The complete classification for quantified equality constraints

Dmitriy Zhuk, Barnaby Martin and Michał Wrona

August 5, 2022

Abstract

We prove that QCSP(N; x = y → y = z) is PSpace-complete, settling a question open for more than ten years. This completes the complexity classification for the QCSP over equality languages as a trichotomy between Logspace, NP-complete and PSpace-complete.

We additionally settle the classification for bounded alternation QCSP(Γ), for Γ an equality language. Such problems are either in Logspace, NP-complete, co-NP-complete or rise in complexity in the Polynomial Hierarchy.

1 Introduction

The quantified constraint satisfaction problem, QCSP(∆), asks whether some input sentence, whose quantifier-free part is positive conjunctive, is true on the constraint language ∆ comprised of a set of relations over the same domain D. The QCSP is itself a generalisation of its better-studied brother, the constraint satisfaction problem, CSP(∆), in which one assumes the quantification of the sentence is entirely existential. By now, a great deal is known about the CSP, while many important questions for the QCSP remain open.

The celebrated Feder-Vardi Dichotomy Conjecture, which asserts that CSP(Δ) is always either in P or is NP-complete, when Δ is over a finite domain, has been proved independently in [12] and [24, 25]. This provides the basis for some CSP classifications over an infinite domain [8, 9]. In the meantime, the study of infinite-domain CSPs has itself matured to become a field of active research incorporating Universal Algebra and Model Theory. Important results from its formative years include [7, 11, 8] (as well as the earlier [22]), but most of the post-modern work concerns the finitely-bounded ω-categorical case [1, 2].

At present, we are far from a complexity classification for finite domain QCSPs. The 3-element case with constants is settled in [27], where evidence is also given that the classification is more complicated than previously thought. The study of infinite-domain QCSPs is also far less advanced than

1
that for infinite-domain CSPs, with only a partial known classification in
the temporal case (see [14] [13] [19] [23]).

The modern, systematic study of infinite-domain CSPs began with a
complexity classification for equality languages, which are defined as those
which have a first-order definition in \((\mathbb{N}; =)\) [6]. For such languages, the CSP
is always in NP and the QCSP always in PSpace. It made sense to begin
the systematic study of infinite-domain QCSPs similarly, with the equality
languages, spawning the works [3] [4]. There was an error in the conference
version [3] which was corrected in the journal version [4] at the cost of
a trichotomy between Logspace, NP-complete and Co-NP-hard instead of
Logspace, NP-complete and PSpace-complete. Thus, a gap was left in this
foundational result. It was known that there were PSpace-complete cases,
e.g. QCSP(\(\mathbb{N}; x = y \rightarrow u = v\)), but it was not known whether QCSP(\(\mathbb{N}; x = y \rightarrow y = z\)) was Co-NP-complete or PSpace-complete (or even somewhere
in between). Indeed, this constraint language became notorious; more so,
because a solution of it as either Co-NP-complete or PSpace-complete would
complete the classification.

In an invited talk at the Arbeitstagung Allgemeine Algebra (Workshop on
General Algebra – AAA98) in July 2019 in Dresden, Hubie Chen posed the
complexity of QCSP(\(\mathbb{N}; x = y \rightarrow y = z\)) as one of three open problems in the
area of the complexity of constraints that apparently needed new insights
to resolve [17]. In this paper we settle the complexity of QCSP(\(\mathbb{N}; x = y \rightarrow
y = z\)) as PSpace-complete.

We further consider the bounded alternation restrictions of the QCSP,
in which there is an a priori bound on the quantifier alternations. The cor-
responding problems are denoted \(\Sigma_k\)-QCSP(\(\Gamma\)) and \(\Pi_k\)-QCSP(\(\Gamma\)), when
the input is restricted to be \(\Sigma_k\) and \(\Pi_k\), respectively. Such problems arise espe-
cially naturally in the 2-element (Boolean) case, for example propositional
abduction and prioritization (see [20]). The complexity classification for
bounded alternation QCSP has been known to display differences from the
unbounded case since [16]. Indeed, there are constraint languages \(\Gamma_1\) and
\(\Gamma_2\), both on three elements, so that QCSP(\(\Gamma_1\)) and QCSP(\(\Gamma_2\)) are PSpace-
complete, yet their bounded alternation versions are Co-NP-complete and
rising in the polynomial hierarchy, respectively.

It has been thought that bounded alternation QCSP(\(\mathbb{N}; x = y \rightarrow u = v\))
is Co-NP-complete. Indeed, Hubie Chen has been saying this for many
years. The idea was that the methods of [10] could be adapted for
(\(\mathbb{N}; x = y \rightarrow u = v\)). However, the direct application of [10] is not completely trivial
in the infinite-domain case. Our proof does not use the machinery of [10],
though it has various similarities with it.

We prove a complete complexity classification for the bounded alterna-
tion QCSP(\(\Gamma\)), along the lines that the authors of [4] would have expected,
where \(\Gamma\) is an equality language. Each of the complexity classes – Logspace,
NP-complete, Co-NP-complete and rising in the polynomial hierarchy – ap-
1.1 The classifications

Equality constraint languages admit quantifier elimination so let us assume \( \Gamma \) is a finite set of relations over an infinite countable set (e.g. \( \mathbb{N} \)) where each relation is defined by some Boolean combination of atoms of the form \( x = y \). Let us additionally assume this Boolean combination is in conjunctive normal form (CNF). A relation is negative if it has a CNF definition in which all of the clauses are either equalities, or are disjunctions of negative literals (i.e. of the form \( x \neq y \)). A relation is positive if it has a CNF definition in which all of the literals are positive (i.e. of the form \( x = y \)). A relation is Horn if it has a CNF definition in which each clause contains at most one positive literal. \( \Gamma \) is negative (respectively, positive, Horn) if all of its relations are negative (respectively, positive, Horn). Let \( I \) be the relation \( x = y \rightarrow y = z \).

**Corollary 1.** Let \( \Gamma \) be an equality constraint language. Either

- \( \Gamma \) is negative, and QCSP(\( \Gamma \)) is in Logspace, or else
- \( \Gamma \) is positive, and QCSP(\( \Gamma \)) is NP-complete, or else
- QCSP(\( \Gamma \)) is PSpace-complete.

**Proof.** We follow the argument as laid out in Theorem 5.5 of [4]. If \( \Gamma \) is negative then QCSP(\( \Gamma \)) is in Logspace by Theorem 6.1 of [4]. If \( \Gamma \) is positive but not negative then QCSP(\( \Gamma \)) is NP-complete by Theorem 7.1 of [4]. Otherwise, by Theorem 8.1 of [4], \( I \) is definable as a quantified conjunctive (positive) formula within \( \Gamma \), and the result follows from our Corollary 6. \( \square \)

Let us give examples of each of the three complexity classes in this classification. A non-trivial negative language might be \((x \neq y \lor u \neq v) \land v = w\). A positive language that is not negative might be \(x = y \lor u = v\). Finally, the canonical example for the third class is, of course, \( I \)!

The classification for bounded alternation QCSP over equality languages is as follows.

**Theorem 2.** Let \( \Gamma \) be an equality language and \( k \geq 2 \).

- If \( \Gamma \) is negative, then \( \Pi_k\text{-QCSP}(\Gamma) \) is in Logspace.
- Else, if \( \Gamma \) is positive, then \( \Pi_k\text{-QCSP}(\Gamma) \) is NP-complete.
- Else, if \( \Gamma \) is Horn, then \( \Pi_k\text{-QCSP}(\Gamma) \) is Co-NP-complete.

