m, \dots, x_k^m))$ 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* $\text{Pol}(\Gamma)$ 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 $\text{Inv}(F)$ we denote the set of all invariants of F . We call an operation f *idempotent* if $f(x, \dots, x) = x$, for all x . An idempotent operation f is a *weak near-unanimity* (WNU) operation if $f(y, x, x, \dots, x) = f(x, y, x, \dots, x) = \dots = f(x, x, \dots, x, y)$. We recall the following form of the (now proved) Feder-Vardi Conjecture.

Theorem 1 (CSP Dichotomy [8, 29]). *Let Γ be a finite constraint language with all constants. If Γ admits some WNU polymorphism, then $\text{CSP}(\Gamma)$ is in P. Otherwise, $\text{CSP}(\Gamma)$ is NP-complete.*

Indeed, this theorem holds with the same criterion even without constants [1]. However, for the general 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 $\text{Pol}(\Gamma)$ 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 [17]). However, all prior conjectures for the QCSP have been made in this safer environment where we may assume idempotency and almost all classifications apply only to this situation. A rare exception to this is the paper [18] 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 $\text{QCSP}(\Gamma)$, denoted $\text{QCSP}^2(\Gamma)$, in which the input is of the form $\forall x_1 \dots \forall x_n \exists y_1 \dots \exists y_m \Phi$. In order to solve this restriction of the problem it suffices to look at (the conjunction of) $|D|^n$ instances of $\text{CSP}(\Gamma)$. It is not hard to show (see [15]) that, if D^n can be generated under $\text{Pol}(\Gamma)$ from some subset $\Sigma \subseteq D^n$, then one need only consult (the conjunction over) of $|\Sigma|$ instances of $\text{CSP}(\Gamma)$. 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 $\text{Pol}(\Gamma)$ from Σ , then we say $\text{Pol}(\Gamma)$ 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 $\text{CSP}(\Gamma)$. It turns out that if the nature of the PGP property is sufficiently benign a similar reduction can be made for the full $\text{QCSP}(\Gamma)$ to the CSP with constants [15, 10]. Another behaviour that might arise with $\text{Pol}(\Gamma)$ is that there is an exponential function f so that the smallest generating sets under $\text{Pol}(\Gamma)$ for $\Sigma \subseteq D^n$ require size at least $f(n)$. We describe this as the *exponentially generated powers* (EGP) property. The outstanding conjecture in the area of QCSPs is the merger of Conjectures 6 and 7 in [16] which we have dubbed in [11] the *Chen Conjecture*.

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

In [16], 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 [15]). Since [31], 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 we know that the PGP cases are in NP [15, 10].

Theorem 2 ([11]). *Let Γ be a finite constraint language with all constants such that $\text{Pol}(\Gamma)$ has PGP. Then $\text{QCSP}(\Gamma)$ reduces to a polynomial number of instances of $\text{CSP}(\Gamma)$ 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 $\text{Pol}(\Gamma)$ that are EGP is given in [31], where it is shown that Γ must allow the *primitive positive* (pp) definition (of the form $\exists x_1 \dots \exists x_n \Phi$) 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 $\text{QCSP}(\Gamma)$ is co-NP-hard [11]. 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 $\text{QCSP}(\Gamma)$ where Γ is a finite 3-element constraint language with constants. Not only do we find Γ so that $\text{QCSP}(\Gamma)$ is co-NP-complete, but also we find Γ so that $\text{Pol}(\Gamma)$ has EGP yet $\text{QCSP}(\Gamma)$ 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 $\text{QCSP}(\Gamma)$ is DP-complete (from 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 $\text{QCSP}(\Gamma)$ 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 among the larger natural classes 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 [3]. 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 $\text{Pol}(\Gamma)$ has PGP, then $\text{QCSP}(\Gamma)$ is in NP. If Γ further admits a WNU polymorphism, then $\text{QCSP}(\Gamma)$ is in P, else it is NP-complete. Otherwise, $\text{Pol}(\Gamma)$ has EGP and $\text{QCSP}(\Gamma)$ is PSpace-complete.*

It is hard to exaggerate how surprising our discovery of multitudinous complexities above P for the QCSP is. In Table 1 (reproduced but updated from [23]), all syntactic fragments of first-order logic built from subsets of $\{\forall, \exists, \wedge, \vee, \neg, =\}$ are considered. It can be seen that they

Fragment	Dual	Classification
$\{\exists, \vee\}$ $\{\exists, \vee, =\}$ $\{\exists, \vee, \neq\}$	$\{\forall, \wedge\}$ $\{\forall, \wedge, \neq\}$ $\{\forall, \wedge, =\}$	LogSpace
$\{\exists, \wedge, \vee\}$ $\{\exists, \wedge, \vee, =\}$ $\{\exists, \wedge, \vee, \neq\}$	$\{\forall, \wedge, \vee\}$ $\{\forall, \wedge, \vee, \neq\}$ $\{\forall, \wedge, \vee, =\}$	LogSpace if there is some element a so that all relations contain the tuple $(a, \dots, a)$, and NP-complete otherwise
$\{\exists, \wedge\}$ $\{\exists, \wedge, =\}$ $\{\exists, \wedge, \neq\}$	$\{\forall, \vee\}$ $\{\forall, \vee, \neq\}$ $\{\forall, \vee, =\}$	CSP dichotomy: P if language admits some WNU polymorphism, and NP-complete otherwise.
$\{\exists, \forall, \wedge\}$ $\{\exists, \forall, \wedge, =\}$ $\{\exists, \forall, \wedge, \neq\}$	$\{\exists, \forall, \vee\}$ $\{\exists, \forall, \vee, \neq\}$ $\{\exists, \forall, \vee, =\}$	QCSP polychotomy?
$\{\forall, \exists, \wedge, \vee\}$		Tetrachotomy: P, NP-complete, co-NP-complete or PSpace-complete
$\{\forall, \exists, \wedge, \vee, =\}$ $\{\neg, \exists, \forall, \wedge, \vee, =\}$	$\{\forall, \exists, \wedge, \vee, \neq\}$ $\{\neg, \exists, \forall, \wedge, \vee, =\}$	LogSpace when domain at most 1, PSpace-complete otherwise
$\{\neg, \exists, \forall, \wedge, \vee\}$		LogSpace when language contains only empty or full relations, PSpace-complete otherwise

Table 1: Complexity classifications for model-checking syntactic fragments of first-order logic. The “Schaefer classes” are the domain 2 CSPs that have a polynomial algorithm, as given in the seminal work [28].

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 [11], 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 [11]). *Let $\mathbb{A}$ be an idempotent algebra on a finite domain A where we encode relations in $\text{Inv}(\mathbb{A})$ in quantifier-free logic with constants and equality. If $\mathbb{A}$ satisfies PGP, then $\text{QCSP}(\text{Inv}(\mathbb{A}))$ is in NP. Otherwise, $\text{QCSP}(\text{Inv}(\mathbb{A}))$ is co-NP-hard.*

In this theorem it was known that co-NP-hardness could not be improved to PSpace-completeness, because $\text{QCSP}(\text{Inv}(\mathbb{A}))$ is co-NP-complete when, e.g., $\mathbb{A} = \text{Pol}(\{0, 1, 2\}; 0, 1, 2, \tau_1, \tau_2, \dots)$ where $\alpha = \{0, 2\}$ and $\beta = \{1, 2\}$. However, $\text{Inv}(\mathbb{A})$ is not finitely related, that is, generated from a finite set of relations under pp-closure. It was not thought possible that

there could be finite Γ such that $\text{QCSP}(\Gamma)$ 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 [11].

The systematic complexity-theoretic study of QCSPs dates to the early versions of [5] (the earliest is a technical report from 2002). By the time of the journal version [5], the significance of the semilattice-without-unit $s = s_c$ (definition at opening of Section 6) had become apparent in a series of papers of Chen [12, 14, 15]. Although $\text{CSP}(\text{Inv}(\{s\}))$ is in P it is proved in [5] that $\text{QCSP}(\text{Inv}(\{s\}))$ is PSpace-complete (even for some finite sublanguage of $\text{Inv}(\{s\})$). 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 [21]. 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 (see Figure 1). In Section 2 we formulate the main results of the paper. We start with the classification of the complexity of $\text{QCSP}(\Gamma)$ 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 $\text{DP} = \text{NP} \wedge \text{co-NP}$.

In Sections 3-5 we give necessary definitions, prove Chen's conjecture for the conservative case, and prove how to combine two constraint languages in one constraint language to generate new complexity classes. In Section 6 we prove hardness results we need for the 3-element case and show how $\text{QCSP}(\Gamma)$ can be reduced to $\text{QCSP}^2(\Gamma)$ (the Π_2 restriction of $\text{QCSP}(\Gamma)$). In Sections 7 and 8 we define two families of constraint languages Γ such that $\text{Pol}(\Gamma)$ has the EGP property but $\text{QCSP}(\Gamma)$ can be solved in polynomial time. Additionally, we show that τ_n can be pp-defined from Γ only by a formula of exponential size, which explains why the proof of co-NP-hardness does not work.

In Sections 9-10 we prove the classification of the complexity of $\text{QCSP}(\Gamma)$ for constraint languages on a 3-element domain containing all constants. In Section 9 we derive the existence of necessary relations in Γ from the fact that $\text{Pol}(\Gamma)$ has the EGP property. In Section 10 we complete the proof of the 3-element classification.

2 Main Results

In this section we formulate two main results of the paper: classification of the complexity of $\text{QCSP}(\Gamma)$ 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 $\{0, 1, 2\}$.

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

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

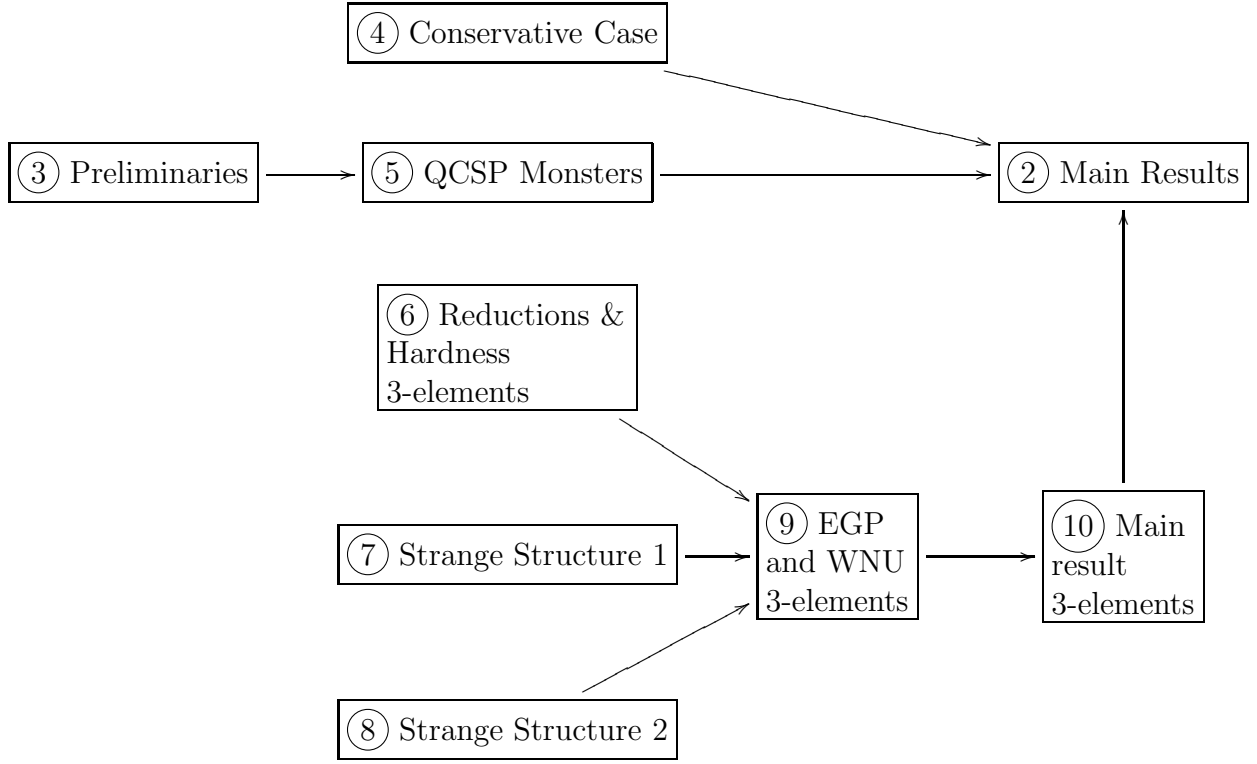


Figure 1: Section dependency diagram for this article.

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

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

If $\{a, b, c\} = \{0, 1, 2\}$ then call any idempotent operation f , such that $f(x, a, b) = x$ and $f(x, c, c) = c$, *ab-stable*. Note that $f(x, y, z) = s_{a,c}(x, y)$ is an *ab-stable* operation. We get the following characterization of the complexity of $\text{QCSP}(\Gamma)$ on a 3-element domain.

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

1. *in P, if $\text{Pol}(\Gamma)$ has the PGP property and has a WNU operation.*
2. *NP-complete, if $\text{Pol}(\Gamma)$ has the PGP property and has no WNU operation.*
3. *PSpace-complete, if $\text{Pol}(\Gamma)$ has the EGP property and has no WNU operation.*
4. *PSpace-complete, if $\text{Pol}(\Gamma)$ has the EGP property and $\text{Pol}(\Gamma)$ does not contain an ab-stable operation.*
5. *in P, if $\text{Pol}(\Gamma)$ contains $s_{a,c}$ and $g_{a,c}$ for some $a, c \in \{0, 1, 2\}$, $a \neq c$.*
6. *in P, if $\text{Pol}(\Gamma)$ contains $f_{a,c}$ for some $a, c \in \{0, 1, 2\}$, $a \neq c$.*
7. *co-NP-complete otherwise (in which case $\text{Pol}(\Gamma)$ additionally has an ab-stable operation).*

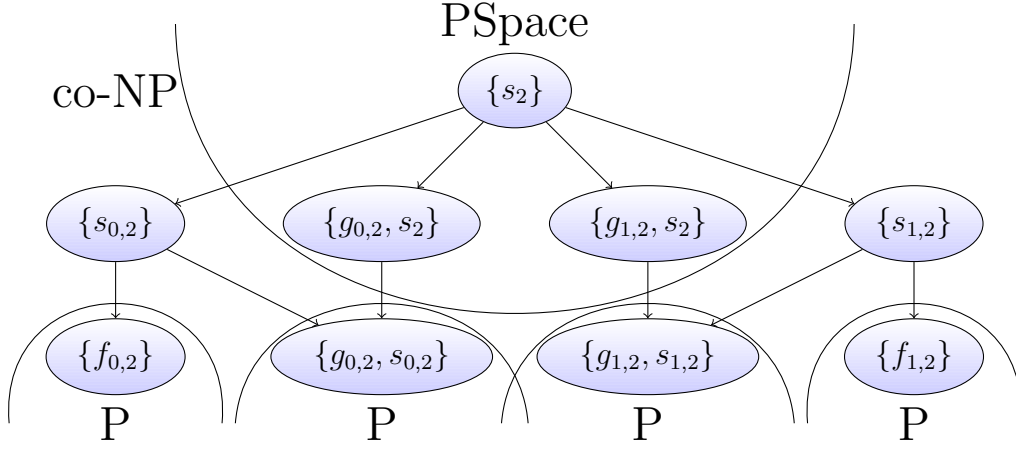


Figure 2: Constraint languages defined as invariants of sets of operations and their complexity.

Note that the semilattice s_c can be derived from each of the operations $f_{a,c}$, $s_{a,c}$. As we know from [5], the problem $\text{QCSP}(\text{Inv}(s_2))$ is PSpace-complete. Figure 2 demonstrates how adding new operations makes the constraint language weaker and the quantified constraint satisfaction problem easier. Note that all the languages in Figure 2 have the EGP property.

2.2 Examples of Constraint Languages

In this section we will show an example of a constraint language on $\{0, 1, 2\}$ for every case of Theorem 6 and explain informally why it gives the respective complexity class.

Case (1). $\text{Pol}(\Gamma)$ has the PGP property and has a WNU operation, and $\text{QCSP}(\Gamma)$ is in P. We can build a constraint language Γ with a single ternary relation $x - y + z = 1 \pmod 3$. That $\text{Pol}(\Gamma)$ has the PGP property and has a WNU operation as witnessed by the operation $x - y + z = 1 \pmod 3$. We can solve $\text{QCSP}(\Gamma)$ by reducing it to a polynomial number of instances of $\text{CSP}(\Gamma)$, as detailed in Theorem 2, then solving these by Gaussian elimination.

Case (2). $\text{Pol}(\Gamma)$ has the PGP property but has no WNU operation, and $\text{QCSP}(\Gamma)$ is NP-complete. We can take a single ternary relation $\{(1, 0, 0), (0, 1, 0), (0, 0, 1)\}$ that does not involve 2. As a result universal quantifiers do not play any role and can be omitted, therefore the problem is equivalent to the CSP over the relation $\{(1, 0, 0), (0, 1, 0), (0, 0, 1)\}$ on the domain $\{0, 1\}$ which is known to be NP-complete [28].

Case (3). $\text{Pol}(\Gamma)$ has the EGP property and has no WNU operation, and $\text{QCSP}(\Gamma)$ is PSpace-complete. We can take the closely related single ternary relation

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

The QCSP over this language is equivalent to the QCSP over the relation $\{(1, 0, 0), (0, 1, 0), (0, 0, 1)\}$ on the domain $\{0, 1\}$ which is known to be PSpace-complete [5].

Case (4). $\text{Pol}(\Gamma)$ has the EGP property and $\text{Pol}(\Gamma)$ does not contain an ab -stable operation, and $\text{QCSP}(\Gamma)$ is PSpace-complete. Unlike case (3), Γ may have a WNU polymorphism. Let τ be the 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 $\text{NAE}_3 = \{0, 1\}^3 \setminus \{(0, 0, 0), (1, 1, 1)\}$. Put

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

We claim that $\Gamma = \{\sigma, x = 0, x = 1, x = 2\}$ satisfies the case (4) and $\text{QCSP}(\Gamma)$ is PSpace-complete (see Section 6 for the detailed explanation). The reduction will be from the complement of (monotone) *Quantified Not-All-Equal 3-Satisfiability* (co-QNAE3SAT) which is

co-PSpace-hard (see [26]) and consequently also PSpace-hard (as PSpace is closed under complement). First, we define a predicate $p(x_1, \dots, x_n, y, y')$ such that for $y \neq y'$, $y, y' \in \{0, 2\}$, and all $x_1, \dots, x_n \in \{0, 1\}$ it equals the quantifier-free part of the instance of co-QNAE3SAT, and p always holds if $x_i = 2$ for some i or $y = y' \in \{0, 2\}$. Put $p(x_1, \dots, x_n, y_0, y_k) = \exists y_1 \dots \exists y_{k-1} \bigwedge_{i=1}^k \sigma(z_i^1, z_i^2, z_i^3, y_{i-1}, y_i)$, which expresses the quantifier-free part of the instance of co-QNAE3SAT

$$\Phi := \neg(\text{NAE}_3(z_1^1, z_1^2, z_1^3) \wedge \dots \wedge \text{NAE}_3(z_k^1, z_k^2, z_k^3)).$$

It remains to show how to add quantifiers to Φ :

- The formula $\forall x_1 p(x_1, \dots, x_n, y, y')$ obviously expresses $\forall x_1 \Phi$.
- It is more complicated with the existential quantifier because we need x_1 to be from $\{0, 1\}$ and we cannot just write $\exists x_1 p(x_1, \dots, x_n, y, y')$ to express $\exists x_1 \Phi$. Nevertheless, the following trick suggested by Miroslav Olšák solves the problem. To express $\exists x_1 \Phi$ we define

$$p'(x_2, \dots, x_n, y, y') = \exists z \forall x_1 \exists z' (p(x_1, \dots, x_n, z, z') \wedge \sigma(x_1, 0, 0, y, z') \wedge \sigma(x_1, 1, 1, y', z')).$$

Thus, instead of adding the existential quantifier to x_1 we add the universal quantifier to x_1 , two existential quantifiers, and two additional constraints. In fact, if z were chosen to be equal to y then to satisfy the formula for $x_1 = 0$ we send z' to y' , which, assuming $y \neq y'$, implies that Φ holds on $x_1 = 0$. Otherwise, if $z \neq y$ then to satisfy the formula for $x_1 = 1$ we send z' to y , which, assuming $y \neq y'$, implies that Φ holds on $x_1 = 1$. Thus, we simulated adding the existential quantifier to Φ .

Cases (5)-(6). Two concrete constraint languages are presented in Sections 7 and 8. These constraint languages have the EGP property but the QCSP over these languages can be solved in polynomial time. The polynomial algorithm is based on the following two ideas.

- The existence of the polymorphism $s_{a,c}$ guarantees that it is sufficient (see Lemma 18) to find a polynomial algorithm for sentences of the form $\forall x_1 \dots \forall x_n \exists y_1 \dots \exists y_m \Phi$.
- Since $\text{Pol}(\Gamma)$ has the EGP property, the usual reduction from QCSP to CSP (without looking at the formula) cannot help because theoretically we need to check exponentially many tuples $(x_1, \dots, x_n)$. The key idea here is to calculate the list of tuples $(x_1, \dots, x_n)$ we need to check to be sure that the formula holds for all the tuples. To calculate them we look at the formula Φ , substitute necessary values to some variables, and solve polynomially many CSP instances solvable in polynomial time (we have the semilattice polymorphism). See the algorithms in Sections 7 and 8 for more details.

