Identifying Tractable Quantified Temporal Constraints Within Ord-Horn

Jakub Rydval
Technische Universität Wien, Austria

Žaneta Semanišinová
Technische Universität Dresden, Germany

Michał Wrona
Jagiellonian University, Kraków, Poland

Abstract

The constraint satisfaction problem, parameterized by a relational structure, provides a general framework for expressing computational decision problems. Already the restriction to the class of all finite structures forms an interesting microcosm on its own, but to express decision problems in temporal reasoning one has to take a step beyond the finite-domain realm. An important class of templates used in this context are temporal structures, i.e., structures over $\mathbb{Q}$ whose relations are first-order definable using the usual countable dense linear order without endpoints.

In the standard setting, which allows only existential quantification over input variables, the complexity of finite and temporal constraints has been fully classified. In the quantified setting, i.e., when one also allows universal quantifiers, there is only a handful of partial classification results and many concrete cases of unknown complexity. This paper presents a significant progress towards understanding the complexity of the quantified constraint satisfaction problem for temporal structures. We provide a complexity dichotomy for quantified constraints over the Ord-Horn fragment, which played an important role in understanding the complexity of constraints both over temporal structures and in Allen's interval algebra. We show that all problems under consideration are in P or coNP-hard. In particular, we determine the complexity of the quantified constraint satisfaction problem for $(\mathbb{Q}; x = y \Rightarrow x \geq z)$, hereby settling a question open for more than ten years.

2012 ACM Subject Classification Theory of computation → Design and analysis of algorithms; Theory of computation → Logic; Theory of computation → Computational complexity and cryptography

Keywords and phrases constraint satisfaction problems, quantifiers, dichotomy, temporal reasoning, Ord-Horn

Digital Object Identifier 10.4230/LIPIcs.ICALP.2024.151

Category Track B: Automata, Logic, Semantics, and Theory of Programming

Related Version *Full Version*: `https://arxiv.org/abs/2402.09187` [17]

Funding *Jakub Rydval*: This research was funded in whole or in part by the Austrian Science Fund (FWF) [I 5948]. For the purpose of Open Access, the authors have applied a CC BY public copyright licence to any Author Accepted Manuscript (AAM) version arising from this submission.
Žaneta Semanišinová: The author has been funded by the European Research Council (Project POCOCOP, ERC Synergy Grant 101071674) and by the DFG (Project FinHom, Grant 467967530). Views and opinions expressed are however those of the authors only and do not necessarily reflect those of the European Union or the European Research Council Executive Agency. Neither the European Union nor the granting authority can be held responsible for them.
Michał Wrona: The author is partially supported by National Science Centre, Poland grant number 2020/37/B/ST6/01179.

Acknowledgements The authors thank Dmitriy Zhuk for many inspiring discussions on the topic, and the anonymous reviewers for many helpful suggestions.

1 Introduction

The constraint satisfaction problem (CSP) of a structure $\mathfrak{B}$ in a finite relational signature τ, denoted by CSP($\mathfrak{B}$), is the problem of deciding whether a given primitive positive τ-sentence holds in $\mathfrak{B}$. The class of all *finite-domain* CSPs, i.e., where $\mathfrak{B}$ can be chosen finite, famously constitutes a large fragment of NP that admits a dichotomy between P and NP-completeness [19]. Quantified constraint satisfaction problems (QCSPs) generalize CSPs by allowing both existential *and* universal quantification over input variables. The complexity of such problems is much less understood already for finite structures, the state of the art being a complexity classification for QCSPs of finite structures with all unary relations and three-element structures with all singleton unary relations [21]. For infinite structures, the investigations essentially follow the CSP programme, which was initiated by the study of the CSPs of structures over $\mathbb{N}$ (or $\mathbb{Q}$) whose relations are definable by Boolean combinations of equalities and disequalities, the so-called *equality structures* [6]. The full complexity classification for quantified equality constraints was completed quite recently [22], by resolving the long-standing question of determining the complexity of QCSP($\mathbb{Q}$; D), where

$$\mathrm{D} := \{(x, y, z) \in \mathbb{Q}^3 \mid x = y \Rightarrow x = z\}.$$

This question was left open in [3], where all the remaining results have been provided. The next in line are *temporal structures*, which are structures with domain $\mathbb{Q}$ whose relations are first-order definable over $\{<\}$, where $<$ interprets as the usual unbounded dense linear order. The relations of such structures are called *temporal*.