\(^1\)Indeed, equality sits within all models, so must be dealt with in any discussion of quantifier elimination. For more details on this, we refer the reader to Section 2.7 in [21].
• Else, $\Pi_k$-$QCSP(\Gamma)$ is $\Pi_{k-2}^P$-hard.

2 Preliminaries

We identify a constraint language $\Gamma$ with a set of relations over a fixed domain $D$, which we may always take in this paper to be $\mathbb{N}$. We may also think of this as a first-order relational structure. All $\Gamma$ in this paper will have a finite signature.

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

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

since if it is not it may readily be brought into such a form in polynomial time. Then a solution is a sequence of (Skolem) functions $f_1, \ldots, f_n$ such that

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

is a solution of $\Phi$ for all $x_1, \ldots, x_n$ (i.e. $y_i = f_i(x_1, \ldots, x_i)$). This belies a (Hintikka) game semantics for the truth of a QCSP instance in which a player called Universal (male) plays the universal variables and a player called Existential (female) 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-theoretic parlance.

A formula in conjunctive normal form with 3-literals per clause is called a 3-CNF. We indicate negation of a propositional variable $z_i$ with the overline notation $\overline{z}_i$.

A key role in classifying CSP and QCSP has been played by Universal Algebra. We say that a $k$-ary operation $f$ preserves an $m$-ary relation $R$, whenever $(x_1^1, \ldots, x^1_m), \ldots, (x_k^1, \ldots, x^m_k)$ in $R$, then also $(f(x_1^1), \ldots, x^1_k), \ldots, f(x^m_1, \ldots, x^m_k))$ 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 $\Gamma$ if it preserves every relation from $\Gamma$. The polymorphism clone $\text{Pol}(\Gamma)$ is the set of all polymorphisms of $\Gamma$. The restriction of this to surjective operations is denoted $s\text{Pol}(\Gamma)$.

A formula of the form $\exists y_1 \ldots \exists y_n \Phi$, where $\Phi$ is a conjunction of relations from $\Gamma$ is called a positive primitive formula (pp-formula) over $\Gamma$. If $R(x_1, \ldots, x_n) = \exists y_1 \ldots \exists y_n \Phi$, then we say that $R$ is pp-defined by $\exists y_1 \ldots \exists y_n \Phi$, and $\exists y_1 \ldots \exists y_n \Phi$ is called a pp-definition. If we augment the definition of pp-formula with universal quantification, then we get the richer class of quantified conjunctive formulas for which we can also define quantified conjunctive definitions. If $\Gamma$ is an equality language, it is known that the class of relations pp-definable on $\Gamma$ is precisely those relations that are preserved by the polymorphisms of $\Gamma$ [10]. Furthermore, it is known that the class of relations quantified conjunctive definable on $\Gamma$ is precisely those relations
that are preserved by the surjective polymorphisms of $\Gamma$ [4]. This latter property is famously not yet known to hold for $\omega$-categorical structures in general [18].

A $k$-ary operation $f$ is essentially unary if there exists $i \in [k]$ so that $f(x_1, \ldots, x_k) = g(x_i)$ for some unary function $g$. An operation that is not essentially unary is said to be essential.

3 QCSP$(\mathbb{N}; x = y \rightarrow y = z)$

We will define formulas (constraints) over $I := x = y \rightarrow y = z$ by diagrams. An edge labeled with $z$ from a vertex $x$ to a vertex $y$ means the constraint $I(x, z, y)$. Sometimes we do not label a vertex. This means that the corresponding variable is existentially quantified after all other variables are quantified (innermost). For instance, in Figure 1 there is one unlabeled vertex. If we call it $u$ then Figure 1 can be interpreted as the formula

$$\exists u (I(x, y, u) \land I(x, z, u) \land I(t, z, u)).$$

We define a reduction from the problem Quantified 3-Satisfiability. Suppose we have a formula $\Phi$, whose quantifier-free part is a propositional formula in 3-CNF, that is a formula of the form

$$\exists x_1 \forall y_1 \exists x_2 \forall y_2 \ldots \exists x_n \forall y_n ((x_2 \lor \overline{y}_1 \lor y_3) \land (\overline{x}_1 \lor \overline{y}_2 \lor x_5) \land \ldots) \quad (1)$$

We define a formula $\Psi$ by

$$\forall t \forall f \exists x_0 \forall x_1 \exists y_0 \forall y_1 \ldots \exists x_n \forall y_n \exists z \exists \ldots \exists (C_0 \land C_1 \land \cdots \land C_n \land F),$$

where

- $C_0$ is defined in Figure 2;
- $C_1, \ldots, C_n$ are defined in Figure 3; for the extreme cases $C_{n-1}$ and $C_n$ to be clear, they are also shown in Figures 4 and 5;
- $F$ encodes the 3-CNF and is defined as follows. Associated with each clause $(\ell_1 \lor \ell_2 \lor \ell_3)$ is a path of length 3 from $t$ to $z$ with edge labels, in sequence, $\lambda(\ell_1), \lambda(\ell_2), \lambda(\ell_3)$, where $\lambda(\ell) = u^0$, iff $\ell$ is a positive variable $u$, and $\lambda(\ell) = u^1$, iff $\ell$ is a negative variable $\overline{u}$. In Figure 6 we show $F$ for the formula (1);
- $\exists \ldots \exists$ means that all the remaining variables are existentially quantified.

Below we explain how to interpret different values of the variables:

1. variables $t$ and $f$ encode the value true and the value false;
2. variables $x_i^0$ and $x_i^1$ encode the variable $x_i$; $x_i^0 = t$ means that $x_i = 0$, $x_i^1 = t$ means that $x_i = 1$.

3. variables $y_i^0$ and $y_i^1$ encode the variable $y_i$; $y_i^0 = t$ means that $y_i = 0$, $y_i^1 = t$ means that $y_i = 1$.
Figure 3: Set of constraints $C_i$.

Figure 4: Set of constraints $C_{n-1}$.

Figure 5: Set of constraints $C_n$. 
Figure 6: Set of constraints $\mathcal{F}$.

Theorem 3. Formulas $\Phi$ and $\Psi$ are equivalent.

Lemma 4. $\Phi \rightarrow \Psi$.

Proof. In the following, we make no assumption that $t \neq f$, though the more interesting cases arise when this is true. Suppose we have a strategy for the existential player (EP) in $\Phi$. Let us define a strategy for the existential player (EP) in $\Psi$. First, we want at most one of the two values $x_i^0$ and $x_i^1$ to be equal to $t$, and at most one of $y_i^0$ and $y_i^1$ to be equal to $t$. Assume that the above rule was violated for the first time by the universal player (UP) in $\Psi$ when he plays $y_i^0 = t$ and $y_i^1 = t$, or when he plays $x_i^1 = t$ after $x_i^0 = t$. In both cases the winning strategy for the EP is to choose $x_i^0 = \cdots = x_n^0 = z = t$. Thus, we may assume that the above rule always holds.

Then we interpret $x_i^0 = t$ as $x_i = 0$, $x_i^1 = t$ as $x_i = 1$, $y_i^0 = t$ as $y_i = 0$, $y_i^1 = t$ as $y_i = 1$. It will be convenient for us to assume that $x_i = 0$ if both $x_i^0$ and $x_i^1$ are different from $t$. Similarly, we assume that $y_i = 0$ if both $y_i^0$ and $y_i^1$ are different from $t$.