Case (7). The first and probably easiest example of a constraint language Γ such that $\text{QCSP}(\Gamma)$ is co-NP-complete is as follows. Let $R_{AND} = \begin{pmatrix} 0 & 0 & 1 & 1 & 2 & \cdot \\ 0 & 1 & 0 & 1 & \cdot & 2 \\ 0 & 0 & 0 & 1 & \cdot & \cdot \end{pmatrix}$ and $R_{OR} = \begin{pmatrix} 0 & 0 & 1 & 1 & 2 & \cdot \\ 0 & 1 & 0 & 1 & \cdot & 2 \\ 0 & 1 & 1 & 1 & \cdot & \cdot \end{pmatrix}$ where columns are tuples of the relations and $\cdot$ means any element of $\{0, 1, 2\}$, that is R_{AND} and R_{OR} are row-wise the truth table of AND and OR respectively. Put $\Gamma = \{R_{AND}, R_{OR}\}$. The proof of co-NP-completeness can be divided into two steps (see Section 6 for the details):

- Using existential quantifiers we can compose these relations in the same way as we compose the operations *AND* and *OR* on $\{0, 1\}$. Using this idea we efficiently define (see Lemma 20) the complement to Not-All-Equal 3-Satisfiability. Hence, adding universal quantifiers makes the problem co-NP-hard.
- To show that the problem is in co-NP we need to explain a polynomial strategy for the Existential player in a (Hintikka) game corresponding to the truth of the instance of the QCSP. It turns out that to calculate the best move Existential can assume that all the later choices of the Universal player are 0. It is also important that the Existential player should play 2 if several moves are possible. Since the CSP over this language is solvable in polynomial time, after fixing the choice of the Universal player this can be done in polynomial time.

2.3 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, and Γ_1 contains a constant relation $(x = a)$. Then there exist constraint languages Δ_1 , Δ_2 , Δ_3 , Δ_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 $\text{QCSP}(\Delta_i)$ is polynomially equivalent to the following problem:*

- $i=1$ Given an instance of $\text{QCSP}(\Gamma_1)$ and instance of an NP-complete problem; decide whether both of them hold, i.e. $\text{QCSP}(\Gamma_1) \wedge \text{NP}$.*
- $i=2$ Given an instance of $\text{QCSP}(\Gamma_2)$ and an instance of $\text{QCSP}(\Gamma_3)$; decide whether both of them hold, i.e. $\text{QCSP}(\Gamma_2) \wedge \text{QCSP}(\Gamma_3)$.*
- $i=3$ Given $n > 0$, instances $I_1, \dots, I_n$ of $\text{QCSP}(\Gamma_2)$, and instances $J_1, \dots, J_n$ of $\text{CSP}(\Gamma_3)$; decide whether $(I_1 \vee J_1) \wedge \dots \wedge (I_n \vee J_n)$ holds, i.e. $(\text{QCSP}(\Gamma_2) \vee \text{CSP}(\Gamma_3)) \wedge \dots \wedge (\text{QCSP}(\Gamma_2) \vee \text{CSP}(\Gamma_3))$.*
- $i=4$ Given $n > 0$, instances $I_1, \dots, I_n$ of $\text{QCSP}(\Gamma_2)$, and instances $J_1, \dots, J_n$ of $\text{QCSP}(\Gamma_3)$; decide whether $(I_1 \vee J_1) \wedge \dots \wedge (I_n \vee J_n)$ holds, i.e. $(\text{QCSP}(\Gamma_2) \vee \text{QCSP}(\Gamma_3)) \wedge \dots \wedge (\text{QCSP}(\Gamma_2) \vee \text{QCSP}(\Gamma_3))$.*

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

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

Proof. By Theorem 6, there exists a constraint language Γ_1 on a 3-element domain with constants such that $\text{QCSP}(\Gamma_1)$ 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 $\text{QCSP}(\Gamma)$ is polynomially equivalent to DP. $\square$

The complexity class Θ_2^P (see [22] 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 [9], is as follows. In this theorem $i \leq p(|x|)$ indicates i is a positive integer smaller than $p(|x|)$, where x is a string of length $|x|$.

Theorem 9 ([9]). *Every predicate in Θ_2^P can be defined by a formula of the form $\exists i \leq p(|x|) A(i, x) \wedge \neg B(i, x)$ as well as by a formula of the form $\forall i \leq p'(|x|) A'(i, x) \vee \neg B'(i, x)$, where A, B, A' and B' are NP-predicates and p and 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 $\text{QCSP}(\Gamma)$ 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 $\text{QCSP}(\Gamma_1)$ is co-NP-complete. Choose a constraint language Γ_2 on a 2-element domain such that $\text{CSP}(\Gamma_2)$ is NP-complete. Using item 3 of Theorem 7, we construct a constraint language Γ so that $\text{QCSP}(\Gamma)$ is equivalent to the truth of $(I_1 \vee J_1) \wedge \dots \wedge (I_n \vee J_n)$, where $I_1, \dots, I_n$ are instances of $\text{QCSP}(\Gamma_1)$ and $J_1, \dots, J_n$ are instances of $\text{CSP}(\Gamma_2)$.

To prove membership of $\text{QCSP}(\Gamma)$ 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 $\text{CSP}(\Gamma_2)$ 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 $\text{QCSP}(\Gamma_1)$. 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 $\text{QCSP}(\Gamma)$ 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 $p'(|x|)$. $\square$

3 Preliminaries

Let $[n] = \{1, \dots, n\}$. 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 $x_1, \dots, x_n$ among its free variables and not containing $y_1, \dots, y_n$ in any capacity, then $\Phi_{y_1, \dots, y_n}^{x_1, \dots, x_n}$ is the result of substituting the free occurrences of $x_1, \dots, x_n$ by $y_1, \dots, y_n$, respectively. If I is an instance of $\text{QCSP}(\Gamma)$, then $\text{Var}(I)$ refers to the variables mentioned in I .

We always may assume that an instance of $\text{QCSP}(\Gamma)$ is of the prenex form

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

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 $f_1, \dots, f_n$ such that

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

is a solution of Φ for all $x_1, \dots, x_n$ (i.e. $y_i = f_i(x_1, \dots, x_i)$). 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 $\text{Clo}(G)$ 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 homomorphic image of a subalgebra 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 $f \in \text{Pol}(\Gamma)$ [4, 19].

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. We do not require the variables $x_1, \dots, x_n$ to be different; for instance, $\Phi(x_1, x_1)$ defines pairs of equal elements. 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 $\text{pr}_B(R)$ to be the $|B|$ -ary relation obtained from R by projecting to 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 $\alpha \cap \beta \neq \emptyset$ 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 [31].

4 The conservative case

In this section we prove Theorem 4 describing the complexity of $\text{QCSP}(\Gamma)$ for conservative constraint languages Γ , i.e. languages containing all unary relations. As it was mentioned in the introduction, if $\text{Pol}(\Gamma)$ has the PGP property then we can reduce $\text{QCSP}(\Gamma)$ 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 [11].

Lemma 11 ([11]). *Suppose Γ is a constraint language on domain D with constants, such that $\text{Pol}(\Gamma)$ 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 $n \geq 1$.*

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

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

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

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 . $\square$

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 $\text{Pol}(\Gamma)$ has PGP, then $\text{QCSP}(\Gamma)$ is in NP. If Γ further admits a WNU polymorphism, then $\text{QCSP}(\Gamma)$ is in P, else it is NP-complete. Otherwise, $\text{Pol}(\Gamma)$ has EGP and $\text{QCSP}(\Gamma)$ is PSpace-complete.*

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

Suppose now $\text{Pol}(\Gamma)$ 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., [26]). From an instance $\phi = \neg \forall x_1 \exists y_1 \dots \forall x_n \exists y_n \Phi$ of co-QNAE3SAT, where $\Phi = \text{NAE}_3(z_1^1, z_1^2, z_1^3) \wedge \dots \wedge \text{NAE}_3(z_k^1, z_k^2, z_k^3)$ 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 $\text{QCSP}(\Gamma)$ as follows. Consider ϕ to be $\exists x_1 \forall y_1 \dots \exists x_n \forall y_n \neg \Phi$ and set $\phi' =$

$$\exists x_1 \forall y_1 \dots \exists x_n \forall y_n x_1, \dots, x_n \in (\alpha \setminus \beta \cup \beta \setminus \alpha) \wedge \tau_k(z_1^1, z_1^2, z_1^3, \dots, z_k^1, z_k^2, z_k^3).$$

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

($\phi \in \text{co-QNAE3SAT}$ implies $\phi' \in \text{QCSP}(\Gamma)$.) 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 $\phi' \in \text{QCSP}(\Gamma)$.

($\phi' \in \text{QCSP}(\Gamma)$ implies $\phi \in \text{co-QNAE3SAT}$.) Suppose $\phi' \in \text{QCSP}(\Gamma)$. We will prove $\phi \in \text{co-QNAE3SAT}$ again using the form of ϕ being $\exists x_1 \forall y_1 \dots \exists x_n \forall y_n \neg \Phi$. 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 $\phi \in \text{co-QNAE3SAT}$. $\square$

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

Proof. Let $a \in A$ be as in the statement of the lemma and let a' be some element not in 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:

$R' = \{(a_1, \dots, a_h) \mid (\phi(a_1), \dots, \phi(a_h)) \in R\}$. Let $\text{NAE}_3 \subseteq \{a, a'\}^3$ be the ternary relation containing all tuples on $\{a, a'\}$ except for $(a, a, a), (a', a', a')$. Let $\Gamma' = \{R' \mid R \in \Gamma\} \cup \{\text{NAE}_3\}$.

Suppose I is an instance of $\text{QCSP}(\Gamma)$ and J is an instance of $\text{CSP}(\{\text{NAE}_3\})$, 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 $\text{QCSP}(\Gamma')$ that is equivalent to $I \wedge J$.

Suppose I' is an instance of $\text{QCSP}(\Gamma')$. W.l.o.g. we will assume that no variable appearing in an NAE_3 relation is universally quantified, else, since $|A| > 1$, this is a no-instance of $\text{QCSP}(\Gamma')$ and can be reduced to a fixed no-instance (e.g.) J of $\text{CSP}(\{\text{NAE}_3\})$. Now, we define an instance I of $\text{QCSP}(\Gamma)$ and an instance J of $\text{CSP}(\{\text{NAE}_3\})$ 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 $\text{QCSP}(\Gamma)$. 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 $I' \in \text{QCSP}(\Gamma')$ iff $I \in \text{QCSP}(\Gamma)$ and $J \in \text{CSP}(\{\text{NAE}_3\})$ it is enough to observe that $\text{QCSP}(\Gamma)$ and $\text{QCSP}(\Gamma' \setminus \{\text{NAE}_3\})$ are equivalent on all instances. $\square$

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 $\text{QCSP}(\Gamma)$ is polynomially equivalent to $(\text{QCSP}(\Gamma_1) \vee \text{QCSP}(\Gamma_2)) \wedge \dots \wedge (\text{QCSP}(\Gamma_1) \vee \text{QCSP}(\Gamma_2))$, i.e. the following decision problem: given n , instances $I_1, \dots, I_n$ of $\text{QCSP}(\Gamma_1)$, and instances $J_1, \dots, J_n$ of $\text{QCSP}(\Gamma_2)$; decide whether $(I_1 \vee J_1) \wedge \dots \wedge (I_n \vee J_n)$ holds.*

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

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

We would like to assume, w.l.o.g., that Γ contains constants a_1 and a_2 . This would follow from our definitions, so long as both Γ_1 and Γ_2 contain a unary empty relation. We may assume this, through definitions of the form $\forall x_1, \dots, x_k R(x_1, \dots, x_k)$, so long as not every relation in either Γ_1 or Γ_2 contains all tuples. Suppose, w.l.o.g., that every relation in Γ_1 contains all tuples, then we take Γ to be Γ_1 and we are done. Thus we may assume w.l.o.g., that Γ contains constants a_1 and a_2 .

Suppose we have an instance I_1 of $\text{QCSP}(\Gamma_1)$ and an instance I_2 of $\text{QCSP}(\Gamma_2)$. W.l.o.g. we will assume that neither I_1 nor I_2 is empty and that they are variable disjoint. We will explain how to build an instance J of $\text{QCSP}(\Gamma)$. Let $x_1, \dots, x_n$ be all universally quantified variables of I_1 .

We replace every atomic relation R of I_1 by $R \cup \{(a_2, \dots, a_2)\}$ and add relational constraints $\sigma_1(x_i, y_i)$ for every $i \in [n]$ (where the variables y_i are new). Also we replace $\forall x_i$ by $\forall y_i \exists x_i$ for every $i \in [n]$. 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

$$I'_1 \wedge I'_2 \wedge \bigwedge_{u \in \text{Var}(I_1), v \in \text{Var}(I_2)} \sigma(u, v)$$

(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).

Let us guide our proof with an example. Suppose that I_1 is $\exists x_1 \forall x_2 \exists x_3 \forall x_4 \Phi_1(x_1, x_2, x_3, x_4)$ and I_2 is $\forall x'_1 \exists x'_2 \Phi_2(x'_1, x'_2)$ then J has the form

$$\begin{aligned} \exists x_1 \forall y_2 \exists x_2 \exists x_3 \forall y_4 \exists x_4 \forall y'_1 \exists x'_1 \exists x'_2 \quad & \Phi_1(x_1, x_2, x_3, x_4) \wedge \Phi_2(x'_1, x'_2) \wedge \\ & \bigwedge_i \sigma_1(x_i, y_i) \wedge \bigwedge_j \sigma_2(x'_j, y'_j) \wedge \bigwedge_{i,j} \sigma(x_i, x'_j). \end{aligned}$$

($I_1 \in \text{QCSP}(\Gamma_1) \vee I_2 \in \text{QCSP}(\Gamma_2)$) implies $J \in \text{QCSP}(\Gamma)$.) W.l.o.g. $I_1 \in \text{QCSP}(\Gamma_1)$. Let us give a winning strategy for Existential in J based on the winning strategy Existential enjoys on I_1 . Existential will evaluate all (existential) variables of J coming from I_2 as a_1 . It follows that all of the atoms in J arising purely from I_2 , in our example $\Phi_2(x'_1, x'_2)$ and $\bigwedge_j \sigma_2(x'_j, y'_j)$, will be satisfied. There is no longer a need to worry about how Universal plays in J on variables y'_i coming from I_2 . Now, consider Universal's play in J on a variable y_i coming from I_1 . If he plays some pair (a, b) , with $a \in A_1$ and $b \in A_2$, then let Existential respond with x_i set to a . If he plays some element $a \in A_1 \cup A_2 \cup \{a_1, a_2\}$, then let Existential respond with x_i set to any arbitrary $a \in A_1$ (we could imagine this corresponding to Universal instead having played some (a, b) for y_i). Finally, Existential plays the remaining (existential) variables of J matching her winning strategy in the game on I_1 , supposing Universal plays x_i in I_1 whatever we just described Existential playing for x_i (associated with y_i in J). This is a winning strategy on J by construction of Γ .

($J \in \text{QCSP}(\Gamma)$ implies $I_1 \in \text{QCSP}(\Gamma_1) \vee I_2 \in \text{QCSP}(\Gamma_2)$.) Consider a winning strategy for Existential in $J \in \text{QCSP}(\Gamma)$ where Universal only played 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 played by Existential 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 . Existential can now witness $I_1 \in \text{QCSP}(\Gamma_1)$ by considering that Universal in I_1 plays a , where Universal in J played (a, b) . 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 a_2 . Existential can now witness $I_2 \in \text{QCSP}(\Gamma_1)$ by considering Universal in I_2 plays b , where Universal in J played (a, b) .

We can reduce a more complicated set of instances $I_1, \dots, I_n$ of $\text{QCSP}(\Gamma_1)$ and $J_1, \dots, J_n$ of $\text{QCSP}(\Gamma_2)$ to K in $\text{QCSP}(\Gamma)$, in such a way that $K \in \text{QCSP}(\Gamma)$ iff $(I_1 \in \text{QCSP}(\Gamma_1) \vee J_1 \in \text{QCSP}(\Gamma_2)) \wedge \dots \wedge (I_n \in \text{QCSP}(\Gamma_1) \vee J_n \in \text{QCSP}(\Gamma_2))$ by taking the conjunction of our given reduction over each pair I_i and J_i .

Now, let us prove that any instance of $\text{QCSP}(\Gamma)$ can be reduced to some conjunction of instances of $\text{QCSP}(\Gamma_1) \vee \text{QCSP}(\Gamma_2)$. Call an instance K of $\text{QCSP}(\Gamma)$ *connected* if the Gaifman graph of the existential variables of K is connected. The number of connected components of K will give the number of conjuncts $\text{QCSP}(\Gamma_1) \vee \text{QCSP}(\Gamma_2)$. Let us assume now w.l.o.g. that K is connected otherwise we can split K into a conjunction of its connected instances where each connected instance contains all universal variables but only the atoms containing instances of (universal variables and) its existential variables. 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.

Let us remark that Universal has a winning strategy in K iff he has this winning strategy only playing elements of $A_1 \times A_2$. Indeed, we already used this property in the simpler case above. Suppose we have in K some $\sigma_1(x_1, y) \wedge \sigma_1(x_2, y)$, where y is universally quantified before x_1 and x_2 . Then adding the constraint $x_1 = x_2$ doesn't change the result (since y will be played in $A_1 \times A_2$). Let us do this and propagate out the innermost of x_1 and x_2 .

Let us process likewise similar instances in K of the form $\sigma_2(x_1, y) \wedge \sigma_2(x_2, y)$.

Suppose we have in K some $\sigma_1(x_1, y) \wedge \sigma_2(x_2, y)$, where y is universally quantified before x_1 and x_2 . Note that, since we applied already previous rewriting rules, these are the only occurrences of y in the whole sentence. Owing to this, and since $\forall y \exists x_1 \exists x_2 \sigma_1(x_1, y) \wedge \sigma_2(x_2, y)$ is logically equivalent to $\forall y_1 \forall y_2 \exists x_1 \exists x_2 \sigma_1(x_1, y_1) \wedge \sigma_2(x_2, y_2)$, we may substitute the quantification and atoms in our sentence of the first form for those of the latter form.

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 $x = a_1$. In the former case, K_2 becomes redundant, and in the latter case, K_1 becomes redundant.

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 . If one of K_1 or K_2 is empty, then we may play all existential variables to a_2 or a_1 , respectively. 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 $I_1, \dots, I_n$ of $\text{QCSP}(\Gamma_1)$, and instances $J_1, \dots, J_n$ 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

$$\begin{aligned} A &= A'_1 \cup A_1 \cup A_2 \cup \{a_1, a_2\}, \\ \sigma &= (A_1 \times \{a_1\}) \cup (\{a_2\} \times A_2), \\ \sigma_1 &= \{(a, a') \mid a \in A_1\} \cup (\{a_2\} \times A) \cup (A_1 \times (A_1 \cup A_2 \cup \{a_1, a_2\})), \\ \Gamma &= \{R \cup \{(a_2, \dots, a_2)\} \mid R \in \Gamma_1\} \cup \{R \cup \{(a_1, \dots, a_1)\} \mid R \in \Gamma_2\} \cup \{\sigma_1, \sigma\}. \end{aligned}$$

□

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 $|A_1| \cdot |A_2| + |A_1| + |A_2|$ 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

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

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

□

6 Reductions and hardness results for the 3-element domain

In this section we consider the domain $A = \{0, 1, 2\}$ and prove all the hardness results we need for the 3-element domain.

Lemma 17. Suppose $b \in \{0, 1\}$,

$$\begin{aligned} \sigma_0 &= \{(a_1, a_2, a_3) : a_1 \in A, a_2, a_3 \in \{b, 2\}, (a_1 \in \{0, 2\} \vee a_2 = a_3)\}, \\ \sigma_1 &= \{(a_1, a_2, a_3) : a_1 \in A, a_2, a_3 \in \{b, 2\}, (a_1 \in \{1, 2\} \vee a_2 = a_3)\}. \end{aligned}$$

Then $\text{QCSP}(\{\sigma_0, \sigma_1, \{b\}, \{2\}\})$ 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 [26]) and consequently also PSpace-hard (as PSpace is closed under complement). Consider an instance I of co-QNAE3SAT of the form

$$Q_1 x_1 Q_2 x_2 \dots Q_n x_n \text{AE}_3(x_{a_1}, x_{b_1}, x_{c_1}) \vee \cdots \vee \text{AE}_3(x_{a_s}, x_{b_s}, x_{c_s}),$$

where $Q_1, Q_2, \dots, Q_n \in \{\forall, \exists\}$ and $\text{AE}_3 = \{(0, 0, 0), (1, 1, 1)\}$.