Then the EP in $\Psi$ should play according to the strategy of the EP in $\Phi$, that is, if the EP in $\Phi$ chooses $x_i = 0$, then the EP in $\Psi$ should choose $x_i^0 = t$, otherwise the EP in $\Psi$ should choose $x_i^0 = f$. Additionally, the EP in $\Psi$ chooses $z = f$, and the remaining values (for the variables that are not labeled in figures) can be chosen uniquely.

It is not hard to see that the constraints in $\mathcal{F}$ are equivalent to the 3-CNF part of $\Phi$. This is because at least one of the edge labels in the path must not be evaluated to $t$, which corresponds to one of the literals being true. \hfill $\Box$

Lemma 5. $\Psi \rightarrow \Phi$.

Proof. Suppose we have a winning strategy for the EP in $\Psi$. To define a winning strategy for the EP in $\Phi$, we will show how the UP in $\Psi$ should play and how moves of the EP in $\Psi$ should be interpreted by the EP in $\Phi$. First, the UP in $\Psi$ assigns two different values to $t$ and $f$. If the EP plays
$x_i^0 = t$ then the EP in Φ plays $x_i = 0$, if the EP plays $x_i^0 \neq t$ then the EP in Φ plays $x_i = 1$. Let us show how the UP in Ψ should play. Let $d$ be the last value in the sequence $f, x^0_1, x^0_2, \ldots, x^0_i$ different from $t$. He plays $x^1_i = d$ if $x^0_i = t$, and $x^1_i = t$ if $x^0_i \neq t$. He plays $y^0_i = t$ and $y^1_i = d$ if the UP in Φ plays $y_i = 0$. He plays $y^0_i = d$ and $y^1_i = t$ if the UP in Φ plays $y_i = 1$.

Let us show that if the EP in Φ follows this strategy then all the constraints are satisfied. Let $i$ be the maximal number such that $x_i = 1$, which means that $x^0_i \neq t$. Put $d = x^0_i$. If $x_j = 0$ for every $j$, then we put $i = 0$ and $d = f$. Since $x_{i+1} = \cdots = x_n = 0$, we have $x^0_{i+1} = \cdots = x^0_n = t$ and $x^1_{i+1} = \cdots = x^1_n = d$. Notice that for every $i \in \{i, i+1, \ldots, n\}$ exactly one of the two variables $y^0_i$ and $y^1_i$ is equal to $d$. Hence, the constraints from $C_i$ imply $z \neq t$. Then the constraints from $F$ imply that the 3-CNF part holds, since, for each path from $z$ to $t$, at least one of the edge labels must not be evaluated as $t$. For instance, the path $x^0_2 - y^1_1 - y^0_3$ in Figure 6 implies that at least one of the variables $x^0_2, y^1_1, y^0_3$ does not equal $t$, which means that $x_2 = 1$, $y_1 = 0$, or $y_3 = 1$ and the constraint $(x_2 \lor \overline{y}_1 \lor y_3)$ holds.

\[ \text{Corollary 6. } QCSP(\mathbb{N}; x = y \rightarrow y = z) \text{ is PSpace-complete.} \]

Let us remark that an alternative construction of Ψ has $t$ and $f$ quantified existentially (still outermost) but adds the new constraint $t \neq f$ (note that disequality $x \neq y$ can be defined over $I$ by $\forall z \ (x, y, z)$).

4 Bounded alternation

We begin with the most substantial contribution of this section which is the Co-NP membership result.

4.1 Bounded alternation QCSP($\mathbb{N}; x = y \rightarrow u = v$) is in Co-NP

Recall that $R$ is defined by a $\Sigma_k$-formula over $\Gamma$ if

\[
R(x_1, \ldots, x_n) = \exists \cdots \exists \forall \cdots \exists \forall \cdots \forall
\]

\[
R_1(\ldots) \land \cdots \land R_s(\ldots),
\]

where the formula has $(k - 1)$ alternations and $R_1, \ldots, R_s$ are from $\Gamma$. In our case $\Gamma := (\mathbb{N}; x = y \rightarrow u = v, x = y)$ where we emphasise that atoms $x = y$ can also appear.

First, we define inductively what a $k$-proof means. Suppose $R$ is defined by a $\Sigma_1$-formula $\Phi$ over $\Gamma$, that is

\[
R(x_1, \ldots, x_n) = \exists x_{n+1} \ldots \exists x_{n+m} \; C_1 \land C_2 \land \cdots \land C_s,
\]

where $C_1, C_2, \ldots, C_s$ are constraints.
Let \( P = (e_1, e_2, \ldots, e_r) \) be a sequence of equalities on \( \{x_1, \ldots, x_{n+m}\} \), where \( e_r \) is an equality on \( \{x_1, \ldots, x_n\} \). We say that \( P \) is a \( 0 \)-proof in \( \Phi \) of the equality \( e_r \) from some set of equalities \( E \) on \( \{x_1, \ldots, x_n\} \), if for every \( i \in [r] \) the equality \( e_i \) satisfies one of the following conditions:

1. \( e_i \) is from \( E \);
2. \( e_i \) can be derived by transitivity from \( e_{i_1} \) and \( e_{i_2} \), where \( i_1, i_2 < i \);
3. \( \Phi \) contains the constraint \( e_i \);
4. \( \Phi \) contains the constraint \( (e_j \to e_i) \) for \( j < i \).

Note that we respect the commutativity of equality, in the sense that a derivation of \( x = y \) a fortiori gives also \( y = x \).

Suppose \( R \) is defined by a \( \Sigma_{2k+1} \)-formula \( \Phi \) over \( \Gamma \), that is

\[
R(x_1, \ldots, x_n) = \exists x_{n+1} \ldots \exists x_{n+m} \forall u_1 \ldots \forall u_t
\]

\[
R'(x_1, \ldots, x_{n+m}, u_1, \ldots, u_t),
\]

where \( R' \) is defined by \( \Sigma_{2k-1} \)-formula \( \Phi' \) over \( \Gamma \).

A sequence \( P = (o_1, \ldots, o_r) \) is called a \( k \)-proof in \( \Phi \) of an equality \( e_r \) on \( \{x_1, \ldots, x_n\} \) from some set of equalities \( E \) on \( \{x_1, \ldots, x_n\} \) if for every \( i \in [r] \) the following conditions hold:

- \( o_i \) is a triple \( (e_i, E_i^u, P_i^0) \);
- \( e_i \) is an equality on \( \{x_1, \ldots, x_{n+m}\} \);
- \( E_i^u \) is a set of equalities on \( \{x_1, \ldots, x_{n+m}, u_1, \ldots, u_t\} \), such that each equality is \( u_i = z \), where \( z \in \{x_1, \ldots, x_{n+m}, u_1, \ldots, u_{i-1}\} \), and each variable \( u_i \) appears at most once as the left variable of the equality;
- \( P_i^0 \) is a \( (k-1) \)-proof in \( \Phi' \) of the equality \( e_i \) from the set of equalities \( E \cup E_i^u \cup \{e_1, \ldots, e_{i-1}\} \).

Similarly, we define a \( k \)-proof of a contradiction. A sequence \( P = (o_1, \ldots, o_r) \) is called a \( k \)-proof in \( \Phi \) of a contradiction from some set of equalities \( E \) on \( \{x_1, \ldots, x_n\} \) if for every \( i \in [r-1] \) the following conditions hold:

- \( o_i \) is a triple \( (e_i, E_i^u, P_i^0) \);
- \( e_i \) is an equality on \( \{x_1, \ldots, x_{n+m}\} \);
- \( E_i^u \) is a set of equalities on \( \{x_1, \ldots, x_{n+m}, u_1, \ldots, u_t\} \), such that each equality is \( u_i = z \), where \( z \in \{x_1, \ldots, x_{n+m}, u_1, \ldots, u_{i-1}\} \), and each variable \( u_i \) appears at most once as the left variable of the equality;
• $P^0_i$ is a $(k - 1)$-proof in $\Phi'$ of the equality $e_i$ from the set of equalities $E \cup E_i^u \cup \{e_1, \ldots, e_{i-1}\}$.

and $o_r$ satisfies one of the following conditions:

1. $o_r = (\emptyset, E_r^u, P^0_r)$, where $E_r^u$ is defined as before, and $P^0_r$ is a $(k - 1)$-proof in $\Phi'$ of a contradiction from the set of equalities $E \cup E_r^u \cup \{e_1, \ldots, e_{i-1}\}$;

2. $o_r = (e_r, E_r^u, P^0_r)$, where $e_r, E_r^u, P^0_r$ are defined as above but $e_r$ is an equality $u_i = z$, where $z \in \{x_1, \ldots, x_{n+m}, u_1, \ldots, u_{i-1}\}$ and $E_r^u$ does not contain the equality $u_i = z'$ for any $z'$.

Note that there is no such a notion as 0-proof of a contradiction. We need to prove the following lemmas about the soundness of our proofs.

**Lemma 7.** Suppose $R(x_1, \ldots, x_n)$ is defined by a $\Sigma_{2k+1}$-formula $\Phi$ over $\Gamma$, $P$ is a $k$-proof in $\Phi$ of an equality $e$ on $\{x_1, \ldots, x_n\}$ from some set of equalities $E$ on $\{x_1, \ldots, x_n\}$. Then any tuple from $R$ that satisfies all the equalities from $E$ also satisfies $e$.

**Proof.** We proceed by induction on $k$. At $k = 0$, the property holds from the soundness of the steps in the definition of 0-proof. Suppose it holds at $k = m$, then it holds at $k = m + 1$ by the soundness of the steps in the definition of $(m + 1)$-proof. Note that each step $(e_i, E_i^u, P_i^0)$ is sound because all of the equalities in $E_i^u$ can be simultaneously realised by the appropriate evaluation of $u_1, \ldots, u_t$.

**Lemma 8.** Suppose $R(x_1, \ldots, x_n)$ is defined by a $\Sigma_{2k+1}$-formula $\Phi$ over $\Gamma$, $P$ is a $k$-proof in $\Phi$ of a contradiction from some set of equalities $E$ on $\{x_1, \ldots, x_n\}$. Then $R$ does not contain tuples satisfying all the equalities from $E$.

**Proof.** We proceed by induction on $k$. At $k = 1$, the property holds from the soundness of the steps in the definition of 1-contradiction. In particular, if $e_r$ is of the form $u_i = z$, where $E_r^u$ holds no information on $u_i$, then assigning $u_i$ to be anything other than (the value of) $z$ demonstrates that the formula must be false (a contradiction).

Suppose it holds at $k = m$, then it holds at $k = m + 1$ by the soundness of the steps in the definition of $(m + 1)$-contradiction (as described in the previous paragraph).

We now need a technical lemma showing that we have transitivity not only in 0-proofs but also in $k$-proofs.

**Lemma 9.** Suppose $R(x_1, \ldots, x_n)$ is defined by a $\Sigma_{2k+1}$-formula $\Phi$ over $\Gamma$, then there exists a proof of the equality $x_{i_1} = x_{i_3}$ from the set of equalities $\{x_{i_1} = x_{i_2}, x_{i_2} = x_{i_3}\}$.
Proof. We prove by induction on $k$. For $k = 0$ it follows from the definition. For larger $k$ take a one step proof $(x_i = x_i, \emptyset, P^0_1)$, where $P^0_1$ is a $(k - 1)$-proof of transitivity, which exists by the inductive assumption. 

We now give our principal completeness theorem.

**Theorem 10.** Suppose $R(x_1, \ldots, x_n)$ is defined by a $\Sigma_{2k+1}$-formula $\Phi$ over $\Gamma := (\mathbb{N}; x = y \rightarrow u = v, x = y)$, $R(a_1, \ldots, a_n)$ does not hold, and $E$ is the set of all equalities on $\{x_1, \ldots, x_n\}$ satisfied by the tuple $(a_1, \ldots, a_n)$. Then one of the following conditions holds:

- there exists a $k$-proof in $\Phi$ of some equality $x_i = x_j$ from the set of equalities $E$ that is not satisfied by $(a_1, \ldots, a_n)$, i.e. $a_i \neq a_j$;
- there exists a $k$-proof in $\Phi$ of a contradiction from the set of equalities $E$.

**Proof.** We proceed by induction on $k$.

**Base.** Suppose $k = 0$, then

$$R(x_1, \ldots, x_n) = \exists x_{n+1} \ldots \exists x_{n+m} C_1 \land C_2 \land \cdots \land C_s,$$

where $C_1, C_2, \ldots, C_s$ are constraints. We put in the sequence $P$ all the equalities we can derive. If we can derive an equality on $\{x_1, \ldots, x_n\}$ that is not from $E$, then this equality is not satisfied by $(a_1, \ldots, a_n)$, which completes the proof. Assume that we did not derive such an equality. We need to prove that $R$ holds on $(a_1, \ldots, a_n)$ to get a contradiction. Since we can use transitivity, all the variables in $\{x_1, \ldots, x_{n+m}\}$ are divided into equivalence classes such that each equality in $P$ is inside one class. Note that an equivalence class cannot contain two variables $x_i$ and $x_j$ such that $a_i \neq a_j$.

Then we send the equivalence class to the value $a_i$ if it contains a variable $x_i$ for $i \in [n]$, and we choose a new value for the equivalence class if all the variables in it are from $\{x_{n+1}, \ldots, x_{n+m}\}$. We claim that this assignment satisfies all the constraints of $\Phi$. All the constraints $x_i = y \rightarrow u = v$ are obviously satisfied. A constraint $x = y \rightarrow u = v$ is satisfied because either $x$ and $y$ are from different equivalence classes, or the equality $u = v$ should have been added to $P$, and $u, v$ are from one equivalence class.

**Inductive step.** Suppose $R$ is defined by

$$R(x_1, \ldots, x_n) = \exists x_{n+1} \ldots \exists x_{n+m} \forall u_1 \ldots \forall u_t,$$

$$R'(x_1, \ldots, x_{n+m}, u_1, \ldots, u_t),$$

where $R'$ is defined by $\Sigma_{2k-1}$-formula $\Phi'$ over $\Gamma$. Again, we add to the sequence $P$ of all the equalities on $\{x_1, \ldots, x_{n+m}\}$ we can derive using $(k-1)$-proofs. By Lemma 9 we still have transitivity, which means that all the variables in $\{x_1, \ldots, x_{n+m}\}$ are divided into equivalence classes. Again if an equivalence class contains two variables $x_i, x_j$ such that $a_i \neq a_j$, then the
first condition of the theorem holds, which finishes the proof. Then we send all the variables in the equivalence class to \(a_i\) if it contains \(x_i\) for \(i \in [n]\), otherwise we send all the variables of this class to a new value that never appeared. Thus we get an assignment \((x_1, \ldots, x_{n+m}) = (b_1, \ldots, b_{n+m})\). We claim that this assignment satisfies

\[
\forall u_1 \ldots \forall u_t \ R'(x_1, \ldots, x_{n+m}, u_1, \ldots, u_t).
\]

Assume that it does not, then for some evaluations of \((u_1, \ldots, u_t) = (c_1, \ldots, c_t)\) the tuple \((b_1, \ldots, b_{n+m}, c_1, \ldots, c_t)\) is not in the relation \(R'\). By \(E^u\) we denote the set of equalities of the form \(u_i = z\) that are chosen as follows. If \(c_i\) appears in the tuple \((b_1, \ldots, b_{n+m}, c_1, \ldots, c_{i-1})\) then we add the equality \(u_i = z\), where \(z\) is the variable corresponding to the first appearance of \(c_i\) in the tuple. By the inductive assumption applied to \(R'\) we have one of the following cases (here we cheated a bit because we didn’t add all the equalities satisfied by the tuple \((b_1, \ldots, b_{n+m}, c_1, \ldots, c_{i-1})\) to \(E^u\) but they can be derived by transitivity).