By $p_k(x_1, \dots, x_k)$ we denote the predicate (relation) on $\{0, 1\}$ defined by

$$Q_{k+1}x_{k+1}Q_{k+2}x_{k+2}\dots Q_nx_n \text{AE}_3(x_{a_1}, x_{b_1}, x_{c_1}) \vee \dots \vee \text{AE}_3(x_{a_s}, x_{b_s}, x_{c_s}).$$

We inductively define a predicate $\delta_k(x_1, \dots, x_k, y, y')$ for $k = n, n-1, \dots, 1, 0$ satisfying the following conditions

1. if $y, y' \in \{b, 2\}$, $y \neq y'$, $x_1, \dots, x_k \in \{0, 1\}$ then $\delta_k(x_1, \dots, x_k, y, y') = p_k(x_1, \dots, x_k)$;
2. If $y = y' \in \{b, 2\}$ then $\delta_k(x_1, \dots, x_k, y, y')$ holds;
3. If $\delta_k(x_1, \dots, x_k, y, y')$ holds, then it holds if we replace some of the values $x_1, \dots, x_k$ by 2;
4. δ_k is definable by a quantified formula over $\{\sigma_0, \sigma_1, \{b\}, \{2\}\}$ that can be efficiently computed.

Put

$$\delta_n(x_1, \dots, x_n, y_0, y_{2s}) = \exists y_1 \dots \exists y_{2s-1} \bigwedge_{1 \leq i \leq s} (\sigma_0(x_{a_i}, y_{i-1}, y_i) \wedge \sigma_0(x_{b_i}, y_{i-1}, y_i) \wedge \sigma_0(x_{c_i}, y_{i-1}, y_i)) \\ (\sigma_1(x_{s+a_i}, y_{s+i-1}, y_{s+i}) \wedge \sigma_1(x_{s+b_i}, y_{s+i-1}, y_{s+i}) \wedge \sigma_1(x_{s+c_i}, y_{s+i-1}, y_{s+i}))$$

If $Q_k = \forall$ then we put $\delta_{k-1}(x_1, \dots, x_{k-1}, y, y') = \forall x_k \delta_k(x_1, \dots, x_k, y, y')$.

If $Q_k = \exists$ then we put

$$\delta_{k-1}(x_1, \dots, x_{k-1}, y, y') = \exists u \forall x_k \exists u' \delta_k(x_1, \dots, x_k, u, u') \wedge \sigma_0(x_k, y, u') \wedge \sigma_1(x_k, y', u').$$

Let us check that δ_n satisfies the above properties (1)-(4). If $p_n(x_1, \dots, x_n)$ holds, then there exists i such that $x_{a_i} = x_{b_i} = x_{c_i} = d$. If $d = 0$ then to satisfy $\delta_n(x_1, \dots, x_n, y_0, y_{2s})$ we send $y_1, y_2, \dots, y_{i-1}$ to y_0 , and $y_i, y_{i+1}, \dots, y_{2s-1}$ to y_{2s} . If $d = 1$ then we send $y_1, y_2, \dots, y_{s+i-1}$ to y_0 , and $y_{s+i}, y_{s+i+1}, \dots, y_{2s-1}$ to y_{2s} . Suppose $\delta_n(x_1, \dots, x_n, y_0, y_{2s})$ holds on $y, y' \in \{b, 2\}$, $y \neq y'$, $x_1, \dots, x_k \in \{0, 1\}$. Then there should be i such that $y_{i-1} \neq y_i$. If $i \leq s$ then $x_{a_i} = x_{b_i} = x_{c_i} = 0$, if $i > s$ then $x_{a_{i-s}} = x_{b_{i-s}} = x_{c_{i-s}} = 1$. Thus, in both cases $p_n(x_1, \dots, x_n)$ holds, which completes the proof of (1). To prove (2) it is sufficient to send $y_1, \dots, y_{2s-1}$ to $y = y'$. (3) and (4) follow from the definition of δ_n .

Let us prove by induction on k that $\delta_{n-1}, \dots, \delta_0$ satisfy the above properties (1)-(4). Suppose they hold for δ_k . (4) for δ_{k-1} follows from the definition and (4) for δ_k . (3) for δ_{k-1} follows from the definition and the properties of σ_0 and σ_1 . If $Q_k = \forall$ then (1) for δ_{k-1} follows from (1) and (3) for δ_k .

Suppose $Q_k = \exists$. Notice that the Universal player in the definition of δ_{k-1} should always play $x_k = 0$ if $u = y$ and $x_k = 1$ if $u = y'$, since otherwise a winning strategy for the Existential player is to play $u' = u$. Thus, the Existential player controls the choice of x_k , and we have

$$u = y \wedge (\forall x_k \exists u' \delta_k(x_1, \dots, x_k, u, u') \wedge \sigma_0(x_k, y, u') \wedge \sigma_1(x_k, y', u')) = \\ \exists u' \delta_k(x_1, \dots, x_{k-1}, 0, y, u') \wedge \sigma_0(0, y, u') \wedge \sigma_1(0, y', u') = \\ \delta_k(x_1, \dots, x_{k-1}, 0, y, y') \\ u = y' \wedge (\forall x_k \exists u' \delta_k(x_1, \dots, x_k, u, u') \wedge \sigma_0(x_k, y, u') \wedge \sigma_1(x_k, y', u')) = \\ \exists u' \delta_k(x_1, \dots, x_{k-1}, 1, y', u') \wedge \sigma_0(1, y, u') \wedge \sigma_1(1, y', u') = \\ \delta_k(x_1, \dots, x_{k-1}, 1, y', y)$$

From the above equations and (1) for δ_k we derive (1) for δ_{k-1} . To prove (2) for δ_{k-1} we just check that if $y = y'$ then the Existential player can play $u = u' = y = y'$. Thus, we proved the properties (1)-(4) for δ_{k-1} , and by the inductive assumption we conclude that they hold for δ_0 . Since the quantified formula over $\{\sigma_0, \sigma_1, \{b\}, \{2\}\}$ defining δ_0 is efficiently computable, it remains to check that $\exists y \exists y' \delta_0(y, y') \wedge y = b \wedge y = 2$ holds if and only if I holds. $\square$

Recall that an operation f is 01-stable if it is idempotent, $f(x, 0, 1) = x$, and $f(x, 2, 2) = 2$.

Lemma 18. *Suppose Γ is preserved by s_2 and a 01-stable operation h_{01} . Then an instance*

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

of QCSP(Γ) is equivalent to

$$\forall x_1 \forall x_2 \dots \forall x_n \exists \exists ((\exists' \exists' \Phi_1^0 \wedge \Phi_1^1) \wedge (\exists' \exists' \Phi_2^0 \wedge \Phi_2^1) \wedge \dots \wedge (\exists' \exists' \Phi_n^0 \wedge \Phi_n^1)),$$

where

$$\begin{aligned} \Phi_i^0 &= \Phi_{x_{i+1}^0, \dots, x_n^0, y_{i+1}^0, \dots, y_n^0}^{x_{i+1}, \dots, x_n, y_{i+1}, \dots, y_n} \wedge x_{i+1}^0 = 0 \wedge \dots \wedge x_n^0 = 0, \\ \Phi_i^1 &= \Phi_{x_{i+1}^1, \dots, x_n^1, y_{i+1}^1, \dots, y_n^1}^{x_{i+1}, \dots, x_n, y_{i+1}, \dots, y_n} \wedge x_{i+1}^1 = 1 \wedge \dots \wedge x_n^1 = 1, \end{aligned}$$

(note that $\Phi_n^0 = \Phi_n^1 = \Phi$) and by $\exists \exists$ and $\exists' \exists'$ we mean that we add all necessary existential quantifiers for y -variables without a superscript and with a superscript, 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 $x_j^0 = 0$, $x_j^1 = 1$, $y_j^0 = f_j(x_1, \dots, x_i, 0, \dots, 0)$, and $y_j^1 = f_j(x_1, \dots, x_i, 1, \dots, 1)$ in the definition of $\Phi_i^0 \wedge \Phi_i^1$ for every $j > i$.

(Backwards/ upwards.) First, introduce several notations. By $\vee$ denote s_2 , by $\leq$ denote the partial order on $\{0, 1, 2\}$ such that $0 \leq 2$ and $1 \leq 2$ but 0 and 1 are incomparable. For an operation $f(x_1, \dots, x_s)$ by $f(x_1, \dots, x_t)$, where $t < s$ we denote $\bigvee_{(a_{t+1}, \dots, a_s)} f(x_1, \dots, x_t, a_{t+1}, \dots, a_s)$.

By h'_{01} we denote the operation defined by $h'_{01}(x, y, z) = s_2(x, h_{01}(x, y, z))$. Notice that h'_{01} is also a 01-stable operation, but it satisfies the property $h'_{01}(x, y, z) \geq x$.

Let us show how to build a solution to the original instance from a solution of the new instance. Let a solution of the new instance be defined by $y_i = f_i(x_1, \dots, x_n)$, $y_i^0 = g_{j,i}^0(x_1, \dots, x_n)$ and $y_i^1 = g_{j,i}^1(x_1, \dots, x_n)$ for the variables y_i^0 and y_i^1 from $\Phi_j^0 \wedge \Phi_j^1$.

First, let us show that

$$(x_1, \dots, x_j, c, \dots, c, f_1(x_1, \dots, x_j), \dots, f_j(x_1, \dots, x_j), g_{j,j+1}^c(x_1, \dots, x_j), \dots, g_{j,n}^c(x_1, \dots, x_j))$$

is a solution of Φ for every $j \in [n]$ and $c \in \{0, 1\}$. In fact, from Φ_j^c the tuple

$$(x_1, \dots, x_j, c, \dots, c, f_1(x_1, \dots, x_n), \dots, f_j(x_1, \dots, x_n), g_{j,j+1}^c(x_1, \dots, x_n), \dots, g_{j,n}^c(x_1, \dots, x_n))$$

is a solution of Φ . Consider all the evaluations of the variables $x_{j+1}, \dots, x_n$ to obtain 3^{n-j} solutions of Φ , and apply the semilattice s_2 to them. As a result we obtain the required tuple, which implies that this tuple satisfies Φ .

We prove by induction on $N = 0, 1, \dots, n$ that there exist operations $q_1, \dots, q_n$ satisfying

1. $(x_1, \dots, x_n, q_1(x_1, \dots, x_n), \dots, q_n(x_1, \dots, x_n))$ is a solution of Φ ,
2. $q_i(x_1, \dots, x_n) \geq f_i(x_1, \dots, x_n)$ for every $i \in [n]$,
3. $q_i(x_1, \dots, x_n)$ depends fictitiously on $x_{i+1}, \dots, x_n$ for every $i \in [N]$,

4. $q_i(x_1, \dots, x_n)$ is uniquely determined by $x_1, \dots, x_i$ and $f_i(x_1, \dots, x_n)$ for every $i \in [n]$.

For $N = 0$ we can put $q_i(x_1, \dots, x_n) = f_i(x_1, \dots, x_n)$ for every $i \in [n]$. Let us prove the induction step. Assume that $q_1, \dots, q_n$ satisfy conditions (1)-(4) for some N . Apply the 01-stable operation h'_{01} to the three solutions of Φ

$$\begin{aligned} & (x_1, \dots, x_n, q_1(x_1, \dots, x_n), \dots, q_n(x_1, \dots, x_n)) \\ & (x_1, \dots, x_{N+1}, 0, \dots, 0, f_1(x_1, \dots, x_{N+1}), \dots, f_{N+1}(x_1, \dots, x_{N+1}), \\ & \quad g_{N+1, N+2}^0(x_1, \dots, x_{N+1}), \dots, g_{N+1, n}^0(x_1, \dots, x_{N+1})) \\ & (x_1, \dots, x_{N+1}, 1, \dots, 1, f_1(x_1, \dots, x_{N+1}), \dots, f_{N+1}(x_1, \dots, x_{N+1}), \\ & \quad g_{N+1, N+2}^1(x_1, \dots, x_{N+1}), \dots, g_{N+1, n}^1(x_1, \dots, x_{N+1})) \end{aligned}$$

The first n coordinates of the obtained tuple are $(x_1, \dots, x_n)$ and we denote this tuple by $(x_1, \dots, x_n, q'_1(x_1, \dots, x_n), \dots, q'_n(x_1, \dots, x_n))$. Let us show that $q'_1, \dots, q'_n$ satisfy (1)-(4) for $N := N + 1$. Condition (1) follows from the fact that h'_{01} preserves Γ . Condition (2) follows from the fact that $q_1, \dots, q_n$ satisfy (2) and $h'_{01}(x, y, z) \geq x$.

For $i \in [N]$ we have

$$q'_i(x_1, \dots, x_n) = h'_{01}(q_i(x_1, \dots, x_n), f_i(x_1, \dots, x_{N+1}), f_i(x_1, \dots, x_{N+1})).$$

Let us show that q'_i depends essentially only on $x_1, \dots, x_i$. Since q_i depends fictitiously on $x_{i+1}, \dots, x_n$ and $q_i(x_1, \dots, x_n) \geq f_i(x_1, \dots, x_n)$, we have $q_i(x_1, \dots, x_n) = q_i(x_1, \dots, x_i) \geq f_i(x_1, \dots, x_i)$. If $f_i(x_1, \dots, x_i) = 2$ then $q_i(x_1, \dots, x_i) = 2$ and $q'_i(x_1, \dots, x_n) = 2$. If $f_i(x_1, \dots, x_i) = d \neq 2$, then $f_i(x_1, \dots, x_{N+1}) = d$ and $q_i(x_1, \dots, x_i) \in \{d, 2\}$. Therefore, $q'_i(x_1, \dots, x_n) = q_i(x_1, \dots, x_i)$ and q'_i depends fictitiously on $x_{i+1}, \dots, x_n$ and satisfies (3) and (4).

We have

$$q'_{N+1}(x_1, \dots, x_n) = h'_{01}(q_{N+1}(x_1, \dots, x_n), f_{N+1}(x_1, \dots, x_{N+1}), f_{N+1}(x_1, \dots, x_{N+1})).$$

If $f_{N+1}(x_1, \dots, x_{N+1}) = 2$ then $q'_{N+1}(x_1, \dots, x_n)$ equals 2, otherwise it is uniquely determined by $x_1, \dots, x_{N+1}$, and $f_{N+1}(x_1, \dots, x_n) = f_{N+1}(x_1, \dots, x_{N+1})$. That is, in all the cases q'_{N+1} is uniquely determined by $x_1, \dots, x_{N+1}$ and satisfies (3) and (4).

We have $q'_i(x_1, \dots, x_n) = h'_{01}(q_i(x_1, \dots, x_n), g_{N+1, i}^0(x_1, \dots, x_{N+1}), g_{N+1, i}^0(x_1, \dots, x_{N+1}))$ for $i > N + 1$. Since q_i is uniquely determined by $x_1, \dots, x_i$ and $f_i(x_1, \dots, x_n)$, the same is true for q'_i , that is, q'_i satisfies (4). Thus, we defined operations $q'_1, \dots, q'_n$ satisfying conditions (1)-(4) with $N := N + 1$.

Hence, by the induction there exist operations $q_1, \dots, q_n$ satisfying conditions (1)-(4) for $N = n$. We may check that $(x_1, \dots, x_n, q_1(x_1, \dots, x_n), \dots, q_n(x_1, \dots, x_n))$ is a solution of the original instance, which completes the proof. $\square$

The next lemma follows from Lemma 18 and the fact that if Γ is preserved by a semilattice then $\text{CSP}(\Gamma, 0, 1)$ can be solved in polynomial time. Nevertheless, to explain how a 01-stable operation can be used in an algorithm we give an alternative proof.

Lemma 19. *Suppose Γ is preserved by s_2 and a 01-stable operation g . Then $\text{QCSP}(\Gamma)$ 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 check the satisfiability of the two instances

$$\Phi \wedge x_1 = a_1 \wedge x_2 = x_3 = \dots = x_n = 0,$$

$$\Phi \wedge x_1 = a_1 \wedge x_2 = x_3 = \dots = x_n = 1.$$

These are CSP instances, which can be solved in polynomial time because the semilattice preserves $(\Gamma, 0, 1)$. We check whether both have a solution with $y_1 = 0$, $y_1 = 1$, $y_1 = 2$ (we solve six instances). Let Y be the set of possible values for y_1 such that both instances have solutions. If Y is empty, then the QCSP instance does not hold and we are done. 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 an instance that does not hold. If $|Y| > 1$ then $2 \in Y$ due to the semilattice polymorphism. Let the solutions of the CSP instances with $y_1 = 2$ be $(x_1, y_1, x_2, y_2, \dots, x_n, y_n) = (a_1, 2, 0, c_2, \dots, 0, c_n)$ and $(x_1, y_1, x_2, y_2, \dots, x_n, y_n) = (a_1, 2, 1, d_2, \dots, 1, d_n)$. 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 g to this solution, $(a_1, 2, 0, c_2, \dots, 0, c_n)$, and $(a_1, 2, 1, d_2, \dots, 1, d_n)$, 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 $y_2, y_3, \dots, y_n$. $\square$

We say that a ternary relation R is an *AND-type* relation if $R \cap (\{0, 1\} \times \{0, 1\} \times \{0, 1, 2\}) = \begin{pmatrix} 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 \end{pmatrix}$, that is, it is row-wise the truth table of AND. If s_2 preserves R , then it also contains $(2, 2, 0)$ and $(2, 2, 2)$. We say that a ternary relation R is an *OR-type* relation if

$R \cap (\{0, 1\} \times \{0, 1\} \times \{0, 1, 2\}) = \begin{pmatrix} 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 1 \end{pmatrix}$, that is, it is row-wise the truth table of OR.

Again, if s_2 preserves R , then it also contains $(2, 2, 1)$ and $(2, 2, 2)$. One can readily imagine AND/OR-type relations of arity $k + 1$ higher than three built from AND/OR operations of arity k .

Lemma 20. *Suppose Γ contains an AND-type relation and an OR-type relation, and Γ is preserved by s_2 . Then QCSP(Γ) is co-NP-hard.*

Proof. By $R_{and,2}$ and $R_{or,2}$ we denote an AND-type and an OR-type relation, respectively. 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

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

It follows from the definition that $\xi_n \cap \{0, 1\}^{3n}$ is defined by $AE_3(x_1, y_1, z_1) \vee AE_3(x_2, y_2, z_2) \vee \dots \vee AE_3(x_n, y_n, z_n)$, where $AE_3 = \{(0, 0, 0), (1, 1, 1)\}$.

Suppose $R(y_1, \dots, y_t) = \xi_n(y_{i_1}, y_{i_2}, \dots, y_{i_{3n}})$, where $i_1, \dots, i_{3n} \in \{1, 2, \dots, t\}$. Let us show that $\forall y_1 \dots \forall y_t \xi_n(y_{i_1}, y_{i_2}, \dots, y_{i_{3n}})$ holds if $\xi_n(y_{i_1}, y_{i_2}, \dots, y_{i_{3n}})$ holds for all $y_1, \dots, y_t \in \{0, 1\}$.

We need to prove that if $\{0, 1\}^t \subseteq R$ then $A^t \subseteq R$, which follows from the fact that the semilattice s_2 preserves R .

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:

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

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 $\text{QCSP}(\Gamma)$, which completes the proof. $\square$

Lemma 21. *Suppose $\Gamma \subseteq \text{Inv}(s_2)$, Γ contains the constant 0, $\delta(x, y, z) = (x \neq 0) \vee (y = z)$ and an AND-type relation R_2 . Then $\text{QCSP}(\Gamma)$ is co-NP-hard.*

Proof. Here we will define a reduction from the complement of $\text{CSP}(\{1\text{IN}3\})$, which is known

to be NP-complete [28], where $1\text{IN}3 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$. First, put

$$\delta_1(x_1, x_2, x_3, x_4) = \exists t \delta(x_1, x_3, t) \wedge \delta(x_2, t, x_4).$$

It is not hard to see that $\delta_1(x_1, x_2, x_3, x_4) = (x_1 \neq 0) \vee (x_2 \neq 0) \vee (x_3 = x_4)$. Put

$$\delta_2(x_1, x_2, x_3, x_4) = \delta_1(x_1, x_2, x_3, x_4) \wedge \delta_1(x_1, x_3, x_2, x_4) \wedge \delta_1(x_2, x_3, x_1, x_4).$$

By the definition, the first three variables of δ_2 are symmetric. Suppose $(a, b, c, d) \in \delta_2$. It is not hard to see that $a = b = c = 0$ implies $d = 0$. Also $a = 1, b = c = 0$ implies $d = 1$. If there are at least two 1s in (a, b, c) then d can be arbitrary. Put $R'(x, y) = R_2(x, x, y)$. By the definition of an AND-type relation, R' contains $(0, 0), (1, 1), (2, 2), (2, 0)$ but doesn't contain $(0, 1), (0, 2), (1, 0), (1, 2)$. Put

$$\delta_3(x_1, x_2, x_3, x_4) = \exists x'_1 \exists x'_2 \exists x'_3 \delta_2(x'_1, x'_2, x'_3, x_4) \wedge R'(x_1, x'_1) \wedge R'(x_2, x'_2) \wedge R'(x_3, x'_3).$$

Then $(a, b, c, 0) \in \delta_3$ for all tuples (a, b, c) but the tuples from $1\text{IN}3$.