Case 1. There is a \((k−1)\)-proof of a contradiction, which gives us a \(k\)-proof of a contradiction.

Case 2. There is a \((k−1)\)-proof of some equality \(z = z'\) that is violated by our assignment. Since we have transitivity, we may choose variables \(z\) and \(z'\) so that variables that appear earlier in the list \(x_1, \ldots, x_{n+m}, u_1, \ldots, u_t\) do not have the same assignment as \(z\) and \(z'\) respectively. Consider two cases.

Case 2A. The equality \(z = z'\) is on the set \(\{x_1, \ldots, x_{n+m}\}\). This means that this equality connects two different equivalence classes of variables and should have been added to \(P\). Contradiction.

Case 2B. The equality is of the form \(u_i = z'\), where \(z' \in \{x_1, \ldots, x_{n+m}, u_1, \ldots, u_{i-1}\}\) (we can always switch \(z\) and \(z'\)). This gives us a \(k\)-proof of a contradiction (see item 2 of the definition).

It remains to evaluate the size of proof. We assume that we never repeat the equalities \(e_i\) because it doesn’t make any sense. So, we forbid such proofs.

**Lemma 11.** Suppose \(R\) is defined by a \(\Sigma_{2k+1}\)-formula \(\Phi\) over \(\Gamma := (\mathbb{N}; x = y \rightarrow u = v, x = y)\), \(\Phi\) contains \(\ell\) different variables, and \(P\) is a \(k\)-proof of an equality or a contradiction, then the length of \(P\) is smaller than \(10^{k+1} \ell^{2k^2+3}\).

**Proof.** We denote the upper bound on the length of a \(k\)-proof for a formula with \(\ell\) variables by \(L_\ell(k)\). We prove by induction on \(k\). For \(k = 0\) we just list equalities, we can have at most \(\ell^2\) equalities, thus to write the proof we need \(\ell^2 \cdot (2 \log_2 \ell + 3) \leq 10\ell^3\) symbols.

For larger \(k\) we again have at most \(\ell^2\) equalities, but here we also define
$E_i^u$ and a $(k-1)$-proof for each $i$. Thus we have

$$L_\ell(k) \leq \ell^2((2 \log_2 \ell + 3) + \ell^2 \cdot (2 \log_2 \ell + 3) + L_\ell(k-1))$$

$$\leq \ell^2((2 \log_2 \ell + 3) + \ell^2 \cdot (2 \log_2 \ell + 3) + 10^k \ell^{2k+1})$$

$$\leq 10^{k+1} \ell^{2k+3}$$

\[\blacksquare\]

**Theorem 12.** $\Sigma_{2k+1}$-QCSP($\mathbb{N}$; $x = y \rightarrow u = v$) is in Co-NP.

**Proof.** The sentence defines a null-ary relation $R$, which is either true or false. Since the sentence doesn’t have free variables, if it is false then by Theorem 10 there should be a $k$-proof of a contradiction. By Lemma 11, the size of this proof is of polynomial size in the size of the sentence. It is clear that the proof can be verified in polynomial time. Hence, whenever the sentence doesn’t hold we have a polynomial-size proof of a contradiction which puts the problem in Co-NP. \[\blacksquare\]

### 4.2 Transformation of a quantified conjunctive formula to an equivalent $\Pi_2$-formula.

Let us commence with a trivial observation about equality languages.

**Lemma 13.** If $\Gamma$ is an equality language and $\phi$ is an input for QCSP($\Gamma$) on $n$ variables, then $\phi$ is a yes-instance iff it is a yes-instance when all variable quantification is relativised to the finite subset $[n]$.

**Proof.** The simplest way to see this is through the game between Universal and Existential. As the game progresses through the quantifier order, we update some bijection on the domain of $\Gamma$, to assume, without loss of generality, that all variables are evaluated within $[n]$. The bijection begins as the identity and is updated whenever an element is played outside of $[n]$ by swapping that element with an unplayed element within $[n]$. \[\blacksquare\]

It is well-known on finite domains that any property that is definable in quantified conjunctive logic is already definable by a $\Pi_2$ formula of quantified conjunctive logic (this was first noted explicitly in [15]). Zhuk gave another proof of this in [26], which we reproduce here mildly modified so as to work also for the case equality languages.

Suppose $\Psi = \exists y_1 \forall x_1 \exists y_2 \forall x_2 \ldots \exists y_n \forall x_n \Phi$. We define a transformation $\zeta$ that transforms $\Psi$ into an equivalent $\Pi_2$-formula. Note that $\zeta(\Psi)$ is of exponential size in the size of $\Psi$.

For each tuple $(a_1, \ldots, a_n) \in [2n]^n$ the formula $\Phi_{a_1, \ldots, a_n}$ is obtained from $\Phi$ by

1. replacement of each variable $y_i$ by $y_i^{a_1 \ldots a_{i-1}}$ (at $i = 1$ keep variable as $y_i$)
2. replacement of each variable $x_i$ by $x_i^{a_1,\ldots,a_i}$

By $\zeta(\Psi)$ we denote the sentence such that

1. its quantifier-free part is \( \bigwedge_{(a_1,\ldots,a_n) \in \{2^n\}} \Phi_{a_1,\ldots,a_n} \)

2. first we universally quantify the $x$-variables, then we existentially quantify the $y$-variables.

Let us show that $\Psi$ and $\zeta(\Psi)$ are equivalent.

**Lemma 14.** $\Psi \rightarrow \zeta(\Psi)$.

**Proof.** Suppose $(f_1(), x_1, f_2(x_1), x_2, f_3(x_1, x_2), x_3, \ldots)$ is a solution of $\Psi$. Let us show that a solution of $\zeta(\Psi)$ can be defined by

\[ y_{a_1,\ldots,a_{i-1}} = f_i(x_{a_1}^{a_1}, x_{a_2}^{a_2}, x_{a_3}^{a_3}, \ldots, x_{a_{i-1}}^{a_{i-1}}). \]

In fact, each conjunctive formula $\Phi_{a_1,\ldots,a_n}$ holds as it is just $\Phi$ with all the variables renamed. $\square$

**Lemma 15.** $\zeta(\Psi) \rightarrow \Psi$.

**Proof.** Owing to Lemma 14 we may assume that all variable quantification of $\Psi$ is relativised to the set $\{2^n\}$. Set $x_{a_1,\ldots,a_i} = a_i$ for every $i$ and $a_1,\ldots,a_i \in \{2^n\}$. Since $\zeta(\Psi)$ holds, for each assignment of the $x$-variables there exists a proper assignment of the $y$-variables. We define a solution to $\Psi$ by $(f_1(), x_1, f_2(x_1), x_2, f_3(x_1, x_2), x_3, \ldots)$, where $f_i(a_1,\ldots,a_{i-1}) = y_{a_1,\ldots,a_{i-1}}$. To check that the conjunctive formula $\Phi$ holds for $(x_1,\ldots,x_n) = (a_1,\ldots,a_n)$ we consider $\Phi_{a_1,\ldots,a_n}$. $\square$

Thus, we have proved the following.