Again, we apply the recursive formula to build an n -ary “and”

$$R_{n+1}(x_1, \dots, x_{n+1}, z) = \exists t R_n(x_1, \dots, x_n, t) \wedge R_2(t, x_{n+1}, z).$$

Define $\zeta_n(x_1, \dots, x_{3n})$ by

$$\exists z_1 \dots \exists z_n \exists t R_n(z_1, \dots, z_n, t) \wedge \delta_3(x_1, x_2, x_3, z_1) \wedge \dots \wedge \delta_3(x_{3n-2}, x_{3n-1}, x_{3n}, z_n) \wedge (t = 0).$$

Now we may encode the complement of $\text{CSP}(\{1\text{IN}3\})$ using Γ . If $\overline{1\text{IN}3}$ is the complement of $1\text{IN}3$ with respect to $\{0, 1\}^3$, then we use a formula of the following form:

$$\forall y_1 \dots \forall y_t \overline{1\text{IN}3}(y_{i_1}, y_{i_2}, y_{i_3}) \vee \dots \vee \overline{1\text{IN}3}(y_{i_{3n-2}}, y_{i_{3n-1}}, y_{i_{3n}}),$$

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 $\text{QCSP}(\Gamma)$, which completes the proof. $\square$

Lemma 22. *Suppose $\sigma = \{0, 2\}^2 \cup \{1, 2\}^2$,*

$$\begin{aligned} \sigma_0 &= \{(a_1, a_2, a_3) : a_1 \in A, a_2, a_3 \in \{1, 2\}, (a_1 \in \{0, 2\} \vee a_2 = a_3)\}, \\ \sigma_1 &= \{(a_1, a_2, a_3) : a_1 \in A, a_2, a_3 \in \{0, 2\}, (a_1 \in \{1, 2\} \vee a_2 = a_3)\}. \end{aligned}$$

Then $\text{QCSP}(\{\sigma, \sigma_0, \sigma_1, \{0\}, \{1\}\})$ is co-NP-hard.

Proof. Let us encode the complement of *Not-All-Equal 3-Satisfiability* using $\{\sigma, \sigma_0, \sigma_1, \{0\}, \{1\}\}$. An instance $\mathcal{I}$ of this complement can be expressed by a formula of the following form:

$$\forall x_1 \dots \forall x_t \text{AE}_3(x_{a_1}, x_{b_1}, x_{c_1}) \vee \dots \vee \text{AE}_3(x_{a_s}, x_{b_s}, x_{c_s}),$$

where $\text{AE}_3 = \{(0, 0, 0), (1, 1, 1)\}$. Let

$$\begin{aligned} \Phi_0 &= \bigwedge_{1 \leq i \leq s} (\sigma_0(x_{a_i}, y_{i-1}, y_i) \wedge \sigma_0(x_{b_i}, y_{i-1}, y_i) \wedge \sigma_0(x_{c_i}, y_{i-1}, y_i)) \\ \Phi_1 &= \bigwedge_{1 \leq i \leq s} (\sigma_1(x_{a_i}, z_{i-1}, z_i) \wedge \sigma_1(x_{b_i}, z_{i-1}, z_i) \wedge \sigma_1(x_{c_i}, z_{i-1}, z_i)) \\ \Psi &= \forall x_1 \dots \forall x_t \exists y_0 \exists y_1 \dots \exists y_s \exists z_0 \exists z_1 \dots \exists z_s (y_0 = 1 \wedge \Phi_0 \wedge z_0 = 0 \wedge \Phi_1 \wedge \sigma(y_s, z_s)). \end{aligned}$$

Let us show that $\mathcal{I}$ is equivalent to Ψ .

($I \in \text{co-NAE3SAT}$ implies $\Psi \in \text{QCSP}(\{\sigma, \sigma_0, \sigma_1\})$). Suppose in Ψ that Universal plays some sequence of 0s, 1s and 2s for the x_i s. Let this be duplicated by Universal in I where we additionally consider 2 in Ψ as 1 in I . We may assume that the resulting instance of co-NAE3SAT has a clause j that is either $(0, 0, 0)$ or $(1, 1, 1)$. Suppose first it is $(1, 1, 1)$, which arises from a setting in Ψ of $(x_{a_j}, x_{b_j}, x_{c_j}) \in \{1, 2\}^3$. Existential plays in Ψ as follows. $y_0, \dots, y_{j-1}, y_j, \dots, y_s$ are set to 1, $z_0, \dots, z_{j-1}$ are set to 0 and $z_j, \dots, z_s$ are set to 2. Let us look at the interesting conjunction in Φ_1 , which is $\sigma_1(x_{a_j}, z_{j-1}, z_j) \wedge \sigma_1(x_{b_j}, z_{j-1}, z_j) \wedge \sigma_1(x_{c_j}, z_{j-1}, z_j)$, and note that this is satisfied trivially as $(x_{a_j}, x_{b_j}, x_{c_j}) \in \{1, 2\}^3$. The fact that the other parts are satisfied is straightforward since $y_i = y_{i-1}$ and $z_i = z_{i-1}$. Suppose now that it is $(0, 0, 0)$, which arises only from a setting in Ψ of $(x_{a_j}, x_{b_j}, x_{c_j}) = (0, 0, 0)$. Existential plays in Ψ as follows. $z_0, \dots, z_{j-1}, z_j, \dots, z_s$ are set to 0, $y_0, \dots, y_{j-1}$ are set to 1 and $y_j, \dots, y_s$ are set to 2. This case concludes as the previous.

($\Psi \in \text{QCSP}(\{\sigma, \sigma_0, \sigma_1\})$ implies $I \in \text{co-NAE3SAT}$). Let us consider Universal playing in I and we duplicate this in Ψ (in other words, we only consider Universal plays in Ψ on $\{0, 1\}$). We cannot have $y_0, \dots, y_s = 1$ and $z_0, \dots, z_s = 0$ because we violate $\sigma(y_s, z_s)$. Thus, at least at some point in one of these sequences, we make a transition to 2. First, let us assume that it is in the y_i at the point $y_{j-1} = 1$ and $y_j = 2$. Then the conjunction $\sigma_0(x_{a_j}, y_{j-1}, y_j) \wedge \sigma_0(x_{b_j}, y_{j-1}, y_j) \wedge \sigma_0(x_{c_j}, y_{j-1}, y_j)$ enforces that $(x_{a_j}, x_{b_j}, x_{c_j}) = (0, 0, 0)$. Now let us assume that it is in the z_i at the point $z_{j-1} = 0$ and $z_j = 2$. Then the conjunction $\sigma_1(x_{a_j}, z_{j-1}, z_j) \wedge \sigma_1(x_{b_j}, z_{j-1}, z_j) \wedge \sigma_1(x_{c_j}, z_{j-1}, z_j)$ enforces that $(x_{a_j}, x_{b_j}, x_{c_j}) = (1, 1, 1)$. Thus, in both cases, we deduce that I is a no-instance of NAE3SAT.

Thus, we reduced a co-NP-complete problem to $\text{QCSP}(\{\sigma, \sigma_0, \sigma_2, \{0\}, \{1\}\})$, which completes the proof. $\square$

7 Strange structure 1

In this section we will define a constraint language Γ consisting of just 2 relations and constants such that $\text{Pol}(\Gamma)$ 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 $\text{QCSP}(\Gamma)$ can be solved in polynomial time.

Let $A = \{0, 1, 2\}$. Recall that τ_n for $\alpha = \{0, 2\}$ and $\beta = \{1, 2\}$ is the $3n$ -ary relation defined by

$$\{(x_1, y_1, z_1, x_2, y_2, z_2, \dots, x_n, y_n, z_n) \mid \exists i: \{0, 1\} \not\subseteq \{x_i, y_i, z_i\}\}.$$

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

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

Note τ_n can be pp-defined from σ_n but the obvious definition is of size exponential in n (see [11]). At the same time, σ_n can be pp-defined from τ_n by identification of variables. Let

$$\begin{aligned} R_{and,2} &= \{(0,0,0), (1,0,0), (0,1,0), (1,1,1)\} \cup \{(2,a,b), (a,2,b) \mid a,b \in A\}, \\ \delta &= \{(0,0), (1,0), (2,0), (1,2), (2,2)\}, \\ \Gamma &= \{R_{and,2}, \delta, \{0\}, \{1\}, \{2\}\}. \end{aligned}$$

The relation ρ of arity $2n$ omitting just one tuple $1^n 0^n$ 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

$$\begin{aligned} \rho(x_1, \dots, x_n, y_1, \dots, y_n) &= \exists y'_1 \dots \exists y'_n \exists z \exists t R_{and,n}(x_1, \dots, x_n, z) \wedge \\ &\quad \delta(y_1, y'_1) \wedge \dots \wedge \delta(y_n, y'_n) \wedge R_{and,n}(y'_1, \dots, y'_n, t) \wedge R_{and,2}(z, z, t). \end{aligned}$$

As a conjunction of 2^n copies of ρ with permuted variables we can define the relation σ_n but this definition will be of exponential size. Then, we know from [31] that $\text{Pol}(\Gamma)$ has the EGP property, and from [11] 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 $\text{QCSP}(\Gamma)$ can be solved in polynomial time. In the following $<$ is the coordinatewise partial order on $\{0,1\}^n$ built from $0 < 1$. Recall that for a conjunctive formula Φ by $\Phi(x_1, \dots, x_n)$ we denote the set of all tuples $(a_1, \dots, a_n)$ such that Φ has a solution with $(x_1, \dots, x_n) = (a_1, \dots, a_n)$.

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*

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

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 $(x_1, \dots, x_n) = \beta$. Thus, we need to prove that there exists a variable y such that $C_y = \{0\}$ and $D_y = \{1\}$.

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 $\alpha(i) = 1$ and then $C_{x_i} \subseteq \{0,1\}$ $D_{x_i} = \{1\}$. If $C_{x_i} = \{0\}$ then we found the required variable and we are done; otherwise we have $v(x_i) = 1$. 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 $\{0, 2\}$.

□

We may check that operations $s_{0,2}$ and $g_{0,2}$ are polymorphisms of Γ . Moreover, we will show later that $\text{Pol}(\Gamma) = \text{Clo}(\{s_{0,2}, g_{0,2}\})$. Put

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

Since, $s_2(x, y) = s_{0,2}(y, s_{0,2}(x, y))$ and $h_{0,2}(x, y, z) = g_{0,2}(s_{0,2}(x, z), s_2(y, z))$, the operations s_2 and $h_{0,2}$ are also polymorphisms of Γ .

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 $n \geq 3$, 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 $\beta_1, \beta_2, \beta_3 \in \sigma_n$.

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

$$\beta_3(i) = \begin{cases} \alpha_1(i), & \text{if } \alpha_1(i) = \alpha_2(i) \text{ or } i \leq 4. \\ 0, & \text{otherwise} \end{cases}.$$

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 $(x_1, \dots, x_{2n}) \in \{\beta_1, \beta_2\}$, and it should be equal to 0 in any solution of Φ such that $(x_1, \dots, x_{2n}) = \beta_3$. Since $h_{0,2}(\beta_1, \beta_2, \beta_3) = \beta_1$, $h_{0,2}(1, 1, 0) = 2$ and $h_{0,2}$ is a polymorphism

of Γ , we get a contradiction to the uniqueness of d in Lemma 23. 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 $(x_1, \dots, x_{2n}) \in \{\beta_1, \beta_2\}$. 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 $(x_1, \dots, x_{2n}) = \beta_3$). 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 would obtain $h_{0,2}(1, 1, 0) = 1$. However, in fact $h_{0,2}(1, 1, 0) = 2$, which, since $h_{0,2}$ is a polymorphism of Γ , contradicts the uniqueness of d in Lemma 23.

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

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.*

Let us characterize the clone $\text{Pol}(\Gamma)$. First, let us define relations on the set A . Put $\epsilon_2(x_1, x_2) = (x_1 = 0) \vee (x_1 = 1 \wedge x_2 \neq 2)$, $\zeta_2(x_1, x_2) = (x_1 = 0) \vee (x_1 = 1 \wedge x_2 = 1)$, $\epsilon_{n+1}(x_1, \dots, x_n, x_{n+1}) = (x_1 = 0) \vee (x_1 = 1 \wedge \epsilon_n(x_2, \dots, x_{n+1}))$, $\zeta_{n+1}(x_1, \dots, x_n, x_{n+1}) = (x_1 = 0) \vee (x_1 = 1 \wedge \zeta_n(x_2, \dots, x_{n+1}))$. Put $\Delta = \{\{1\}, \{0, 1\}, \epsilon_2, \epsilon_3, \epsilon_4, \dots, \zeta_2, \zeta_3, \zeta_4, \dots\}$.

Suppose $B \subseteq \{1, 2, \dots, n\}$. Suppose $\rho \subseteq A^n$ can be defined as a conjunction of relations from Δ and $\text{pr}_B \rho \subseteq \{0, 1\}^{|B|}$. Then we define an operation $f_{B,\rho}(x_0, x_1, \dots, x_n)$ as follows. If $x_i = 0$ for every $i \in \{0\} \cup B$ then it returns 0. If $x_0 = 1$ and $(x_1, \dots, x_n) \in \rho$ then it returns 1. In all other cases it returns 2.

By $\mathcal{C}$ we denote the set of all operations that can be obtained from $f_{B,\rho}$ for all B and ρ by a permutation of variables (it is sufficient to move just the first variable). We now begin our journey towards proving Theorem 29, which says that $\text{Pol}(\Gamma) = \text{Clo}(\{g_{0,2}, s_{0,2}\})$.

Lemma 26. *Suppose $\rho \subseteq A^n$, $(1, 1, \dots, 1) \in \rho$, and in every $(m+1) \times n$ -matrix, whose first m rows are from ρ and all columns are from $R_{\text{and},m}$, the last row is also from ρ . Then ρ can be represented as a conjunction of relations from Δ .*

Proof. We prove by induction on the arity of ρ . If $n = 1$, then $\rho \in \{\{1\}, \{0, 1\}, \{0, 1, 2\}\}$. Thus, either $\rho \in \Delta$, or ρ is full. Assume that $n \geq 2$.

We want to build a conjunction of relations, which we denote by Φ . We start with $\Phi = \emptyset$. For every tuple $\alpha = (a_1, \dots, a_n) \notin \rho$ we add the corresponding constraints to Φ to exclude this tuple. If we can exclude every tuple then Φ defines ρ .

Let $I \subseteq \{1, 2, \dots, n\}$ be a minimal subset such that $\text{pr}_I(\alpha) \notin \text{pr}_I(\rho)$. It is not hard to see that $\text{pr}_I(\rho)$ satisfies all the assumptions of this lemma. Therefore, if $I \neq \{1, 2, \dots, n\}$, then by the inductive assumption $\text{pr}_I(\rho)$ can be represented as a conjunction of relations from Δ . Then, we may add this representation to Φ to exclude α . Thus, we may assume that $I = \{1, 2, \dots, n\}$.

Since I is minimal, there exist $b_1, \dots, b_n$ such that $\beta_j = (a_1, \dots, a_{j-1}, b_j, a_{j+1}, \dots, a_n) \in \rho$ for every j .

There should be at most one i such that $(a_i, b_i, a_i) \in R_{and,2}$. Otherwise (if we have i_1 and i_2), we build a matrix whose rows are $\beta_{i_1}, \beta_{i_2}, \alpha$ for different i_1, i_2 . It is not hard to see that every column of the matrix is from $R_{and,2}$, which means that $\alpha \in \rho$. Contradiction. Hence, there exists at most one i such that $(a_i, b_i, a_i) \in R_{and,2}$.

Notice that $(a, b, a) \in R_{and,2}$ if $a \in \{0, 2\}$ or $b = 2$. Then there should be exactly one i such that $(a_i, b_i, a_i) \in R_{and,2}$ (otherwise $a_i = 1$ for every i , which contradicts the fact that $(1, 1, \dots, 1) \in \rho$). Without loss of generality, let this i be equal to n , then $\alpha = (1, 1, \dots, 1, a_n)$, where $a_n \in \{0, 2\}$.

Assume that $\text{pr}_i(\rho) \not\subseteq \{0, 1\}$ for every $i \in \{1, 2, \dots, n-1\}$, then we consider tuples $\mathbf{b}^1, \dots, \mathbf{b}^{n-1} \in \rho$ such that the i -th element of $\mathbf{b}^i$ is 2, and build a matrix whose rows are $\mathbf{b}^1, \dots, \mathbf{b}^{n-1}, \beta_1, \alpha$. We can check that every column of this matrix is from $R_{and,n}$, therefore $\alpha \in \rho$, which contradicts our assumptions.

Thus, there exists $i \in \{1, 2, \dots, n-1\}$ such that $\text{pr}_i(\rho) \subseteq \{0, 1\}$. Without loss of generality we assume that $i = 1$. Put $\delta(x_2, \dots, x_n) = \rho(1, x_2, \dots, x_n)$. Since any tuple from $R_{and,m}$ starting with m 1s should end with 1, δ satisfies all the assumptions of this lemma. By the inductive assumption, δ can be represented as a conjunction of relations from Δ . In this representation we replace $\epsilon_s(x_{i_1}, \dots, x_{i_s})$ by $\epsilon_{s+1}(x_1, x_{i_1}, \dots, x_{i_s})$, $\zeta_s(x_{i_1}, \dots, x_{i_s})$ by $\zeta_{s+1}(x_1, x_{i_1}, \dots, x_{i_s})$, $x_j \in \{0, 1\}$ by $\epsilon_2(x_1, x_j)$, $x_j \in \{1\}$ by $\zeta_2(x_1, x_j)$ and add the obtained constraints to Φ . Since $a_1 = 1$, we excluded the tuple α from the solution set by adding these constraints. Note that we maintain the property that every tuple $\beta \in \rho$ satisfies Φ . Thus, we can exclude every tuple $\alpha \in A^n \setminus \rho$, which means that Φ defines ρ . $\square$

Lemma 27. $\text{Pol}(\Gamma) \subseteq \mathcal{C}$.

Proof. Suppose $f \in \text{Pol}(\Gamma)$ is an operation of arity $n+1$. Since σ_m can be pp-defined from Γ for every m , f is $\{0, 2\}\{1, 2\}$ -projective [31]. Without loss of generality we assume that it is an $\{0, 2\}\{1, 2\}$ -projection to the first variable.

Let B be the set of all $i \in \{1, 2, \dots, n\}$ such that $f(a_0, \dots, a_n) = 0$ implies $a_i = 0$. Let $\mathbf{a}^1, \dots, \mathbf{a}^s$ be all the tuples such that $f(\mathbf{a}) = 0$. Let $\omega(x_1, \dots, x_s, z)$ be defined by

$$\exists x'_1 \dots \exists x'_s R_{and,s+1}(x'_1, \dots, x'_s, z) \wedge \delta(x_1, x'_1) \wedge \dots \wedge \delta(x_s, x'_s).$$

Let $\mathbf{b} = (0, b_1, \dots, b_n)$ be a tuple such that $b_i = 0$ if $i \in B$. Let us build a matrix whose rows are tuples $\mathbf{a}^1, \dots, \mathbf{a}^s, \mathbf{b}$. By the definition of B , every column of this matrix is from ω . Since f preserves ω , the result of applying f to the matrix (that is $(f(\mathbf{a}^1), \dots, f(\mathbf{a}^s), f(\mathbf{b}))$) should be from ω . But the first s elements of the result equal 0, therefore $f(\mathbf{b}) = 0$. Thus, we proved that if $x_i = 0$ for every $i \in \{0\} \cup B$ then $f(x_0, \dots, x_n) = 0$.

Let ρ be the set of all tuples $(a_1, \dots, a_n)$ such that $f(1, a_1, \dots, a_n) = 1$. Since f preserves $\{1\}$, we have $(1, 1, \dots, 1) \in \rho$.

Assume that for some tuple $\mathbf{a} = (a_1, \dots, a_n) \in \rho$ and $i \in B$ we have $a_i = 2$. Let $\mathbf{c} = (c_1, \dots, c_n)$ be the tuple such that $c_j = 0$ if $j \in B \setminus \{i\}$ and $c_j = 2$ otherwise. Let $\mathbf{d} = (d_1, \dots, d_n)$ be the tuple such that $d_j = 0$ if $j \in B$ and $d_j = 2$ otherwise. It is not hard to see that the rows $\mathbf{a}, \mathbf{d}, \mathbf{c}$ form a matrix whose columns are from $R_{and,2}$. Therefore $(f(1\mathbf{a}), f(0\mathbf{d}), f(0\mathbf{c})) \in R_{and,2}$. Since $f(1\mathbf{a}) = 1$ and $f(0\mathbf{d}) = 0$, we have $f(0\mathbf{c}) = 0$, which contradicts the definition of B . Thus, $\text{pr}_B(\rho) \subseteq \{0, 1\}^{|B|}$.

It remains to prove that ρ can be represented as a conjunction of relations from Δ . Let us show that ρ satisfies the assumptions of Lemma 26. Consider a matrix whose columns are from $R_{and,m}$ and whose first m rows are from ρ . Add a column with 1s in the beginning of the matrix and apply f . Since f preserves $R_{and,m}$, the result should be from $R_{and,m}$. Since the first m elements are equal to 1, the last element should be equal to 1. Therefore, the last row of the matrix is from ρ . It remains to apply Lemma 26 to complete the proof. $\square$

Lemma 28. $\mathcal{C} \subseteq \text{Clo}(\{g_{0,2}, s_{0,2}\})$.

Proof. We need to prove that every operation from $\mathcal{C}$ can be built from $g_{0,2}$ and $s_{0,2}$. For $B \subseteq \{1, 2, \dots, n\}$ by $\psi(B)$ we denote the set of all tuples $\alpha \in A^n$ such that $\text{pr}_B(\alpha) \in \{0, 1\}^{|B|}$. The following formulas show how to generate $f_{\emptyset, \epsilon_n}$, $f_{\emptyset, \zeta_n}$, and $f_{B, \psi(B)}$ for all $n \geq 2$ and $B \neq \emptyset$:

$$\begin{aligned} f_{\emptyset, \{0,1\}}(x_0, x_1) &= g_{0,2}(x_0, x_1), \\ f_{\emptyset, \{1\}}(x_0, x_1) &= g_{0,2}(x_0, s_2(x_0, x_1)), \\ f_{\emptyset, \epsilon_2}(x_0, x_1, x_2) &= g_{0,2}(x_0, g_{0,2}(x_1, x_2)), \\ f_{\emptyset, \epsilon_{n+1}}(x_0, \dots, x_{n+1}) &= g_{0,2}(x_0, f_{\emptyset, \epsilon_n}(x_1, \dots, x_{n+1})), \\ f_{\emptyset, \zeta_2}(x_0, x_1, x_2) &= g_{0,2}(x_0, g_{0,2}(x_1, s_{0,2}(x_2, x_1))), \\ f_{\emptyset, \zeta_{n+1}}(x_0, \dots, x_{n+1}) &= g_{0,2}(x_0, f_{\emptyset, \zeta_n}(x_1, \dots, x_{n+1})), \\ f_{\emptyset, \psi(\emptyset)}(x_0, x_1, \dots, x_n) &= x_0, \\ f_{B, \psi(B)}(x_0, x_1, \dots, x_n) &= \bigvee_{i \in B} s_{0,2}(x_0, x_i), \end{aligned}$$

where by $\bigvee$ we mean the semilattice operation s_2 .

It remains to show how to combine such operations. For two relations $\rho_1, \rho_2 \subseteq A^n$, the following equality holds

$$f_{\emptyset, \rho_1 \cap \rho_2}(x_0, x_1, \dots, x_n) = s_2(f_{\emptyset, \rho_1}(x_0, x_1, \dots, x_n), f_{\emptyset, \rho_2}(x_0, x_1, \dots, x_n)).$$

Note that adding dummy variables to a relation ρ is equivalent to adding dummy variables to the operation $f_{\emptyset, \rho}$.

To finish the proof it is sufficient to show that $f_{B, \rho} \in \text{Clo}(\{g_{0,2}, s_{0,2}\})$ for any $B \subseteq \{1, 2, \dots, n\}$ and $\rho \subseteq A^n$ such that $\text{pr}_B(\rho) \subseteq \{0, 1\}^{|B|}$ and ρ is a conjunction of relations from Δ . Since for every $\delta \in \Delta$ we showed how to generate $f_{\emptyset, \delta}$ and how to define conjunction (intersection), $f_{\emptyset, \rho}$ can be generated from $g_{0,2}$ and $s_{0,2}$. Then

$$f_{B, \rho}(x_0, x_1, \dots, x_n) = s_2(f_{\emptyset, \rho}(x_0, x_1, \dots, x_n), f_{B, \psi(B)}(x_0, x_1, \dots, x_n)).$$

□

Theorem 29. $\text{Pol}(\Gamma) = \mathcal{C} = \text{Clo}(\{g_{0,2}, s_{0,2}\})$.

Proof. The claim follows from the following inclusions

$$\text{Pol}(\Gamma) \subseteq \mathcal{C} \subseteq \text{Clo}(\{g_{0,2}, s_{0,2}\}) \subseteq \text{Pol}(\Gamma).$$

The first inclusion is by Lemma 27, the next inclusion is by Lemma 28, and the last one follows from the fact that $g_{0,2}$ and $s_{0,2}$ preserve $R_{\text{and},2}$ and δ . □

Below we present an algorithm that solves $\text{QCSP}^2(\Gamma)$ in polynomial time (see the pseudocode). For an input $\forall x_1 \dots \forall x_n \exists y_1 \dots \exists y_s \Phi$, the function SOLVE_1 first checks whether Φ holds on $\mathbf{x} = (0, \dots, 0)$, $\mathbf{x} = (1, \dots, 1)$, and on each tuple containing exactly one 1. Then for every variable y_j and every variable x_i it calculates the set $D_{i,j}$ of possible values for y_j when $\mathbf{x} = 1^{i-1}01^{n-i}$. Finally, it checks another s tuples determined by $D_{i,j}$. Note that it would be an exponential algorithm if it just checks all possible $\mathbf{x}$. Moreover, since the relation ρ omitting exactly one tuple is pp-definable over Γ , the tuples we need to check could not be found without looking into the formula Φ . 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.

```

1: function SOLVE1( $\Theta$ )
2:   Input: QCSP2( $\Gamma$ ) instance  $\Theta = \forall x_1 \dots \forall x_n \exists y_1 \dots \exists y_s \Phi$ .
3:   if  $\neg \text{SOLVECSP}(\mathbf{x} = (0, \dots, 0) \wedge \Phi)$  then return false  $\triangleright \mathbf{x} = (x_1, \dots, x_n)$ 
4:   if  $\neg \text{SOLVECSP}(\mathbf{x} = (1, \dots, 1) \wedge \Phi)$  then return false
5:   for  $i := 1, \dots, n$  do
6:     if  $\neg \text{SOLVECSP}(\mathbf{x} = (\underbrace{0, \dots, 0}_{i-1}, 1, 0, \dots, 0) \wedge \Phi)$  then return false

7:   for  $j := 1, \dots, s$  do
8:     for  $i := 1, \dots, n$  do
9:        $D_{i,j} := \emptyset$ 
10:       $\mathbf{c} := (\underbrace{1, \dots, 1}_{i-1}, 0, 1, \dots, 1)$ 
11:      for  $a \in A$  do
12:        if  $\text{SOLVECSP}(\mathbf{x} = \mathbf{c} \wedge y_j = a \wedge \Phi)$  then
13:           $D_{i,j} := D_{i,j} \cup \{a\}$ 
14:        if  $D_{i,j} = \emptyset$  then return false
15:      if  $\neg \text{SOLVECSP}(\mathbf{x} = (h(D_{1,j}), \dots, h(D_{n,j})) \wedge \Phi)$  then return false
return true

```

Lemma 30. *Function SOLVE₁ solves QCSP²(Γ) 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 $(x_1, \dots, x_n)$ such that the corresponding CSP has no solutions, which means that the answer is correct.

Assume that the answer is true. Let $R(x_1, \dots, x_n)$ be defined by the formula $\exists y_1 \dots \exists y_s \Phi$. 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 $\{0, 1\}^n \not\subseteq R$. 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 . We checked that R contains $(0, 0, \dots, 0)$ and all tuples with exactly one 1, hence α contains at least two 1s. 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 $R' = \Phi(x_1, \dots, x_n, y)$. Since $D_{i,j} \neq \emptyset$ for all i, j , $\alpha_i \in R$ for every i , and for every $i > k$ we have $\alpha_i d \in R' \Rightarrow d = 1$. Assume that $y = x_i$ for some i . If $\alpha(i) = 0$ then $\alpha_i > \alpha$ and $\alpha_i d \in R' \Rightarrow d = 1$ contradicts $y = x_i$. If $\alpha(i) = 1$, then there is a tuple $\alpha' < \alpha$ such that $\alpha'(i) = 1$, which contradicts $\alpha' d \in R' \Rightarrow d = 0$. Hence $y \neq x_i$ and $y = y_j$ for some j .

Let $\beta = 0 1^{k-1} 0^{n-k}$. Since $\beta < \alpha$, we have $\beta 0 \in R'$. Assume that $D_{1,j}$ 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 $i = 1, 2, \dots, n-k$. Since $g_{0,2}(1, 2) = 2$, we have $\gamma_i 2 \in R'$ for every i . Note that $\gamma_{n-k} \in \{0, 1\}^n$. We can check that $g_{0,2}(\alpha_1 1, \gamma_{n-k} 2) = \alpha_1 2$, which contradicts the fact that $D_{1,j} = \{1\}$. In this way we can show that $D_{1,j}, \dots, D_{k,j}$ are not equal to $\{1\}$. We also know that $D_{k+1,j}, \dots, D_{n,j}$ are equal to $\{1\}$. Hence, the tuple $(h(D_{1,j}), \dots, h(D_{n,j})) = \alpha$ 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 $3 \cdot s \cdot n + s + n + 2$ CSP instances over a language preserved by the semilattice operation s_2 . $\square$

Corollary 31. *QCSP(Γ) is in P .*

Proof. Since $f(x, y, z) = s_{0,2}(x, y)$ is a 01-stable operation, by Lemma 18 $\text{QCSP}(\Gamma)$ can be polynomially reduced to $\text{QCSP}^2(\Gamma)$, and $\text{QCSP}^2(\Gamma)$ can be solved by the function SOLVE_1 . $\square$

Corollary 32. $\text{QCSP}(\Gamma_0)$ is in P for every finite $\Gamma_0 \subseteq \text{Inv}(\{g_{0,2}, s_{0,2}\})$.

Proof. By Theorem 29, $\text{Pol}(\Gamma) = \text{Clo}(\{g_{0,2}, s_{0,2}\})$. Then $\text{Pol}(\Gamma_0) \supseteq \text{Pol}(\Gamma)$, which implies that each relation from Γ_0 can be pp-defined from Γ . Therefore, $\text{QCSP}(\Gamma_0)$ can be polynomially reduced to $\text{QCSP}(\Gamma)$ (we just replace every relation by its pp-definition). Hence, by Corollary 31, $\text{QCSP}(\Gamma_0)$ is in P . $\square$

8 Strange structure 2

In this section we show that $\text{QCSP}(\Gamma)$ is solvable in polynomial time if $\Gamma \subseteq \text{Inv}(f_{0,2})$. 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 $\Gamma \subseteq \text{Inv}(f_{0,2})$. For a variable y of Φ we define a partial operation $F_y(x_1, \dots, x_n)$ on A 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 $<$ or $\leq$ we mean that both tuples are on $\{0, 1\}$).

The following lemma proves that F_y is monotonic.

Lemma 33. Suppose $\alpha \leq \beta$, $F_y(\alpha)$ and $F_y(\beta)$ are defined. Then $F_y(\alpha) \leq F_y(\beta)$.

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. $\square$

Lemma 34. 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 . $\square$

Lemma 35. 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 $\text{pr}_S C$, 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 $\text{Inv}(f_{0,2})$.

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 operation 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, $c_i \leq a_i$ 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 $k \geq 1$. For each $i \in \{1, 2, \dots, k\}$ 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 $\alpha_i \leq \beta$ 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 $i \in \{1, 2, \dots, k\}$. Therefore, $(z_1, \dots, z_l) = (a_1, \dots, a_l)$ in this solution, which means that $(a_1, \dots, a_l) \in \rho$ and contradicts the assumption.

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

Lemma 36. *Suppose α is a minimal 1-set for y , $i \in \{1, 2, \dots, n\}$, 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 $(x_1, \dots, x_n) = 2^{i-1}02^{n-i}$ 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 $(x_1, \dots, x_n) = 2^{i-1}02^{n-i}$ 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 $(x_1, \dots, x_n) = \alpha$ and $y = 2$, which contradicts the definition of a minimal 1-set. $\square$

Example. Let $R'_{and,2} = R_{and,2} \setminus \{(0, 2, 1), (0, 2, 2), (2, 0, 1), (2, 0, 2)\}$ (see Section 7), $\delta' = \{(0, 1)\} \cup (\{1, 2\} \times \{0, 1, 2\})$, $\Gamma' = \{R'_{and,2}, \delta', \{0\}, \{1\}, \{2\}\}$. We can check that Γ' is preserved by $f_{0,2}$. It follows from the following lemma that $\text{Pol}(\Gamma')$ has the EGP property.

As in the previous section here we use the notations τ_n and σ_n for $\alpha = \{0, 2\}$ and $\beta = \{1, 2\}$.

Lemma 37. *σ_n can be pp-defined from $\{R'_{and,2}, \delta', \{0\}\}$.*

Proof. Recursively we define

$$\begin{aligned} R'_{and,n+1}(x_1, \dots, x_n, x_{n+1}, y) &= \exists z R'_{and,n}(x_1, \dots, x_n, z) \wedge R'_{and,2}(x_{n+1}, z, y), \\ \omega_n(x_1, y_1, x_2, y_2, \dots, x_n, y_n) &= \exists u_1 \dots \exists u_n \exists z R'_{and,2n}(x_1, \dots, x_n, u_1, \dots, u_n, z) \wedge \\ &\quad \delta'(y_1, u_1) \wedge \dots \wedge \delta'(y_n, u_n) \wedge z = 0. \end{aligned}$$

It is not hard to see that 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. $\square$

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

Proof. Suppose $\sigma_n = \Phi(x_1, \dots, x_n)$, where Φ is a conjunctive formula over Γ . There exist 2^n tuples from $A^{2^n} \setminus \sigma_n$ and each of them has at least two 1s. By Lemma 35, for each $\alpha \in A^{2^n} \setminus \sigma_n$ there should be a constraint $\rho(z_1, \dots, z_l)$ such that $\alpha = \bigvee_{i \in B} \alpha_i$ for some $B \subseteq \{1, 2, \dots, l\}$. 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 $A^{2^n} \setminus \sigma_n$. Thus, Φ has at least $2^n/2^k$ constraints.

Similarly, if V is the set of all variables in Φ , then the above argument gives an injection from the set $A^{2^n} \setminus \sigma_n$ of size 2^n to the set of all subsets of V of size at most k . This implies that $|V|^k \geq 2^n$ and $|V| \geq 2^{n-k}$. $\square$

Thus, for a fixed Γ we need exponentially many constraints to define σ_n .

Corollary 39. *Suppose $\Gamma \subseteq \text{Inv}(f_{0,2})$, all relations in Γ are of arity at most k . Then any pp-definition of τ_n over Γ , where $n \geq 2$, has at least $2^n/2^k$ constraints and at least $2^n/2^k$ variables.*

Below we present an algorithm that solves $\text{QCSP}^2(\Gamma)$ in polynomial time for $\Gamma \subseteq \text{Inv}(f_{0,2})$ (see the pseudocode of the function SOLVE_2). For an input $\forall x_1 \dots \forall x_n \exists y_1 \dots \exists y_s \Phi$, the function SOLVE_2 first checks whether Φ holds on $\mathbf{x} = (0, \dots, 0)$ and on each tuple containing exactly one 1. Then for every variable y_j it calculates the minimal 1-set using Lemma 36. Precisely, for every variable y_j it calculates the set $D_{i,j}$ of possible values for y_j when $\mathbf{x} = 2^{i-1}02^{n-i}$, and sets the i -th element of the 1-set α_j to 1 whenever $D_{i,j} = \{0\}$. Note that the minimal 1-set for a variable x_i is $0^{i-1}10^{n-i}$. Then, for each constraint $\rho(z_1, \dots, z_l)$ and each subset V of its variables the algorithm calculates the disjunction of the minimal 1-sets for the variables from V , and checks that Φ holds on the result. Again, it would be an exponential algorithm if it just checks all possible $\mathbf{x}$. By 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 40. *Function SOLVE_2 solves $\text{QCSP}^2(\Gamma)$ in polynomial time for a finite constraint language $\Gamma \subseteq \text{Inv}(f_{0,2})$.*

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

Assume that the answer is true. Let $R(x_1, \dots, x_n)$ be defined by the formula $\exists y_1 \dots \exists y_s \Phi$. 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 $\{0, 1\}^n \not\subseteq R$. Then let α be a minimal tuple from $\{0, 1\}^n \setminus R$. Since we checked that $(0, 0, \dots, 0)$ and all tuples having just one 1 are from R , α contains at least two 1. Then, by Lemma 35, there should be a constraint $\rho(z_1, \dots, z_l)$ of Φ and a subset $V \subseteq \{1, 2, \dots, l\}$ such that α is a disjunction of the minimal

```

1: function SOLVE2( $\Theta$ )
2:   Input: QCSP2( $\Gamma$ ) instance  $\Theta = \forall x_1 \dots \forall x_n \exists y_1 \dots \exists y_s \Phi$ .
3:   if  $\neg \text{SOLVECSP}(\mathbf{x} = (0, \dots, 0) \wedge \Phi)$  then return false  $\triangleright \mathbf{x} = (x_1, \dots, x_n)$ 
4:   for  $i := 1, \dots, n$  do  $\triangleright$  Check all tuples with just one 1
5:      $\mathbf{c} := (\underbrace{0, \dots, 0}_{i-1}, 1, 0, \dots, 0)$ 
6:     if  $\neg \text{SOLVECSP}(\mathbf{x} = \mathbf{c} \wedge \Phi)$  then return false
7:   for  $j := 1, \dots, s$  do  $\triangleright$  Calculate the minimal 1-set for every  $y_j$ 
8:      $\alpha_j := (0, \dots, 0)$ 
9:     for  $i := 1, \dots, n$  do
10:       $D_{i,j} := \emptyset$ 
11:       $\mathbf{c} := (\underbrace{2, \dots, 2}_{i-1}, 0, 2, \dots, 2)$ 
12:      for  $a \in A$  do
13:        if  $\text{SOLVECSP}(\mathbf{x} = \mathbf{c} \wedge y_j = a \wedge \Phi)$  then
14:           $D_{i,j} := D_{i,j} \cup \{a\}$ 
15:      if  $D_{i,j} = \emptyset$  then return false
16:      if  $D_{i,j} = \{0\}$  then
17:         $\alpha_j := \alpha_j \vee (\underbrace{0, \dots, 0}_{i-1}, 1, 0, \dots, 0)$ 
18:   for a constraint  $\rho(z_1, \dots, z_l)$  of  $\Phi$  do  $\triangleright$  Check all constraints
19:     for  $V \subseteq \{1, 2, \dots, l\}$  do  $\triangleright$  Check all subsets of variables
20:        $\beta := (0, \dots, 0)$ 
21:       for  $j \in V$  do
22:         if  $z_j = x_i$  for some  $i$  then
23:            $\beta := \beta \vee (\underbrace{0, \dots, 0}_{i-1}, 1, 0, \dots, 0)$   $\triangleright$  Add the minimal 1-set for  $x_i$ 
24:         if  $z_j = y_i$  for some  $i$  then
25:            $\beta := \beta \vee \alpha_i$   $\triangleright$  Add the minimal 1-set for  $y_i$ 
26:       if  $\neg \text{SOLVECSP}(\mathbf{x} = \beta \wedge \Phi)$  then return false
return true

```

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 36, if α_j is 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. Here we use the fact that the minimal 1-set for x_i is $0^{i-1}10^{n-i}$. In line 26 we check whether Φ has a solution with $(x_1, \dots, x_n) = \alpha$. Thus, Lemma 35 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. $\square$

Corollary 41. *QCSP(Γ) is in P for every finite $\Gamma \subseteq \text{Inv}(f_{0,2})$.*

Proof. Since $f(x, y, z) = s_{0,2}(x, y)$ is a 01-stable operation, by Lemma 18 QCSP(Γ) can be polynomially reduced to QCSP²(Γ), and QCSP²(Γ) can be solved by the function SOLVE₂. $\square$

9 EGP and WNU on 3-element domain

In this section we consider constraint languages on $\{0, 1, 2\}$ with constants for which $\text{Pol}(\Gamma)$ has the EGP property and contains a WNU operation. It can be shown (see the proof of Theorem 6) that in this case $\text{Pol}(\Gamma)$ contains a semilattice operation. Since adding pp-definable relations to Γ does not change the complexity of QCSP(Γ), we may assume that k is the maximal arity of the relations in Γ and Γ contains all pp-definable relations of arity at most k . Also, EGP implies that there exists α and β , neither equal to A but so that $\alpha \cup \beta = A$, so that all operations of $\text{Pol}(\Gamma)$ are $\alpha\beta$ -projective [31]. If $\alpha \cap \beta = \emptyset$, then there can be no WNU. Hence, we may assume that $\alpha = \{0, 2\}$ and $\beta = \{1, 2\}$, which implies that $\{0, 2\}$, $\{1, 2\}$, and $\{0, 2\}^2 \cup \{1, 2\}^2$ are in Γ . Thus, in this section we have the following assumptions:

1. Γ is a finite constraint language on $A = \{0, 1, 2\}$
2. Γ contains the relations $\{0\}, \{1\}, \{2\}, \{0, 2\}, \{1, 2\}, \{0, 2\}^2 \cup \{1, 2\}^2$, and the equality relation
3. Γ is preserved by the semilattice operation s_2
4. $\text{Pol}(\Gamma)$ is $\{0, 2\}\{1, 2\}$ -projective
5. all relations in Γ are of arity at most $k \geq 4$
6. Γ contains all relations of arity at most k that can be pp-defined from Γ

These assumptions will hold up to, but not including, Theorem 60. Similar to a 01-stable operation, we define 0-stable and 1-stable operations, where f is called *c-stable* if it is idempotent, $f(x, c) = x$, and $f(x, 2) = 2$. Note that by adding a dummy variable we can make it a 01-stable operation.

9.1 Finiteness of the language

In this subsection we will derive some properties of Γ based on the facts that Γ is finite and $\text{Pol}(\Gamma)$ is $\{0, 2\}\{1, 2\}$ -projective.