**Corollary 16.** Let $\Gamma$ be an equality language. Any property that is definable on $\Gamma$ in quantified conjunctive logic is already definable by a $\Pi_2$ formula of quantified conjunctive logic.

### 4.3 The remaining cases

Let us examine the proof of Theorem 9.1 in [4] which proves that the $\Sigma_3$-QCSP($\mathbb{N}; x = y \rightarrow y = z$) is Co-NP-hard. The reduction is from the complement of monotone 3-SAT and the constructed instances of $\Sigma_3$-QCSP($\mathbb{N}; x = y \rightarrow y = z$) have quantifier prefix $\exists^2\forall^*\exists^*$. Indeed, the first two existential variables are $b_0$ and $b_1$ and are constrained to be distinct. Suppose they are instead universally quantified, then the proof still works as we shall now see.

**Theorem 17** ([4]). $\Pi_2$-QCSP($\mathbb{N}; x = y \rightarrow y = z$) is Co-NP-hard.
\textbf{Proof.} We reproduce the construction from [4] with a small amendment as to how we treat the special variables $b_0$ and $b_1$. We reduce from the complement of the problem Monotone 3-Satisfiability (MON-3-SAT). In this version of 3-satisfiability all clauses are either purely negative or purely positive. Let $\phi$ be an instance of MON-3-SAT with variables $v_1, \ldots, v_n$, $l$ negative clauses and $m$ positive clauses. We build an instance $\psi$ of $\Pi_2$-QCSP($\mathbb{N}; x = y \rightarrow y = z$) with variables $v_1, \ldots, v_n$, $N_1, \ldots, N_l, N'_l$ and $P_1, \ldots, P_m, P'_2, \ldots, P'_m$, and two additional variables $b_0$ and $b_1$. The instance has quantifier prefix

$$\forall b_0 \forall b_1 \forall v_1, \ldots, v_n \exists N_1, \ldots, N_l, N'_l, P_1, \ldots, P_m, P'_2, \ldots, P'_m.$$ 

The quantifier-free part of $\psi$ is the conjunction of the following:

1. $(v_i = b_0 \rightarrow b_0 = N_h) \land (v_j = b_0 \rightarrow b_0 = N_h) \land (v_k = b_0 \rightarrow b_0 = N_h)$ for each negative clause $N_h = (\neg v_i \lor \neg v_j \lor \neg v_k)$.

2. $(v_i = b_1 \rightarrow b_1 = P_h) \land (v_j = b_1 \rightarrow b_1 = P_h) \land (v_k = b_1 \rightarrow b_1 = P_h)$ for each positive clause $P_h = (v_i \lor v_j \lor v_k)$.

3. $(N_1 = N_2 \rightarrow N_2 = N'_2) \land \bigwedge_{h=3,\ldots,l}(N'_h = N_h \rightarrow N_h = N'_h)$.

4. $(P_1 = P_2 \rightarrow P_2 = P'_2) \land \bigwedge_{h=3,\ldots,m}(P'_{h-1} = P_h \rightarrow P_h = P'_h)$.

5. $N'_l = P'_m$.

(We assume that the formula $\phi$ contains at least two positive clauses and two negative clauses.)

($\phi$ is a yes-instance of MON-3-SAT implies $\psi$ is a no-instance of our QCSP.) Let us assume that $b_0$ and $b_1$ are evaluated to different elements. Assume that the original formula is satisfiable with some assignment $f : \{ v_1, \ldots, v_n \} \rightarrow \{ 0, 1 \}$. Consider the induced assignment to the remaining universally quantified variables when $v_i$ is set to $b_{f(v_i)}$. Since $f$ satisfies all of the clauses, it must holds that $N_h = b_0$ for all $h \in [l]$ and $P_h = b_1$ for all $h \in [m]$. It must further hold that $N'_h = b_0$ for all $h \in [2, \ldots, l]$ and that $P'_h = b_1$ for all $h \in [2, \ldots, m]$. But this implies that $N'_l \neq P'_m$ and we have that the formula $\psi$ is false.

($\phi$ is a no-instance of MON-3-SAT implies $\psi$ is a yes-instance of our QCSP.) Now consider that $\phi$ is false. If $b_0$ and $b_1$ are evaluated to the same element, then $\psi$ will be true. Let us now assume that they are evaluated to different elements. Consider some assignment $g : \{ v_1, \ldots, v_n \} \rightarrow \{ 0, 1 \}$. Let us say that a negative clause $N_h$ is not satisfied by $g$ if none of its variables is set by $g$ to be $b_0$; similarly, let us say that a positive clause $P_h$ is not satisfied by $g$ is none of its variables is set by $g$ to be $b_1$. By assumption there exists a negative clause not satisfied by $g$ or a positive clause not satisfied by $g$. Let us discuss the negative case as the positive case is dual. If it is a negative clause $N_h$, then all constraints in (1) above can be satisfied without setting
It follows that the constraints in (3) can be satisfied with $N'_h$ set to $b_1$. Now setting all of the variables $P_h$ and $P'_h$ to $b_1$ results in all the constraints being satisfied. It follows that the formula $\psi$ is true. \qed

**Corollary 18.** Suppose that $x = y \rightarrow y = z$ is definable in an equality language $\Gamma$ in a $\Pi_2$ formula of quantified conjunctive logic. Then, $\Pi_2$-$QCSP(\Gamma)$ is Co-NP-hard.

**Proof.** Let us follow the proof of Theorem \[79\]. It is clear to see that we can plug in $\Pi_2$ definitions of $x = y \rightarrow y = z$ innermost and then commute the universal quantifiers across the conjunction to obtain an instance that is at worst $\Pi_4$. Let us now argue that we can draw all the existential quantifiers, over the $N_h, N'_h, P_h, P'_h$, innermost to result in an instance that is $\Pi_2$. We need to check that in the case that $\phi$ is a yes-instance of MON-3-SAT, that this still implies $\psi$ is a no-instance of our QCSP, even when the existential quantifiers are further in (which could advantage Existential). But indeed this is true because they have been forced by the outermost universal quantifiers. \qed

A cursory examination of the proof of Corollary \[6\] where one starts with a universal sentence of Quantified-3-Satisfiability, shows that it produces a $\Pi_2$ instance of QCSP($\mathbb{N}; x = y \lor u = v$). However, the universal fragment of Quantified-3-Satisfiability is in P (since the quantifier free part is in 3-CNF).

We now develop some lemmas before addressing the case of maximal complexity, that is, rising in the polynomial hierarchy.