For $B, C \subseteq A$ and $n \geq 3$ by $\text{nu}_{B,C}^n$ we denote the operation of arity n defined on a tuple $(a_1, \dots, a_n)$ as follows. If $a_1 = \dots = a_n$ then it returns a_1 . If $a_1 = \dots = a_{i-1} = a_{i+1} = \dots = a_n \in B$ and $a_i \in C$ for some i , then it returns a_j for $j \neq i$. Otherwise, it returns 2.

Lemma 42. *Suppose ρ is a relation of the minimal arity N in Γ that is not preserved by $\text{nu}_{B,C}^{2k}$. Then, after some permutation of variables, ρ satisfies one of the following conditions:*

1. *There exist $\alpha_1, \alpha_2 \in \rho$ such that $\text{nu}_{B,C}^{2k}(\alpha_1, \alpha_2, \alpha_2, \dots, \alpha_2) = \beta \notin \rho$, $\alpha_2(i) \in B$ and $\beta(i) = \alpha_2(i) \neq 2$ for $i \in \{1, \dots, N-1\}$, $\beta(N) = 2$, and $\alpha_1(i) \neq \alpha_2(i)$ for every i . Moreover, if $C = A$ then $N = 2$, $\alpha_2(N) \notin B$, and $\alpha_1 = (2, 2)$.*
2. *There exist $\alpha_1, \alpha_2, \alpha_3 \in \rho$ such that $\text{nu}_{B,C}^{2k}(\alpha_1, \alpha_2, \alpha_3, \alpha_3, \dots, \alpha_3) = \beta \notin \rho$, $\beta(i) = \alpha_3(i) \neq 2$ for $i \in \{1, \dots, N-1\}$, $\beta(N) = 2$, $\alpha_1(N), \alpha_2(N) \in C$, $\alpha_3(i) \in B$ for every i , and $\alpha_1(i) \neq \alpha_3(i)$ or $\alpha_2(i) \neq \alpha_3(i)$ for every i . Moreover, if $2 \in C$, then $\alpha_1(i), \alpha_2(i) \in \{\alpha_3(i), 2\}$ for every i , $\alpha_1(N) = \alpha_2(N) = 2$, and $N = 3$.*

Proof. Note that the hypotheses imply that $N \geq 2$. Consider tuples $\alpha_1, \dots, \alpha_{2k} \in \rho$ such that $\text{nu}_{B,C}^{2k}(\alpha_1, \dots, \alpha_{2k}) = \beta \notin \rho$. Since ρ is of the minimal arity, $\text{nu}_{B,C}^{2k}$ preserves any proper projection of ρ . For example, $\text{nu}_{B,C}^{2k}$ preserves $\text{pr}_{1, \dots, N-1}(\rho)$, therefore

$$\text{nu}_{B,C}^{2k}(\text{pr}_{1, \dots, N-1}(\alpha_1), \dots, \text{pr}_{1, \dots, N-1}(\alpha_{2k})) \in \text{pr}_{1, \dots, N-1}(\rho).$$

Hence, there exists a tuple $\beta_N \in \rho$ that differs from β only in the N -th element. Similarly, for every i there exists a tuple $\beta_i \in \rho$ that differs from β only in the i -th element.

Let us consider the matrix M whose columns are $\alpha_1, \dots, \alpha_{2k}$. Assume that for some j we have $\alpha_1(j) = \dots = \alpha_{2k}(j)$. Then if we substitute constant $\alpha_1(j)$ for the j -th variable of ρ we get a relation of smaller arity that is not preserved by $\text{nu}_{B,C}^{2k}$, which contradicts our assumptions. Therefore in any row of the matrix there should be elements that are not equal. Also, all rows of the matrix should be different, since otherwise we can identify the corresponding variables of ρ to obtain a relation of smaller arity that is not preserved by $\text{nu}_{B,C}^{2k}$.

Assume that β has at least two elements equal to 2 and they appear at the i -th and j -th positions. Then $s_2(\beta_i, \beta_j) = \beta$, which contradicts the fact that s_2 preserves ρ . Assume that β has no 2. Every row of the matrix must have exactly one element different from all other elements, and the result of applying $\text{nu}_{B,C}^{2k}$ to every row is the most popular element of the row. Since we have $N \leq k$ rows and $2k$ columns, there should be a column among $\alpha_1, \dots, \alpha_{2k}$ equal to β , which contradicts our assumptions. W.l.o.g., we assume that $\beta(N) = 2$ (otherwise we just permute variables of ρ).

Put $\alpha'_i = \text{pr}_{1, \dots, N-1}(\alpha_i)$ for every i , $\beta' = \text{pr}_{1, \dots, N-1}(\beta)$. We know that for every $i \in \{1, \dots, N-1\}$ the i -th row contains exactly one element different from the others. Since $N \leq k$, at least $k+1$ of the tuples $\alpha'_1, \dots, \alpha'_{2k}$ are equal to β' . W.l.o.g., we assume that $\alpha'_{s+1} = \dots = \alpha'_{2k} = \beta'$, $\alpha'_i \neq \beta'$ for every $i \in \{1, \dots, s\}$, where $s < N$. Since not all elements of the first row of the matrix are equal, we have $s \geq 1$.

Replace the first column of the matrix M by α_{2k} and consider $\text{nu}_{B,C}^{2k}(\alpha_{2k}, \alpha_2, \alpha_3, \dots, \alpha_{2k}) = \gamma_1$. Since $\alpha'_1 \neq \beta'$, there should be a row in the new matrix whose elements are equal. Therefore, $\gamma_1 \in \rho$, otherwise we consider $\alpha_{2k}, \alpha_2, \alpha_3, \dots, \alpha_{2k}$ instead of the original sequence, and get a contradiction with the fact that every row should contain at least two different elements. By the definition $\text{pr}_{1, \dots, N-1}(\gamma_1) = \beta'$, therefore $\gamma_1(N) \neq 2$. This means that at least $2k-2$ among the elements $\alpha_2(N), \dots, \alpha_{2k}(N)$ are equal to $\alpha_{2k}(N)$. In the same way we can

substitute α_j instead of α_i for any $i \in \{1, \dots, s\}$ and $j \in \{s+1, \dots, 2k\}$ to conclude that $2k-2$ elements among the elements $\alpha_1(N), \dots, \alpha_{i-1}(N), \alpha_{i+1}(N), \dots, \alpha_{2k}(N)$ are equal to $\alpha_j(N)$. Since $\beta(N) = 2$, we also have $\alpha_i(N) \neq \alpha_j(N)$. This implies that $\alpha_{s+1}(N) = \dots = \alpha_{2k}(N)$, and one of the following cases holds:

1. $s = 1, \alpha_2 = \dots = \alpha_{2k}$;
2. $s = 2, \alpha_1(N), \alpha_2(N) \in C, \alpha_3 = \dots = \alpha_{2k}, \alpha_3(N) \in B$.

If $s = 1$ then $\alpha_1(i) \neq \alpha_2(i)$ for every i since there should be different elements in every row. Suppose $A = C$. Since $s_2(\alpha_1, \alpha_2) = (2, \dots, 2) \in \rho$, we can take $\alpha_1 = (2, \dots, 2)$. Since $\alpha_2(i) \neq \alpha_1(i)$ and all rows are different, we have $N = 2$ and $\alpha(2)(N) \notin B$.

If $s = 2$, then $\alpha_3(i) \in B$ for all $i < N$ since $\beta(i) \neq 2$. It remains to get an extra condition for the case when $2 \in C$ and $s = 2$. In this case we replace α_1 by $\delta_1 = s_2(\beta_1, \beta_N)$ (see the definition of β_1 and β_N above), and α_2 by $\delta_2 = s_2(\beta_2, \beta_N)$. Then $\delta_1(i), \delta_2(i) \in \{\alpha_3(i), 2\}$ for every i , and $nu_{B,C}^{2k}(\delta_1, \delta_2, \alpha_3, \dots, \alpha_3) = \beta$. Note that δ_1 can differ from α_3 in at most 2 coordinates and one of them is the N -th coordinate. The same is true for δ_2 . Therefore, since $\delta_1, \delta_2, \alpha_3$ (viewed as a matrix with 3 columns) cannot have equal rows, N should be equal to 3. $\square$

Our plan is to apply the above lemma to define relations that will be used to prove hardness results.

Lemma 43. *One of the following cases holds.*

1. $(\{0, 2\} \times \{0, 2\} \times \{0, 2\}) \setminus \{(0, 0, 2)\} \in \Gamma$,
2. $(\{1, 2\} \times \{1, 2\} \times \{1, 2\}) \setminus \{(1, 1, 2)\} \in \Gamma$,
3. $(\{0, 2\} \times \{0, 2\} \times \{1, 2\}) \setminus \{(0, 0, 2)\} \in \Gamma$,
4. $(\{1, 2\} \times \{1, 2\} \times \{0, 2\}) \setminus \{(0, 0, 2)\} \in \Gamma$.

Proof. Since $nu_{A,A}^{2k}$ is not $\{0, 2\}\{1, 2\}$ -projective, Γ is not preserved by $nu_{A,A}^{2k}$. By Lemma 42 there exists a relation $\rho \in \Gamma$ satisfying one of the two cases. The first case is not possible because $nu_{A,A}^{2k}(\alpha_1, \alpha_2, \alpha_2, \dots, \alpha_2) = \alpha_2$ for every α_1, α_2 . Thus, ρ is of arity 3 and there exist $\alpha_1, \alpha_2, \alpha_3 \in \rho$ such that $nu_{A,A}^{2k}(\alpha_1, \alpha_2, \alpha_3, \alpha_3, \dots, \alpha_3) = \beta \notin \rho$, $\beta(i) = \alpha_3(i) \neq 2$ for $i \in \{1, 2\}$, $\beta(3) = 2$, $\alpha_1(i) \neq \alpha_3(i)$ or $\alpha_2(i) \neq \alpha_3(i)$, $\alpha_1(i), \alpha_2(i) \in \{\alpha_3(i), 2\}$ for every i , $\alpha_1(3) = \alpha_2(3) = 2$. Suppose $\alpha_3 = (a_1, a_2, a_3)$. We know that $(a_1, a_2, a_3), (2, a_2, 2), (a_1, 2, 2) \in \rho$ (as $\alpha_3, \alpha_1, \alpha_2$) and $(a_1, a_2, 2) \notin \rho$ (as β). Since the semilattice s_2 applied to $(a_1, a_2, 0)$ and $(a_1, a_2, 1)$ gives $(a_1, a_2, 2) \notin \rho$, $(a_1, a_2, c) \in \rho$ implies $c = a_3$.

Since $a_1, a_2, a_3 \in \{0, 1\}$, at least two of them are equal. We consider three cases:

Case 1. $a_2 = a_3$. Put $\delta(x, y, z) = \exists t \rho(x, a_2, t) \wedge \rho(y, t, z)$. It is not hard to see that $(a_1, a_1, 2) \notin \delta$ (we need to put $t = a_2 = a_3$ and, therefore, $z = a_3$), $(a_1, 2, 2), (a_1, a_1, a_3) \in \delta$ (put $t = a_2 = a_3$), $(2, 2, 2), (2, a_1, 2) \in \delta$ (put $t = 2$).

Case 2. $a_1 = a_3$. Put $\delta(x, y, z) = \exists t \rho(a_1, x, t) \wedge \rho(t, y, z)$. It is not hard to see that $(a_2, a_2, 2) \notin \delta$ (we need to put $t = a_1 = a_3$ and, therefore, $z = a_3$), $(a_2, 2, 2), (a_2, a_2, a_3) \in \delta$ (put $t = a_1 = a_3$), $(2, 2, 2), (2, a_2, 2) \in \delta$ (put $t = 2$).

Case 3. $a_1 = a_2$. In this case we put $\delta = \rho$.

Thus, we have a relation $\delta \in \Gamma$ satisfying the following properties: for some $a, b \in \{0, 1\}$ we have $(a, a, b), (2, a, 2), (a, 2, 2), (2, 2, 2) \in \delta$, $(a, a, 2) \notin \delta$. Since s_2 preserves δ , $(a, a, c) \in \delta$ implies $c = b$.

Put $\delta_1 = \delta \cap (\{a, 2\} \times \{a, 2\} \times \{b, 2\})$, $\delta_2(x, y, z) = \exists t \delta_1(x, y, t) \wedge \delta_1(x, z, t) \wedge \delta_1(z, x, t)$.

Assume that $(a, a, 2) \in \delta_2$. Since we need to set $t = b$, we have $(a, 2, b), (2, a, b) \in \delta_1$. Since s_2 preserves δ_1 we obtain $(2, 2, b) \in \delta_1$, which means that $\delta_1 = (\{a, 2\} \times \{a, 2\} \times \{b, 2\}) \setminus \{(a, a, 2)\}$ and completes this case.

Assume that $(a, a, 2) \notin \delta_2$. We know that $(a, a, a) \in \delta_2$ (put $t = b$), $(a, 2, 2), (2, a, 2), (2, 2, a), (2, 2, 2) \in \delta_2$ (put $t = 2$). Put

$$\delta_3(x, y, z) = \exists x' \exists y' \exists z' \delta_2(x, x, x') \wedge \delta_2(y, y, y') \wedge \delta_2(z', z', z) \wedge \delta_2(x', y', z').$$

Let us show that $\delta_3 = (\{a, 2\} \times \{a, 2\} \times \{a, 2\}) \setminus \{(a, a, 2)\}$. If $x = 2$, then we put $x' = z' = 2$ and $y' = y$. If $y = 2$ then we put $y' = z' = 2$ and $x' = x$. If $x = y = z = a$, then we put $y' = z' = y' = a$. It remains to show that $(a, a, 2) \notin \delta_3$, which follows from the fact that x' should be equal to a , y' should be equal to a , z' should be equal to 2 , $(a, a, 2) \notin \delta_2$. Thus, $\delta_3 = (\{a, 2\} \times \{a, 2\} \times \{a, 2\}) \setminus \{(a, a, 2)\}$, which completes the proof. $\square$

Lemma 44. *One of the following conditions holds.*

1. $(\{0, 2\} \times \{0, 2\} \times \{0, 2\}) \setminus \{(0, 0, 2)\} \in \Gamma$;
2. *there exists a relation $\delta \in \Gamma$ such that $(0, 1), (2, 2) \in \delta$, $(0, 2) \notin \delta$;*

Proof. Since $nu_{\{0\}, A}^{2k}$ is not $\{0, 2\}\{1, 2\}$ -projective, Γ is not preserved by $nu_{\{0\}, A}^{2k}$. By Lemma 42 there exists a relation $\rho \in \Gamma$ satisfying one of the two cases of Lemma 42. Suppose we have the second case of Lemma 42, where we import that lemma's notation. Then $\alpha_3 = (0, 0, 0), \alpha_1 = (2, 0, 2), \alpha_2 = (0, 2, 2), \beta = (0, 0, 2)$ (or α_1 and α_2 can be switched). Since s_2 preserves ρ , we have $(2, 2, 2) \in \rho$.

Let $\delta_0 = \rho \cap \{0, 2\}^3$. Consider two cases:

Case 1. $(2, 2, 0) \in \delta_0$. For $\epsilon(x, y) = \delta_0(y, y, x)$ we have $\epsilon = \begin{pmatrix} 0 & 0 & 2 \\ 0 & 2 & 2 \end{pmatrix}$.

Case 2. $(2, 2, 0) \notin \delta_0$. Put $\epsilon(x, y) = \exists t \delta_0(t, x, y) \wedge \delta_0(x, t, y)$. If $(2, 0) \in \epsilon$, then $(0, 2, 0), (2, 0, 0) \in \delta_0$, which, using the semilattice s_2 , implies $(2, 2, 0) \in \delta_0$. This contradiction shows that $(2, 0) \notin \epsilon$ and $\epsilon = \begin{pmatrix} 0 & 0 & 2 \\ 0 & 2 & 2 \end{pmatrix}$.

Then the required relation $(\{0, 2\} \times \{0, 2\} \times \{0, 2\}) \setminus \{(0, 0, 2)\}$ can be pp-defined by

$$\delta_1(x, y, z) = \exists x' \exists y' \exists z' \epsilon(x', x) \wedge \epsilon(y', y) \wedge \epsilon(z, z') \wedge \delta_0(x', y', z').$$

Suppose we have the first case in Lemma 42. Then $\beta = (0, 2), \alpha_1 = (2, 2), \alpha_2 = (0, 1), s_2(\alpha_1, \alpha_2) = (2, 2) \in \rho$. Thus, ρ satisfies the conditions of item 2. $\square$

Similarly, if we switch 0 and 1 we get the following lemma.

Lemma 45. *One of the following cases holds.*

1. $(\{1, 2\} \times \{1, 2\} \times \{1, 2\}) \setminus \{(1, 1, 2)\} \in \Gamma$
2. *there exists $\delta \in \Gamma$ such that $(1, 0), (2, 2) \in \delta$, $(1, 2) \notin \delta$*

Lemma 46. *One of the following cases holds:*

1. $(\{0, 2\} \times \{0, 2\} \times \{0, 2\}) \setminus \{(0, 0, 2)\}, (\{1, 2\} \times \{1, 2\} \times \{1, 2\}) \setminus \{(1, 1, 2)\} \in \Gamma$
2. $(\{1, 2\} \times \{0, 2\} \times \{0, 2\}) \setminus \{(1, 0, 2)\}, (\{0, 2\} \times \{0, 2\} \times \{0, 2\}) \setminus \{(0, 0, 2)\} \in \Gamma$
3. $(\{0, 2\} \times \{1, 2\} \times \{1, 2\}) \setminus \{(0, 1, 2)\}, (\{1, 2\} \times \{1, 2\} \times \{1, 2\}) \setminus \{(1, 1, 2)\} \in \Gamma$

Proof. We apply Lemmas 44 and 45. If we have the first case in both of them, then the first condition holds and we are done.

Assume that we have the second case in both of them and the corresponding relations are δ_0 and δ_1 , respectively. Put

$$\delta(x, y) = \delta_0(x, y) \wedge \delta_1(y, x) \wedge (x \in \{0, 2\}) \wedge (y \in \{1, 2\}).$$

It is not hard to see that $\delta = \begin{pmatrix} 0 & 2 \\ 1 & 2 \end{pmatrix}$. Using this relation we can easily switch 0 and 1 in relations. By Lemma 43 we have $\rho = (\{a, 2\} \times \{a, 2\} \times \{b, 2\}) \setminus \{(a, a, 2)\} \in \Gamma$ for some $a, b \in \{0, 1\}$. If $a = b = 1$, then we use the relation ρ and the relation defined by the following pp-definition to satisfy condition 1:

$$\rho_1(x, y, z) = \exists x' \exists y' \exists z' \rho(x', y', z') \wedge \delta(x, x') \wedge \delta(y, y') \wedge \delta(z, z').$$

If $a = b = 0$ then the same pp-definition, but with the arguments of δ switched, applies.

If $a \neq b$, and $(a, b) = (0, 1)$, then the relations we need to satisfy condition 1 can be defined by:

$$\rho_2(x, y, z) = \exists z' \rho(x, y, z') \wedge \delta(z, z'),$$

$$\rho_3(x, y, z) = \exists x' \exists y' \rho(x', y', z) \wedge \delta(x', x) \wedge \delta(y', y).$$

If $(a, b) = (1, 0)$ then the same pp-definitions, but with the arguments of δ switched, applies.

Assume that we have the first case in Lemma 44 and the second case in Lemma 45. Thus, $\rho_4 = (\{0, 2\} \times \{0, 2\} \times \{0, 2\}) \setminus \{(0, 0, 2)\} \in \Gamma$ and there exists $\delta \in \Gamma$ such that $(1, 0), (2, 2) \in \delta$, $(1, 2) \notin \delta$. Put

$$\rho_5(x, y, z) = \exists x' \rho_4(x', y, z) \wedge \delta(x, x') \wedge x \in \{1, 2\} \wedge x' \in \{0, 2\}.$$

Then ρ_4 and ρ_5 satisfy the second condition of this lemma.

Assume that we have the second case in Lemma 44 and the first case in Lemma 45. Thus, $\rho_6 = (\{1, 2\} \times \{1, 2\} \times \{1, 2\}) \setminus \{(1, 1, 2)\} \in \Gamma$ and there exists $\delta \in \Gamma$ such that $(0, 1), (2, 2) \in \delta$, $(0, 2) \notin \delta$. Put

$$\rho_7(x, y, z) = \exists x' \rho_6(x', y, z) \wedge \delta(x, x') \wedge x \in \{0, 2\} \wedge x' \in \{1, 2\}.$$

Then ρ_6 and ρ_7 satisfy the third condition of this lemma. □

9.2 Connection with Boolean functions

A mapping $f : \{0, 1\}^n \rightarrow \{0, 1\}$ is called a *Boolean function*. We say that a Boolean function f of arity n is represented by Γ if there exists a relation R of arity $n + 1$ pp-definable using Γ such that for all $a_1, \dots, a_n \in \{0, 1\}$, $b \in \{0, 1, 2\}$

$$(a_1, \dots, a_n, b) \in R \Leftrightarrow f(a_1, \dots, a_n) = b.$$

For example, an AND-type relation is a representation of conjunction, an OR-type relation is a representation of disjunction. A k -ary Boolean function f is *monotonic* if $f(x_1, \dots, x_k) \leq f(y_1, \dots, y_k)$ whenever $x_i \leq y_i$ for every $i \in \{1, 2, \dots, k\}$. A k -ary Boolean function is *linear* if it is of the form $c_1 x_1 + \dots + c_k x_k + c_0$ for $c_0, c_1, \dots, c_k \in \{0, 1\}$.

Lemma 47. *The set of all Boolean functions represented by Γ is a clone generated by one of the following sets: $\{0, 1\}$, $\{+, 0, 1\}$, $\{\wedge, 0, 1\}$, $\{\vee, 0, 1\}$, $\{\wedge, \vee, 0, 1\}$ or $\{\wedge, \neg, 0, 1\}$.*

Note that this is true for arbitrary constraint languages containing $\{0\}$, $\{1\}$ and $=$.

Proof. Notice that constants 0 and 1 are represented by the singleton relations $\{0\}$ and $\{1\}$, which are in Γ , and a projection can be represented by the equality relation, which is assumed to be in Γ .

Suppose we have Boolean functions $g_1, \dots, g_n$ of arity m represented by relations $R_1, \dots, R_n$, respectively. Suppose f is Boolean function of arity n represented by a relation R . Then the composition $h = f(g_1, \dots, g_n)$ can be represented by the relation defined by

$$R'(x_1, \dots, x_m, z) = \exists y_1 \dots \exists y_n R_1(x_1, \dots, x_m, y_1) \wedge \dots \wedge R_n(x_1, \dots, x_m, y_n) \wedge R(y_1, \dots, y_n, z).$$

Since we are closed under composition and above $\{0, 1\}$ we can read from Post's Lattice that we are one of the clones in the statement of the lemma. $\square$

By $\text{BoolClo}(\Gamma)$ we denote the set of all Boolean functions represented by Γ , which is a clone by Lemma 47.

Let $\alpha_1, \dots, \alpha_N$, where $N = 2^{k+1} - 1$, be the set of all tuples from $\{0, 1\}^{k+1}$ except for the tuple $(1, 1, \dots, 1)$. Thus, α_i can be viewed as a binary representation of the number $(i - 1)$.

Let us define an operation f_Γ on A of arity N . If there exists a unique Boolean function $h \in \text{BoolClo}(\Gamma)$ of arity $k + 1$ such that $h(\alpha_i) = a_i$ for every i , then put $f(a_1, \dots, a_N) = h(1, 1, \dots, 1)$. Otherwise, put $f(a_1, \dots, a_N) = 2$.

Lemma 48. *Suppose Boolean functions h_1 and h_2 of arity $n \geq 2$ differ only on the tuple $(1, 1, \dots, 1)$. Then conjunction can be generated from h_1, h_2 and constants 0, 1.*

Proof. Since the functions h_1 and h_2 differ just on one tuple, they cannot be both linear. Assume that they are monotonic. Since $h_1(1, 1, \dots, 1) \neq h_2(1, 1, \dots, 1)$, we have $h_1(x_1, \dots, x_n) = 0$ and $h_2(x_1, \dots, x_n) = x_1 \wedge \dots \wedge x_n$ (or we should switch h_1 and h_2). Thus, we generated conjunction in this case.

Assume that one of the functions h_1 and h_2 is not monotonic. Then $h_1, h_2, 0, 1$ are not contained in any maximal clone of Boolean functions. Hence, by [27], any function, including conjunction, can be generated from $h_1, h_2, 0$, and 1. $\square$

Therefore, if $\text{BoolClo}(\Gamma)$ does not contain conjunction, then the word “unique” can be removed from the definition of f_Γ . Also, since constants 0 and 1 are in $\text{BoolClo}(\Gamma)$, the operation f_Γ is idempotent in this case.

Lemma 49. *Suppose $\text{BoolClo}(\Gamma)$ does not contain conjunction, then f_Γ is not $\{0, 2\}\{1, 2\}$ -projective.*

Proof. Assume that f_Γ is a $\{0, 2\}\{1, 2\}$ -projection to the i -th coordinate. Let α_i have 0 at the j -th coordinate, and h is the $(k + 1)$ -ary Boolean projection to the j -th coordinate. Then $f_\Gamma(h(\alpha_1), h(\alpha_2), \dots, h(\alpha_N)) = h(1, 1, \dots, 1) = 1$ and $h(\alpha_i) = 0$. Contradiction. $\square$

Lemma 50. *Suppose the negation Boolean function is represented by Γ , and Γ is preserved by a 0-stable operation v_0 . Then v_0 is 1-stable.*

Proof. Since v_0 preserves the relation representing the negation and $v_0(1, 0) = 1$, we have $v_0(0, 1) = 0$. Combining this with the fact that v_0 is idempotent and v_0 is a $\{0, 2\}\{1, 2\}$ -projection to the first coordinate, we obtain that v_0 is 1-stable. $\square$

Theorem 51. *Suppose Γ is preserved by a 0-stable operation v_0 , then Γ contains an AND-type relation (equivalently $\text{BoolClo}(\Gamma)$ contains conjunction).*

Proof. Assume that $\text{BoolClo}(\Gamma)$ does not contain conjunction.

By Lemma 49, f_Γ is not $\{0, 2\}\{1, 2\}$ -projective, hence there should be a relation in Γ that is not preserved by f_Γ . Let ρ be a relation of the minimal arity in Γ that is not preserved by f_Γ . Thus, there exists a matrix M whose columns are tuples from ρ such that f_Γ applied to the rows of M gives $\beta \notin \rho$. Let L be the arity of ρ .

Note that each row of the matrix should contain different elements, since otherwise we could substitute a constant into ρ and reduce the arity of ρ (recall that f_Γ is idempotent). Since ρ is a relation of the minimal arity, for every i there should be a tuple $\beta_i \in \rho$ such that β_i differs from β only in the i -th coordinate (otherwise, we consider the projection of ρ to all coordinates but i -th, which has the same properties as ρ but smaller arity).

Assume that β has at least two elements equal to 2 and they appear at the i -th and j -th coordinates. Then $s_2(\beta_i, \beta_j) = \beta$, which contradicts the fact that s_2 preserves ρ .

Assume that β contains exactly one 2. W.l.o.g., we assume that $\beta(L) = 2$. By the definition of f_Γ for every $i \in [L - 1]$ there exists a Boolean function $h_i \in \text{BoolClo}(\Gamma)$ (not a constant function) such that the i -th row of M is $(h_i(\alpha_1), \dots, h_i(\alpha_N))$.

Assume that there exists a tuple $\gamma \in \rho$ such that $\gamma(L) = 2$ and $\gamma(i) \in \{0, 1\}$ for $i \in [L - 1]$. By Lemma 50, we have two cases:

Case 1. $\text{BoolClo}(\Gamma)$ does not contain negation. It follows from Post's lattice (see [27]) that all functions in $\text{BoolClo}(\Gamma)$ are monotonic. Thus, h_i is a monotonic function that is not a constant, therefore $\beta(i) = h_i(1, 1, \dots, 1) = 1$ for every $i \in [L - 1]$. Applying the 0-stable operation v_0 to β_L and γ we obtain $(1, 1, \dots, 1, 2) = \beta$, which contradicts the fact that v_0 preserves ρ .

Case 2. v_0 is 1-stable. Applying v_0 to β_L and γ we obtain β , which contradicts the fact that v_0 preserves ρ .

Thus, we proved that ρ does not contain a tuple whose last element is 2 and the remaining elements are from $\{0, 1\}$. For instance, this means that the matrix M contains only 0 and 1. Let R_i be a relation pp-definable from Γ representing h_i . Define a new relation by

$$R(x_1, \dots, x_{k+1}, z) = \exists y_1 \dots \exists y_{L-1} R_1(x_1, \dots, x_{k+1}, y_1) \wedge \dots \wedge R_{L-1}(x_1, \dots, x_{k+1}, y_{L-1}) \wedge \rho(y_1, \dots, y_{L-1}, z).$$

We can check that for each α_i there exists a unique $b_i \in \{0, 1\}$ such that $\alpha_i b_i \in R$. Note that b_i is the i -th element of the last row of the matrix. Since $(1, 1, \dots, 1, h_i(1, 1, \dots, 1)) \in R_i$ and $\beta(i) = \beta_L(i) = h_i(1, 1, \dots, 1)$ for every $i \in [L - 1]$, we have $(1, 1, \dots, 1, \beta_L(L)) \in R$, which means that R represents the corresponding Boolean function and $f_\Gamma(b_1, \dots, b_N) = \beta_L(L)$. This contradicts the fact that $\beta(L) = 2$.

It remains to consider the case when β has no 2. Therefore, for every $i \in [L]$ there exists a Boolean function $h_i \in \text{BoolClo}(\Gamma)$ (not a constant function) such that the i -th row of M is $(h_i(\alpha_1), \dots, h_i(\alpha_N))$. It follows from Post's Lattice (see [27]) that we have only the following two cases.

Case 1. $\text{BoolClo}(\Gamma)$ contains only linear functions. Consider a system of linear equations

$$\begin{cases} h_1(x_1, \dots, x_{k+1}) = h_1(1, \dots, 1) \\ h_2(x_1, \dots, x_{k+1}) = h_2(1, \dots, 1) \\ \dots \\ h_L(x_1, \dots, x_{k+1}) = h_L(1, \dots, 1) \end{cases}$$

Since $L \leq k$, there should be a solution of this system of equations different from $(1, 1, \dots, 1)$. Let this solution be equal to α_i for $i \in [N]$. Therefore, the i -th column of the matrix M is

equal to $(h_1(1, \dots, 1), h_2(1, \dots, 1), \dots, h_L(1, 1, \dots, 1))$, which is equal to β . This contradicts the fact that $\beta \notin \rho$.

Case 2. $\text{BoolClo}(\Gamma)$ is the clone generated by disjunction and constants. Since every operation h_j is monotonic and not constant, $\beta = (1, 1, \dots, 1)$. Since h_j is not a constant for every $j \in [L]$, we can choose n_j such that $h_j(x_1, \dots, x_{k+1}) \geq x_{n_j}$. Let α_i be the tuple having 1 only on coordinates $n_1, \dots, n_L$ (such α_i exists because $L < k+1$). Then the i -th column of the matrix M is equal to $(1, 1, \dots, 1)$, which contradicts the fact that $\beta \notin \rho$. $\square$

In the same way we can prove the following theorem.

Theorem 52. *Suppose Γ is preserved by a 1-stable operation, then Γ contains an OR-type relation (equivalently $\text{BoolClo}(\Gamma)$ contains disjunction).*

9.3 The proof of the EGP classification

Lemma 53. *Suppose $\text{Pol}(\Gamma)$ does not have a 0-stable operation. Then Γ contains one of the two relations $\begin{pmatrix} 0 & 1 & 2 & 0 & 2 \\ 0 & 0 & 0 & 2 & 2 \end{pmatrix}, \begin{pmatrix} 0 & 1 & 2 & 0 & 2 \\ 1 & 1 & 1 & 2 & 2 \end{pmatrix}$.*

Proof. Let δ_1 and δ_2 be the relations generated from $\begin{pmatrix} 1 & 0 \\ 0 & 2 \end{pmatrix}$ and $\begin{pmatrix} 1 & 0 \\ 1 & 2 \end{pmatrix}$ using $\text{Pol}(\Gamma)$, respectively. Note that since $s_2 \in \text{Pol}(\Gamma)$ we have $(2, 2) \in \delta_1$ and $(2, 2) \in \delta_2$. Assume that $(1, 2) \in \delta_1$ and $(1, 2) \in \delta_2$. Then there exist operations $f_1, f_2 \in \text{Pol}(\Gamma)$ such that $f_1(1, 0) = f_2(1, 0) = 1$, $f_1(0, 2) = f_2(1, 2) = 2$. Since f_1 and f_2 are $\{0, 2\}\{1, 2\}$ -projective, they should be $\{0, 2\}\{1, 2\}$ -projections to the first coordinate and $f_1(2, x) = f_2(2, x) = 2$. Furthermore, $f_1(0, 2), f_2(0, 2) \in \{0, 2\}$ as f_1, f_2 preserves $\{0, 2\} \in \Gamma$. Put $f(x, y) = f_1(f_2(x, y), y)$, then $f(1, 0) = 1$, $f(0, 2) = f(1, 2) = f(2, 0) = 2$. Therefore, f is a 0-stable operation. Contradiction.

Assume that $(1, 2) \notin \delta_1$. Then by Lemma 46, $\psi_1 = (\{0, 2\} \times \{a, 2\} \times \{a, 2\}) \setminus \{(0, a, 2)\} \in \Gamma$ for some $a \in \{0, 1\}$. Then one of the two necessary relations is defined by the formula $\delta'_1(x, y) = \exists z \delta_1(x, z) \wedge \psi_1(z, a, y)$.

It remains to consider the case when $(1, 2) \notin \delta_2$. Then by Lemma 46, $\psi_2 = (\{1, 2\} \times \{a, 2\} \times \{a, 2\}) \setminus \{(1, a, 2)\} \in \Gamma$ for some $a \in \{0, 1\}$. Then one of the two necessary relations is defined by the formula $\delta'_2(x, y) = \exists z \delta_2(x, z) \wedge \psi_2(z, a, y)$. $\square$

In the same way we can prove the following lemma.

Lemma 54. *Suppose $\text{Pol}(\Gamma)$ does not have a 1-stable operation. Then Γ contains one of the two relations $\begin{pmatrix} 0 & 1 & 2 & 1 & 2 \\ 0 & 0 & 0 & 2 & 2 \end{pmatrix}, \begin{pmatrix} 0 & 1 & 2 & 1 & 2 \\ 1 & 1 & 1 & 2 & 2 \end{pmatrix}$.*

Lemma 55. *Suppose $\text{Pol}(\Gamma)$ does not have a 01-stable operation, then there exists $b \in \{0, 1\}$ such that $\sigma_0, \sigma_1 \in \Gamma$, where*

$$\begin{aligned} \sigma_0 &= \{(a_1, a_2, a_3) : a_1 \in A, a_2, a_3 \in \{b, 2\}, (a_1 \in \{0, 2\} \vee a_2 = a_3)\}, \\ \sigma_1 &= \{(a_1, a_2, a_3) : a_1 \in A, a_2, a_3 \in \{b, 2\}, (a_1 \in \{1, 2\} \vee a_2 = a_3)\}. \end{aligned}$$

Proof. Let δ_0 and δ_1 be the relations generated from $\begin{pmatrix} 0 & 0 & 1 \\ 1 & 0 & 1 \\ 0 & 2 & 2 \end{pmatrix}$ and $\begin{pmatrix} 0 & 0 & 1 \\ 1 & 0 & 1 \\ 1 & 2 & 2 \end{pmatrix}$ using $\text{Pol}(\Gamma)$, respectively. Since the semilattice preserves δ_0 and δ_1 , the tuples $(0, 2, 2), (2, 1, 2)$ belong to δ_0 and δ_1 .

If both δ_0 and δ_1 contain $(0, 1, 2)$ then there exist operations f_0 and f_1 in $\text{Pol}(\Gamma)$ such that $f_0(0, 0, 1) = f_1(0, 0, 1) = 0$, $f_0(1, 0, 1) = f_1(1, 0, 1) = 1$, $f_0(0, 2, 2) = f_1(1, 2, 2) = 2$. Since f_0 and f_1 are $\{0, 1\}\{0, 2\}$ -projective, $f_0(2, a, b) = f_1(2, a, b) = 2$ for all $a, b \in A$. Then the operation $f(x, y, z) = f_1(f_0(x, y, z), y, z)$ is a 01-stable operation, which contradicts our assumptions.

Assume that δ_0 does not contain $(0, 1, 2)$. By Lemma 46, for some $b \in \{0, 1\}$ the relation $\psi = (\{0, 2\} \times \{b, 2\} \times \{b, 2\}) \setminus \{(0, b, 2)\}$ is in Γ . Then σ_0 and σ_1 can be defined by

$$\sigma_0(x, y, z) = \exists t \delta_0(0, x, t) \wedge \psi(t, y, z) \wedge \psi(t, z, y),$$

$$\sigma_1(x, y, z) = \exists t \delta_0(x, 1, t) \wedge \psi(t, y, z) \wedge \psi(t, z, y).$$

Assume that δ_1 does not contain $(0, 1, 2)$. By Lemma 46, for some $b \in \{0, 1\}$ the relation $\psi = (\{1, 2\} \times \{b, 2\} \times \{b, 2\}) \setminus \{(1, b, 2)\}$ is in Γ . Then σ_0 and σ_1 can be defined by

$$\sigma_0(x, y, z) = \exists t \delta_1(0, x, t) \wedge \psi(t, y, z) \wedge \psi(t, z, y),$$

$$\sigma_1(x, y, z) = \exists t \delta_1(x, 1, t) \wedge \psi(t, y, z) \wedge \psi(t, z, y).$$

□

Combining the above lemma with Lemma 17 we obtain the following corollary.

Corollary 56. *Suppose $\text{Pol}(\Gamma)$ does not have a 01-stable operation, then $\text{QCSP}(\Gamma)$ is PSpace-complete.*

Lemma 57. *Suppose $\text{Pol}(\Gamma)$ does not have a 0-stable or 1-stable operation, then $\text{QCSP}(\Gamma)$ is co-NP-hard.*

Proof. By Lemmas 53 and 54, we have the relations $\delta_0 = \begin{pmatrix} 0 & 1 & 2 & 0 & 2 \\ a_0 & a_0 & a_0 & 2 & 2 \end{pmatrix}$ and $\delta_1 = \begin{pmatrix} 0 & 1 & 2 & 1 & 2 \\ a_1 & a_1 & a_1 & 2 & 2 \end{pmatrix}$ in Γ for some $a_0, a_1 \in \{0, 1\}$. By Lemma 46, for some $b_0, b_1 \in \{0, 1\}$ the relations $\psi_0 = (\{a_0, 2\} \times \{b_0, 2\} \times \{b_0, 2\}) \setminus \{(a_0, b_0, 2)\}$, $\psi_1 = (\{a_1, 2\} \times \{b_1, 2\} \times \{b_1, 2\}) \setminus \{(a_1, b_1, 2)\}$ are in Γ . We define

$$\sigma_0(x, y, z) = \exists t \delta_0(x, t) \wedge \psi_0(t, y, z) \wedge \psi_0(t, z, y),$$

$$\sigma_1(x, y, z) = \exists t \delta_1(x, t) \wedge \psi_1(t, y, z) \wedge \psi_1(t, z, y).$$

If $b_0 = b_1$ then by Lemma 17 the problem is PSpace-hard, and therefore co-NP-hard.

Since Γ contains $\{0, 2\}^2 \cup \{1, 2\}^2$, if $b_0 = 1$ and $b_1 = 0$ then by Lemma 22 the problem is co-NP-hard. Assume that $b_0 = 0$ and $b_1 = 1$. Put

$$\sigma'_1(x, y, z) = \exists z' \sigma_1(x, 1, z') \wedge \sigma_0(z', y, z),$$

$$\sigma'_0(x, y, z) = \exists z' \sigma_0(x, 0, z') \wedge \sigma_1(z', y, z),$$

and notice that σ'_0 and σ'_1 are the relations required in Lemma 22 for the proof of co-NP-hardness. □

Lemma 58. *Suppose Γ is not preserved by $f_{0,2}$, and Γ is preserved by $s_{0,2}$,*

$$\begin{pmatrix} 0 & 1 & 1 & 1 & 2 & 2 & 2 \\ 1 & 0 & 1 & 2 & 0 & 1 & 2 \end{pmatrix}, \begin{pmatrix} 0 & 1 & 1 & 1 & 2 & 2 & 2 \\ 0 & 0 & 1 & 2 & 0 & 1 & 2 \end{pmatrix}, \begin{pmatrix} 0 & 1 & 2 & 2 & 2 \\ 0 & 1 & 0 & 1 & 2 \end{pmatrix} \in \Gamma,$$

then the relation $\delta(x, y, z) = (x \neq 0) \vee (y = z)$ can be pp-defined from Γ .

Proof. First, denote the relations from the statement by R_1 , R_2 and R_3 , respectively. Let

ρ_1 be the relation generated from $\begin{pmatrix} 0 & 0 & 2 \\ 1 & 0 & 0 \\ 1 & 0 & 1 \end{pmatrix}$ using $\text{Pol}(\Gamma)$. Assume that $(0, 1, 2) \in \rho_1$, then

there exists a function $f \in \text{Pol}(\Gamma)$ such that $f(0, 0, 2) = 0$, $f(1, 0, 0) = 1$, and $f(1, 0, 1) = 2$. Since f is $\{0, 2\}\{1, 2\}$ -projective, f always returns the first variable or 2. Since f preserves R_1 and $f(0, 0, 2) = 0$, we have $f(1, 1, a) = 1$ for every a . Since f preserves R_2 and $f(0, 0, 2) = 0$, we have $f(0, 0, a) = 0$ for every a .