**Lemma 19.** $(P_1(x_1, x'_1) \lor \ldots \lor P_m(x_m, x'_m))$, where each $P_i$ asserts either equality or disequality, is pp-definable in $(\mathbb{N}; \neq, x = y \lor u = v)$

**Proof.** \exists v'(x = y \lor u = v') \land v \neq v' \text{ pp-defines } (x = y \lor u \neq v). \text{ Then the desired pp-definition goes as follows.}

$$\exists y_1, \ldots, y_{m+1} \ (P_1(x_1, x'_1) \lor y_1 = y_2) \land (P_2(x_2, x'_2) \lor y_2 = y_3) \land \ldots \land (P_m(x_m, x'_m) \lor y_m = y_{m+1}) \land y_1 \neq y_{m+1} \text{ pp-defines } (x = y \lor u = v).$$ \qed

**Lemma 20.** The $\Pi_k$-$QCSP(\mathbb{N}; \neq, x = y \lor u = v)$ is $\Pi^P_k$-hard.

**Proof.** We will give a reduction from $\Pi_k$-Quantified-Not-All-Equal-3-SAT ($\Pi_k$-QNAE-3-SAT). Let $I$ be an instance of $\Pi_k$-QNAE-3-SAT. Each variable $v$ in $I$ becomes a pair of variables $v, v'$ in $\Phi$, an instance of $\Pi_k$-$QCSP(\mathbb{N}; \neq$
, x = y ∨ u = v), in which v = v′ encodes that the variable is true and v ≠ v′ encodes that the variable is false.

Now we can define the not-all-equal-3-sat predicate in disjunctive normal form (DNF) as:

\[
\begin{align*}
(x = x' \land y = y' \land z ≠ z') \lor \\
(x = x' \land y ≠ y' \land z = z') \lor \\
(x = x' \land y ≠ y' \land z ≠ z') \lor \\
(x ≠ x' \land y = y' \land z = z') \lor \\
(x ≠ x' \land y ≠ y' \land z ≠ z') \lor \\
(x ≠ x' \land y ≠ y' \land z = z')
\end{align*}
\]

This expands in CNF to a set of clauses each with six literals that would be susceptible to Lemma 19. It is this conjunction that defines our clause gadget. It is the conjunction of all the clause gadgets that define the quantifier-free part of Φ. The quantification from I translates simply, ∀v becomes ∀v, v′ and ∃v becomes ∃v, v′.

(I in Π_k-QNAE-3-SAT implies Φ in Π_k-QCSP(\mathbb{N}; ≠, x = y ∨ u = v).) The winning strategy for Existential on Φ is to choose some x, x′ identical in \mathbb{N}, if x had been chosen true in I; and some x, x′ distinct in \mathbb{N}, if x had been chosen false in I. The universal variables similarly translate back from Φ into I.

(Φ in Π_k-QCSP(\mathbb{N}; ≠, x = y ∨ u = v) implies I in Π_k-QNAE-3-SAT.) The winning strategy for Existential on I is to choose x true, if x, x′ had been chosen identical in I; and some x false, if x, x′ had been chosen identical in I. The universal variables similarly translate back from I into Φ.

Lemma 21. CSP(\mathbb{N}; 0, 1, x = y ∨ y = z) is NP-hard.

Proof. Consider the 2-element (Boolean) language ∆ on domain \{0, 1\} with relations ≠ and x = y ∨ y = z. Since ≠ is not preserved by a semilattice (∧ or ∨) or constant operation, and x = y ∨ y = z is not preserved by majority or minority, it follows from Schaefer Theorem (see the modern presentation of Theorem 3.20 in [15]) that CSP(∆) is NP-hard. We will reduce from CSP(∆) to CSP(\mathbb{N}; 0, 1, x = y ∨ y = z) to prove our result. We can define in (\mathbb{N}; 0, 1, x = y ∨ y = z) the subdomain \{0, 1\} with y = 0 ∨ y = 1. We can define on \{0, 1\} the binary disequality relation by expanding (x = 0 ∧ y = 1) ∨ (x = 1 ∧ y = 0) to

\[
(x = 0 ∨ x = 1) ∧ (x = 0 ∨ y = 0) ∧ (y = 1 ∨ x = 1) ∧ (y = 1 ∨ y = 0).
\]

Finally, we can define x = y ∨ y = z by itself. Thus, we reduce an instance ϕ of CSP(∆) to an instance ψ of CSP(\mathbb{N}; 0, 1, x = y ∨ y = z) in the following fashion. The set of variables stays the same and we replace each relation in ϕ by its definition in (\mathbb{N}; 0, 1, x = y ∨ y = z). Finally, for each variable v, we add the constraint v = 0 ∨ v = 1. That ϕ is a yes-instance iff ψ is a yes-instance follows immediately by construction. □

18
Corollary 22. $\Pi_2$-QCSP($\mathbb{N}; x = y \vee y = z$) is NP-hard.

Proof. We reduce from CSP($\mathbb{N}; 0, 1, x = y \vee y = z$) by adding two new variables $b_0$ and $b_1$ universally quantified outermost and then substituting all instances of 0 with $b_0$ and all instances of 1 with $b_1$. Note that, when $b_0$ and $b_1$ are evaluated as equal, the remainder of the sentence will always be true. 

Lemma 23. Let $\Gamma$ be an equality language that is positive and not negative. There exists a finite set of constants $[m]$ so that there is a pp-definition in $(\Gamma; 1, \ldots, m)$ of $x = y \vee y = z$ over $\mathbb{N} \setminus [m]$.

Proof. By Lemma 20 and Corollary 15, there is a $\Pi_2$ quantified conjunctive formula $\theta$ that defines $x = y \vee y = z$ over $\Gamma$. Suppose it has $m$ universal variables. Obtain $\theta'$ from $\theta$ by substituting, in turn, the $m$ universal variables with the constants $1, \ldots, m$. Since $\Gamma$ is positive, $\theta'$ defines $x = y \vee y = z$ on the set $\mathbb{N} \setminus [m]$ (Universal’s optimal strategy on $\theta$ will always involve playing new elements).

Corollary 24. Let $\Gamma$ be a equality language that is positive and not negative. $\Pi_2$-QCSP($\Gamma$) is NP-hard.

Proof. We use the reduction from Lemma 21: rather as we did in Corollary 22 but now we have $m + 2$ outermost universally quantified variables. Two of them will play the role of 0 and 1 from Lemma 21 and then $m$ further of them will play the role of the constants $[m]$ in Lemma 23. Note that because $\Gamma$ is a positive language, an optimal strategy for Universal will always be to play these variables on distinct elements of $\mathbb{N}$, whereupon the correctness of the simulation of Lemma 21 is guaranteed by Lemma 23.

4.4 The classification

Lemma 25. If $\Gamma$ is an equality language containing some relation that does not have a Horn definition, then the relation $x = y \vee u = v$ is definable in $\Gamma$ by a quantified conjunctive formula.

Proof. If $\Gamma$ is preserved by an essential operation with infinite image $f$, then by Proposition 37 in [2] $f$ preserves $\neq$ or $\Gamma$ is preserved by all operations. In the latter case every relation in $\Gamma$ may be defined by a conjunction of equalities. Hence it is in particular Horn. By Proposition 43 in [2], $\Gamma$ is Horn also in the case where $f$ preserves $\neq$. It contradicts the assumption and implies that $\Gamma$ is preserved by essentially unary operations only. In particular, the surjective polymorphisms of $\Gamma$ preserve the relation $x = y \vee u = v$. But then Lemma 4.1 in [2] gives a quantified conjunctive definition.
Lemma 26. If $\Gamma$ is an equality language that is positive but not negative, then the relation $x = y \lor u = v$ is definable in $\Gamma$ by a quantified conjunctive formula.

Proof. We follow a line of reasoning similar to that in the proof of Theorem 7.1 in [4], except that the relation to be defined is $x = y \lor u = v$ not $x = y \lor y = z$. If $Pol(\Gamma)$ contains an essential operation $f$ with infinite image, then as in the proof of Lemma 25, we show that $\Gamma$ is Horn, and hence negative. It contradicts the assumption. Thus, all surjective polymorphisms of $\Gamma$ are essentially unary. Since $x = y \lor u = v$ is preserved by all essentially unary operation, the lemma follows by Lemma 4.1 in [4].