Let us show that $f_{0,2}(x, y, z) = f(s_{0,2}(x, y), y, z)$. If $x = y$ then by the above property we have $f(s_{0,2}(x, y), y, z) = f(x, y, z) = x$. Also, if $y = z = 0$ then since $f(1, 0, 0) = 1$ and $f(0, 0, 0) = 0$, we obtain $f(s_{0,2}(x, y), y, z) = f(x, y, z) = x$. If $x = 2$ or $y = 2$ then $f(s_{0,2}(x, y), y, z)$ returns 2. If $x = 0$ and $y = 1$ then $s_{0,2}(x, y) = 2$ and $f(s_{0,2}(x, y), y, z) = 2$. If $x = z = 1$ and $y = 0$, then $f(s_{0,2}(x, y), y, z) = f(x, y, z) = f(1, 0, 1) = 2$. Thus, we considered all cases and showed that $f_{0,2}(x, y, z) = f(s_{0,2}(x, y), y, z)$, which contradicts the fact that $f_{0,2} \notin \text{Pol}(\Gamma)$. Therefore $(0, 1, 2) \notin \rho_1$. Put

$$\rho_2(x, y, z) = \exists x' \exists y' \exists z' \rho_1(x', y', z') \wedge R_2(x, x') \wedge R_3(y, y') \wedge R_3(z', z).$$

Note that $\rho_1 \subseteq \rho_2$, and $(0, 1, 2) \notin \rho_2$. Since ρ_1 is preserved by s_2 and $s_{0,2}$, we have $s_2 \begin{pmatrix} 0 & 2 \\ 0 & 0 \\ 0 & 1 \end{pmatrix} = \begin{pmatrix} 2 \\ 0 \\ 2 \end{pmatrix} \in \rho_1$, $s_{0,2} \begin{pmatrix} 0 & 2 \\ 1 & 0 \\ 1 & 2 \end{pmatrix} = \begin{pmatrix} 2 \\ 1 \\ 2 \end{pmatrix} \in \rho_1$, $s_2 \begin{pmatrix} 2 & 2 \\ 0 & 1 \\ 2 & 2 \end{pmatrix} = \begin{pmatrix} 2 \\ 2 \\ 2 \end{pmatrix} \in \rho_1$. Therefore ρ_2 contains all the tuples $(1, a, b)$, $(2, a, b)$ for every $a, b \in A$ (we just put $x' = z' = 2$ in the definition of ρ_2).

Define $\delta'(x, y, z) = \rho_2(x, y, z) \wedge \rho_2(x, z, y)$. By the previous fact, δ' contains all the tuples with the first element different from 0. If ρ_2 contains $(0, 1, 0)$ then we may apply s_2 to $(0, 1, 0)$ and $(0, 1, 1)$ to obtain $(0, 1, 2)$, which is not from ρ_2 . Thus $(0, 1, 0) \notin \rho_2$. Therefore, $(0, 1, 0), (0, 0, 1), (0, 1, 2), (0, 2, 1) \notin \delta'$.

Assume that δ' contains the tuple $(0, 0, 2)$. Then applying $s_{0,2}$ to $(0, 1, 1)$ and $(0, 0, 2)$ we get a tuple $(0, 1, 2)$ which contradicts our assumptions. Therefore $\delta' = \delta$, which completes the proof. $\square$

Lemma 59. Suppose Γ is preserved by a 0-stable operation but not preserved by $g_{0,2}$, and $R = \begin{pmatrix} 0 & 1 & 2 & 2 & 2 \\ 0 & 1 & 0 & 1 & 2 \end{pmatrix} \in \Gamma$. Then Γ contains a relation $\delta \in \left\{ \begin{pmatrix} 0 & 2 \\ 1 & 2 \end{pmatrix}, \begin{pmatrix} 0 & 2 & 2 \\ 1 & 1 & 2 \end{pmatrix} \right\}$.

Proof. Let ρ be the relation generated from $\begin{pmatrix} 0 & 2 \\ 1 & 0 \\ 1 & 2 \end{pmatrix}$ using $\text{Pol}(\Gamma)$. Assume that $(0, 1, 2) \in \rho$,

then there exists an operation $f \in \text{Pol}(\Gamma)$ such that $f(0, 2) = 0$, $f(1, 0) = 0$, and $f(1, 2) = 2$. Let us check that $f = g_{0,2}$, which will give us a contradiction. Since f preserves R and $f(0, 2) = 0$, we have $f(0, a) = 0$ for any a . We also know that $f(1, 0) = 1$, $f(1, 1) = 1$, $f(1, 2) = 2$. It remains to check that f returns 2 if the first variable equals 2. This follows from the fact that f is $\{0, 2\}\{1, 2\}$ -projective.

Therefore $(0, 1, 2) \notin \rho$. Since ρ is preserved by a 0-stable operation and $(0, 1, 1), (2, 0, 2) \in \rho$, we have $(2, 1, 2) \in \rho$. Put $\delta(x, y) = \rho(x, 1, y) \wedge x \in \{0, 2\} \wedge y \in \{1, 2\}$. We know that $(0, 1), (2, 2) \in \delta$, $(0, 2) \notin \delta$, which completes the proof. $\square$

Now we are ready to prove the classification of the complexity for constraint languages we consider in this section.

Theorem 60. Suppose Γ is a finite constraint language on $\{0, 1, 2\}$ with constants, Γ is preserved by s_2 , and $\text{Pol}(\Gamma)$ is $\{0, 2\}\{1, 2\}$ -projective. Then $\text{QCSP}(\Gamma)$ is

1. *PSpace-complete, if $\text{Pol}(\Gamma)$ does not contain a 01-stable operation.*
2. *in P, if Γ is preserved by $s_{a,2}$ and $g_{a,2}$ for some $a \in \{0,1\}$.*
3. *in P, if Γ is preserved by $f_{a,2}$ for some $a \in \{0,1\}$.*
4. *co-NP-complete, otherwise.*

Proof. Let r be the maximal arity of relations in Γ , let $k = \max(4, r)$. Note that if we add to Γ all the relations of arity at most k that can be pp-defined from Γ then we do not change the complexity of $\text{QCSP}(\Gamma)$. Also, adding the equality relation to Γ does not change the complexity since the equalities can be propagated out. Another important fact is that adding pp-definable relations and the equality relation to a constraint language does not affect the set of polymorphisms $\text{Pol}(\Gamma)$. Thus, it is sufficient to prove the claim only for constraint languages Γ containing all such relations. Additionally, since $\text{Pol}(\Gamma)$ is $\{0,2\}\{1,2\}$ -projective, $\{0,2\}$, $\{1,2\}$, and $\{0,2\}^2 \cup \{1,2\}^2$ are from Γ . Therefore, Γ satisfies all the assumptions we formulated in the beginning of this section.

Case 1 follows from Corollary 56. Otherwise, Γ is preserved by a 01-stable operation, and by Lemma 19 the problem $\text{QCSP}(\Gamma)$ is in co-NP.

If Γ is not preserved by a 0-stable or 1-stable operation, then by Lemma 57 $\text{QCSP}(\Gamma)$ is co-NP-complete.

Then we may assume that Γ is preserved by a 0-stable operation (the 1-stable case can be considered in the same way). If Γ is preserved by $f_{0,2}$ then by Corollary 41 $\text{QCSP}(\Gamma)$ is in P. Similarly, if Γ is preserved by $s_{0,2}$ and $g_{0,2}$ then by Corollary 32 $\text{QCSP}(\Gamma)$ is in P. Thus we assume that $f_{0,2}$ does not preserve Γ , and $s_{0,2}$ or $g_{0,2}$ does not preserve Γ .

By Theorem 51, Γ contains an AND-type relation δ . We can check that $\delta_1(x, y) = \delta(x, x, y)$ is equal to $\begin{pmatrix} 0 & 1 & 2 & 2 & 2 \\ 0 & 1 & 0 & 1 & 2 \end{pmatrix}$ (to derive $(2, 1)$ we apply a 0-stable operation to $(1, 1)$ and $(2, 0)$).

If Γ is also preserved by a 1-stable operation, then by Theorem 52, Γ contains an OR-type relation, and by Lemma 20, $\text{QCSP}(\Gamma)$ is co-NP-complete.

Thus, we assume that Γ is not preserved by a 1-stable operation. By Lemma 54, Γ contains a relation $\delta_2 \in \left\{ \begin{pmatrix} 0 & 1 & 2 & 1 & 2 \\ 0 & 0 & 0 & 2 & 2 \end{pmatrix}, \begin{pmatrix} 0 & 1 & 2 & 1 & 2 \\ 1 & 1 & 1 & 2 & 2 \end{pmatrix} \right\}$. Define

$$\delta_3(x, y) = \exists x' \exists y' \delta_2(x, x') \wedge \delta(x, x', y') \wedge \delta_1(y', y).$$

We can check that $(0, a) \in \delta_3$ implies $a = 0$. Applying s_2 to the tuples $(1, 1, 1), (1, 0, 0)$, and to the tuples $(1, 1, 1), (0, 0, 0)$, we obtain that $(1, 2, 2), (2, 2, 2) \in \delta$. Then $(1, a), (2, a) \in \delta_3$ for every a (put $x' = y' = 2$ in the definition of δ_3), which means that $\delta_3 = \begin{pmatrix} 0 & 1 & 1 & 1 & 2 & 2 & 2 \\ 0 & 0 & 1 & 2 & 0 & 1 & 2 \end{pmatrix}$.

Assume that a 0-stable operation we have is h_0 . Since h_0 is $\{0,2\}\{1,2\}$ -projective, it returns the first variable or 2. If $h_0(0, 1) = 0$, then h_0 is also 1-stable operation, which contradicts our assumptions. Thus $h_0(0, 1) = 2$. If $h_0(0, 2) = 0$ then we get a contradiction with the fact that h_0 preserves δ_3 . Thus $h_0 = s_{0,2}$, which means that Γ is not preserved by $g_{0,2}$. By Lemma 59, Γ contains a relation $\delta_4 \in \left\{ \begin{pmatrix} 0 & 2 \\ 1 & 2 \end{pmatrix}, \begin{pmatrix} 0 & 2 & 2 \\ 1 & 1 & 2 \end{pmatrix} \right\}$. Define

$$\delta_5(x, y) = \exists z_1 \exists z_2 \delta_3(x, z_1) \wedge \delta_4(z_1, z_2) \wedge \delta_1(z_2, y).$$

We can check that $\delta_5 = \begin{pmatrix} 0 & 1 & 1 & 1 & 2 & 2 & 2 \\ 1 & 0 & 1 & 2 & 0 & 1 & 2 \end{pmatrix} \in \Gamma$. Applying Lemma 58 to δ_5 , δ_3 , and δ_1 , we derive that the required relation $(x \neq 0) \vee (y = z)$ can be pp-defined from Γ . Then by Lemma 21, $\text{QCSP}(\Gamma)$ is co-NP-complete. $\square$

10 Main result for 3-element domain: proof of Theorem 6

Lemma 61. *Suppose Γ is a finite constraint language on $\{0, 1, 2\}$ with constants, such that $\text{Pol}(\Gamma)$ has the EGP property and has no WNU. Then $\text{QCSP}(\Gamma)$ is PSpace-complete.*

Proof. Since $\text{Pol}(\Gamma)$ has the EGP property, every operation of $\text{Pol}(\Gamma)$ is $\alpha\beta$ -projective for some α and β , strict subsets of $\{0, 1, 2\}$, so that $\alpha \cup \beta = \{0, 1, 2\}$. If $\alpha \cap \beta = \emptyset$, then $\alpha \times \alpha \cup \beta \times \beta$ is an equivalence relation preserved by $\text{Pol}(\Gamma)$ and any operation of $\text{Pol}(\Gamma)$ modulo this congruence is a projection. It follows therefore from Lemma 5 in [18] and Theorem 5.2 of [5] that $\text{QCSP}(\Gamma)$ is PSpace-complete.

Otherwise, w.l.o.g. we assume that $\alpha = \{0, 2\}$ and $\beta = \{1, 2\}$. Since there does not exist a WNU operation, it is known from [7, 24] that there should be a factor of size at least two whose operations are projections.

We consider two cases. There exists a congruence σ such that $\text{Pol}(\Gamma)/\sigma$ has only projections. This case we have already covered using Lemma 5 in [18] and Theorem 5.2 of [5].

Thus we may assume, there exists a subset $B \subsetneq \{0, 1, 2\}$ of size two such that all operations on B are projections and B is a subalgebra of $\text{Pol}(\Gamma)$.

Assume that $B = \{0, 2\}$. Then $\delta = \{0, 2\}^3 \setminus \{(2, 2, 0), (2, 0, 2)\}$ is preserved by $\text{Pol}(\Gamma)$. Recall the relations $\sigma = \alpha^2 \cup \beta^2$ and $\sigma_2(x_1, x_2, x_3, x_4) = \sigma(x_1, x_2) \vee \sigma(x_3, x_4)$ are pp-definable from Γ because $\text{Pol}(\Gamma)$ is $\alpha\beta$ -projective. We have $\sigma'_0(x, y, z) = \exists x' \sigma(x, x') \wedge \delta(x', y, z)$, $\sigma'_1(x, y, z) = \exists x' \sigma_2(x, 1, x', 1) \wedge \delta(x', y, z)$, which by Lemma 17 guarantees PSpace-completeness of $\text{QCSP}(\Gamma)$.

If $B = \{1, 2\}$, then we are in a symmetric case to the last and we again apply Lemma 17 (this time $b = 1$) to prove PSpace-completeness.

If $B = \{0, 1\}$, then we can complete the argument in various ways. Note that B is a subalgebra, and we already know that $\{0, 2\}$ and $\{1, 2\}$ are subalgebras too. It follows that we are in the conservative case, and the result comes from Theorem 4. $\square$

Note that the above lemma does not hold on a larger domain. As a counter example on the 4-element domain we can take the constraint language we build in Corollary 8.

Now, we are ready to prove Theorem 6.

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

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

Proof. Suppose $\text{Pol}(\Gamma)$ has the PGP property. We know from Theorem 2 that $\text{QCSP}(\Gamma)$ can be reduced to a polynomial collection of instances of $\text{CSP}(\Gamma)$. If Γ additionally has a WNU then it follows from [8, 29] that $\text{CSP}(\Gamma)$, and therefore also $\text{QCSP}(\Gamma)$, is in P. If $\text{Pol}(\Gamma)$ does not

have a WNU operation, then NP-hardness follows from the identity reduction from CSP(Γ), whose NP-hardness has long been known (e.g. see again [8, 29]).

Suppose $\text{Pol}(\Gamma)$ has the EGP property. If $\text{Pol}(\Gamma)$ has no WNU operation then by Lemma 61 QCSP(Γ) is PSpace-complete.

Suppose $\text{Pol}(\Gamma)$ contains a WNU operation w . Since $\text{Pol}(\Gamma)$ has the EGP property, all operations of $\text{Pol}(\Gamma)$ are $\alpha\beta$ -projective for some α and β .

If $\alpha \cap \beta = \emptyset$, then $\text{Pol}(\Gamma)$ cannot have a WNU operation, which contradicts our assumption.

Suppose $\alpha \cap \beta \neq \emptyset$. If $\text{Pol}(\Gamma)$ contains an ab -stable operation f , that is $f(x, a, b) = x$ and $f(x, c, c) = c$ for $\{a, b, c\} = \{0, 1, 2\}$, then, since f is $\alpha\beta$ -projective, we have $c \in \alpha \cap \beta$. Similarly, if $\text{Pol}(\Gamma)$ contains $s_{a,c}$, $g_{a,c}$, or $f_{a,c}$, then $c \in \alpha \cap \beta$. Since both variables of the operation $f(x, y) = w(x, \dots, x, y)$ should be $\alpha\beta$ -projective, f is the semilattice operation s_c . W.l.o.g. we assume that $c = 2$, then the remaining part of the classification follows from Theorem 60. $\square$

11 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.

Even though we consider only finite constraint languages in this paper, we believe our classification of the complexity for a three-element domain holds for infinite constraint languages (relations are given by the list of tuples). To prove this stronger result we would need to add many technicalities, which would make our current proof even more complicated. For instance, for the algorithm in Section 8 to work in polynomial time we need to enforce that any constraint relation contains exponentially many tuples on its arity (see line 19 of the pseudocode). This can be achieved by showing that any relation preserved by the semilattice can be decomposed in polynomial time into several relations with exponentially many tuples (see [30] for such a decomposition). That is why we decided to limit ourselves to finite constraint languages.

Acknowledgements

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.

References

- [1] Libor Barto, Jakub Opršal, and Michael Pinsker. The wonderland of reflections. *Israel Journal of Mathematics*, 223(1):363–398, Feb 2018.
- [2] Libor Barto and Michael Pinsker. 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*, pages 615–622, 2016.
- [3] Manuel Bodirsky and Hubie Chen. Relatively quantified constraint satisfaction. *Constraints*, 14(1):3–15, 2009.
- [4] V. G. Bodnarchuk, L. A. Kaluzhnin, V. N. Kotov, and B. A. Romov. Galois theory for post algebras parts I and II. *Cybernetics*, (5):243–252, 531–539, 1969.
- [5] Ferdinand Börner, Andrei A. Bulatov, Hubie Chen, Peter Jeavons, and Andrei A. Krokhin. The complexity of constraint satisfaction games and QCSP. *Inf. Comput.*, 207(9):923–944, 2009.
- [6] Joshua Brakensiek and Venkatesan Guruswami. 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*, pages 1782–1801, 2018.
- [7] Andrei Bulatov, Peter Jeavons, and Andrei Krokhin. Classifying the complexity of constraints using finite algebras. *SIAM J. Comput.*, 34(3):720–742, March 2005.
- [8] Andrei A. Bulatov. A dichotomy theorem for nonuniform CSPs. In *2017 IEEE 58th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 319–330, 2017.
- [9] Samuel R. Buss and Louise Hay. On truth-table reducibility to SAT. *Inf. Comput.*, 91(1):86–102, 1991.
- [10] Catarina Carvalho, Florent R. Madelaine, and Barnaby Martin. From complexity to algebra and back: digraph classes, collapsibility and the PGP. In *30th Annual IEEE Symposium on Logic in Computer Science (LICS)*, 2015.
- [11] Catarina Carvalho, Barnaby Martin, and Dmitriy Zhuk. 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*, pages 27:1–27:14, 2017.
- [12] Hubie Chen. 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*, pages 168–181, 2004.
- [13] Hubie Chen. A rendezvous of logic, complexity, and algebra. *SIGACT News*, 37(4):85–114, 2006.
- [14] Hubie Chen. The complexity of quantified constraint satisfaction: Collapsibility, sink algebras, and the three-element case. *SIAM J. Comput.*, 37(5):1674–1701, 2008.

- [15] Hubie Chen. Quantified constraint satisfaction and the polynomially generated powers property. *Algebra universalis*, 65(3):213–241, 2011. An extended abstract appeared in ICALP B 2008.
- [16] Hubie Chen. Meditations on quantified constraint satisfaction. In *Logic and Program Semantics - Essays Dedicated to Dexter Kozen on the Occasion of His 60th Birthday*, pages 35–49, 2012.
- [17] Hubie Chen, Florent R. Madelaine, and Barnaby Martin. Quantified constraints and containment problems. *Logical Methods in Computer Science*, 11(3), 2015.
- [18] Hubie Chen and Peter Mayr. Quantified constraint satisfaction on monoids. In *25th EACSL Annual Conference on Computer Science Logic, CSL 2016, August 29 - September 1, 2016, Marseille, France*, pages 15:1–15:14, 2016.
- [19] David Geiger. Closed systems of functions and predicates. *Pacific journal of mathematics*, 27(1):95–100, 1968.
- [20] Vladimir Kolmogorov, Andrei A. Krokhin, and Michal Rolínek. The complexity of general-valued CSPs. *SIAM J. Comput.*, 46(3):1087–1110, 2017.
- [21] Victor Lagerkvist and Magnus Wahlström. The power of primitive positive definitions with polynomially many variables. *J. Log. Comput.*, 27(5):1465–1488, 2017.
- [22] Thomas Lukasiewicz and Enrico Malizia. A novel characterization of the complexity class Θ_k^P based on counting and comparison. *Theor. Comput. Sci.*, 694:21–33, 2017.
- [23] Florent R. Madelaine and Barnaby Martin. On the complexity of the model checking problem. *SIAM J. Comput.*, 47(3):769–797, 2018.
- [24] M. Maróti and R. McKenzie. Existence theorems for weakly symmetric operations. *Algebra Universalis*, 59(3), 2008.
- [25] Barnaby Martin. Quantified Constraints in Twenty Seventeen. In Andrei Krokhin and Stanislav Zivny, editors, *The Constraint Satisfaction Problem: Complexity and Approximability*, volume 7 of *Dagstuhl Follow-Ups*, pages 327–346. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, Dagstuhl, Germany, 2017.
- [26] Christos H. Papadimitriou. *Computational Complexity*. Addison-Wesley, 1994.
- [27] E. L. Post. *The Two-Valued Iterative Systems of Mathematical Logic*. Annals of Mathematics Studies, no. 5. Princeton University Press, Princeton, N. J., 1941.
- [28] Thomas J. Schaefer. The complexity of satisfiability problems. In *Proceedings of the Tenth Annual ACM Symposium on Theory of Computing, STOC '78*, pages 216–226, New York, NY, USA, 1978. ACM.
- [29] D. Zhuk. A proof of CSP dichotomy conjecture. In *2017 IEEE 58th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 331–342, Oct 2017.
- [30] Dmitriy Zhuk. A modification of the csp algorithm for infinite languages. *arXiv preprint arXiv:1803.07465*, 2018.
- [31] Dmitriy Zhuk. The size of generating sets of powers. *Journal of Combinatorial Theory, Series A*, 167:91–103, 2019.