Lemma 27. If $\Gamma$ is an equality language that is not positive, then the relation $\neq$ is quantified conjunctive definable in $\Gamma$.

Proof. If $sPol(\Gamma)$ has an operation that violates $\neq$, then Theorems 7.3 and 8.2 from [4] imply that $\Gamma$ is positive. The contrapositive of this yields the result according to [4].

The following is Proposition 43 in [5].

Lemma 28. All Horn relations are pp-definable in $x = y \rightarrow u = v$ and $\neq$.

We are now in a position to prove Theorem 2.

Proof of Theorem 2. The Logspace cases come a fortiori from [4], since $QCSP(\Gamma)$ is in Logspace in this case.

If $\Gamma$ is positive but not negative, then $\Pi_k$-$QCSP(\Gamma)$ is in NP by [4]. Moreover, $\Pi_2$-$QCSP(\Gamma)$ is NP-hard by Corollary 21.

If $\Gamma$ is Horn, but not negative, then $\Pi_k$-$QCSP(\Gamma)$ is in Co-NP by Theorem 12 together with Lemma 28. Moreover, $\Pi_2$-$QCSP(\Gamma)$ is Co-NP-hard by Corollary 15.

Finally, if $\Gamma$ is not Horn and not positive, then, by Lemmas 23 and 27 $\neq$ and $x = y \lor y = z$ are quantified conjunctive definable, and we know from Corollary 16 even by a $\Pi_2$ formula. It follows from Lemma 20 that $\Pi_k$-$QCSP(\Gamma)$ is $\Pi_{k-2}$-hard (since the innermost conjunction of $\Pi_2$ formulas may itself be turned into a single $\Pi_2$ formula by commuting conjunction and universal quantification).

5 Conclusions

In the case where $\Gamma$ is not Horn and not positive, we proved that $\Pi_k$-$QCSP(\Gamma)$ is $\Pi_{k-2}$-hard. It is likely that this can be improved to $\Pi_k^p$-hard. However, neither Lemmas 23 and 27 can be improved from quantified conjunctive definability to pp-definability. Respective counterexamples are $x = y \rightarrow y = z$ and $x = y \lor y = z \lor x = z$. 20
References

[1] Barto, L., Kompatscher, M., Olsák, M., Pham, T. V., and Pinsker, M. The equivalence of two dichotomy conjectures for infinite domain constraint satisfaction problems. In 32nd Annual ACM/IEEE Symposium on Logic in Computer Science, LICS 2017, Reykjavik, Iceland, June 20-23, 2017 (2017), IEEE Computer Society, pp. 1–12.

[2] Barto, L., and Pinsker, M. Topology is irrelevant (in a dichotomy conjecture for infinite domain constraint satisfaction problems). SIAM J. Comput. 49, 2 (2020), 365–393.

[3] Bodirsky, M., and Chen, H. Quantified equality constraints. In 22nd IEEE Symposium on Logic in Computer Science (LICS 2007), 10-12 July 2007, Wroclaw, Poland, Proceedings (2007), IEEE Computer Society, pp. 203–212.

[4] Bodirsky, M., and Chen, H. Quantified equality constraints. SIAM J. Comput. 39, 8 (2010), 3682–3699.

[5] Bodirsky, M., Chen, H., and Pinsker, M. The reducts of equality up to primitive positive interdefinability. J. Symb. Log. 75, 4 (2010), 1249–1292.

[6] Bodirsky, M., and Kára, J. The complexity of equality constraint languages. Theory Comput. Syst. 43, 2 (2008), 136–158.

[7] Bodirsky, M., and Kára, J. The complexity of temporal constraint satisfaction problems. J. ACM 57, 2 (2010), 9:1–9:41.

[8] Bodirsky, M., Martin, B., and Mottet, A. Discrete temporal constraint satisfaction problems. J. ACM 65, 2 (2018), 9:1–9:41.

[9] Bodirsky, M., and Mottet, A. A dichotomy for first-order reducts of unary structures. Log. Methods Comput. Sci. 14, 2 (2018).

[10] Bodirsky, M., and Nesetril, J. Constraint satisfaction with countable homogeneous templates. J. Log. Comput. 16, 3 (2006), 359–373.

[11] Bodirsky, M., and Pinsker, M. Schaefer’s theorem for graphs. J. ACM 62, 3 (2015), 19:1–19:52.

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

[13] Charatonik, W., and Wróna, M. Quantified positive temporal constraints. In Computer Science Logic, 22nd International Workshop,
[14] Charatonik, W., and Wrona, M. Tractable quantified constraint satisfaction problems over positive temporal templates. In Logic for Programming, Artificial Intelligence, and Reasoning, 15th International Conference, LPAR 2008, Doha, Qatar, November 22-27, 2008. Proceedings (2008), I. Cervesato, H. Veith, and A. Voronkov, Eds., vol. 5330 of Lecture Notes in Computer Science, Springer, pp. 543–557.

[15] Chen, H. A rendezvous of logic, complexity, and algebra. SIGACT News 37, 4 (2006), 85–114.

[16] Chen, H. Existentially restricted quantified constraint satisfaction. Inf. Comput. 207, 3 (2009), 369–388.

[17] Chen, H. Three concrete complexity questions about constraints, in search of theories. Invited talk at Arbeitstagung Allgemeine Algebra (Workshop on General Algebra), AAA98, Dresden, Germany, July 2019.

[18] Chen, H., and Müller, M. An algebraic preservation theorem for aleph-zero categorical quantified constraint satisfaction. Log. Methods Comput. Sci. 9, 1 (2012).

[19] Chen, H., and Wrona, M. Guarded ord-horn: A tractable fragment of quantified constraint satisfaction. In 19th International Symposium on Temporal Representation and Reasoning, TIME 2012, Leicester, United Kingdom, September 12-14, 2012 (2012), B. C. Moszkowski, M. Reynolds, and P. Terenziani, Eds., IEEE Computer Society, pp. 99–106.

[20] Eiter, T., and Gottlob, G. The complexity of logic-based abduction. J. ACM 42, 1 (1995), 3–42.

[21] Hodges, W. Model Theory. Encyclopedia of Mathematics and its Applications. Cambridge University Press, 1993.

[22] Krokhin, A. A., Jeavons, P., and Jonsson, P. Reasoning about temporal relations: The tractable subalgebras of allen’s interval algebra. J. ACM 50, 5 (2003), 591–640.

[23] Wrona, M. Tractability frontier for dually-closed ord-horn quantified constraint satisfaction problems. In Mathematical Foundations of Computer Science 2014 - 39th International Symposium, MFCS 2014,
[24] Zhuk, D. A proof of CSP dichotomy conjecture. In 2017 IEEE 58th Annual Symposium on Foundations of Computer Science (FOCS) (Oct 2017), pp. 331–342.

[25] Zhuk, D. A proof of the CSP dichotomy conjecture. J. ACM 67, 5 (2020), 30:1–30:78.

[26] Zhuk, D. The complexity of the quantified CSP having the polynomially generated powers property. CoRR abs/2110.09504 (2021).

[27] Zhuk, D., and Martin, B. QCSP monsters and the demise of the chen conjecture. In Proceedings of the 52nd Annual ACM SIGACT Symposium on Theory of Computing (2020), pp. 91–104.