By definition, temporal structures form a richer class than equality structures. While the complexity of temporal CSPs has been classified more than a decade ago [7], there is only a handful of partial classification results regarding the complexity of temporal QCSPs [4, 10, 11, 12, 18]. Yet, already from this limited amount of available data it is apparent that the majority of the pathological cases is concentrated in the *Ord-Horn* (OH) fragment, we elaborate on this below. The OH fragment comprises all temporal structures whose relations are definable by an OH formula, i.e., a conjunction of clauses of the form

$$(x_1 \neq y_1 \vee \cdots \vee x_k \neq y_k \vee x_{k+1} \geq y_{k+1}) \tag{1}$$

for $k \geq 0$, where the last disjunct is optional and some variables might be identified [2].

1.1 Ord-Horn

OH was first introduced and used by Nebel and Bürckert to describe a maximally tractable constraint language containing all basic relations on Allen's interval algebra [16]. For a full classification of maximally tractable subalgebras of Allen, see [15]. In the context of CSPs over temporal structures, OH is not even a maximally tractable language as it is properly contained in two of the nine maximally tractable fragments characterized by the eight binary operations *min*, *max*, *mx*, *dual mx*, *mi*, *dual mi*, $\ell\ell$, *dual* $\ell\ell$, and a constant operation [7]. The *dual* of an operation f on $\mathbb{Q}$ is the operation $(x_1, \dots, x_n) \mapsto -f(-x_1, \dots, -x_n)$, e.g., *max* is the dual of *min*. The description of maximally tractable languages by operations is typical for the so-called algebraic approach to constraint satisfaction problems. For the sake of the reader unfamiliar with this approach, we simply refer to a maximally tractable temporal language characterized by an operation *op* as the *op* fragment and refrain from defining the operations. For example, we write "the *min* fragment" or "the *max* fragment." The question which of the nine fragments are also maximal w.r.t. tractability of the QCSP was investigated in [4], and answered positively in the first four cases. The answer is negative in the last three cases [22], and the question remains open for *mi* and *dual mi* fragments. In the intersections of $\ell\ell$, *mi* and *dual mi* fragments lie the OH structures $(\mathbb{Q}; \mathrm{M}^+)$ and $(\mathbb{Q}; \mathrm{M}^-)$, respectively, where

$$\mathrm{M}^+ := \{(x, y, z) \in \mathbb{Q}^3 \mid x = y \Rightarrow x \geq z\} \quad \text{and} \quad \mathrm{M}^- := \{(x, y, z) \in \mathbb{Q}^3 \mid x = y \Rightarrow x \leq z\}.$$

Determining the complexity of $\mathrm{QCSP}(\mathbb{Q}; \mathrm{M}^+)$ was posed as an open question in [4]; it could have been anywhere between PTIME and PSPACE. Note that its counterpart $\mathrm{QCSP}(\mathbb{Q}; \mathrm{M}^-)$ is essentially the same problem with the order reversed.

Apart from temporal structures preserved by a constant operation, OH captures precisely those temporal structures whose CSP is solvable by local consistency checking [8]. This well-known generic preprocessing algorithm can be formulated for any CSP satisfying some reasonable structural assumptions [5], and thus OH constraints are fairly well understood from the CSP perspective. However, the analysis of OH constraints in the quantified setting requires a surprisingly large amount of creativity. As a simple example, already $\mathrm{QCSP}(\mathbb{Q}; R)$ for the OH relation R defined by $(x_1 \neq x_2 \vee x_3 \geq x_4) \wedge \phi$ is in PTIME if ϕ equals $(x_3 \geq x_1) \wedge (x_1 \geq x_3) \wedge (x_3 \neq x_4)$ [12], coNP-complete if ϕ equals $(\bigwedge_{i,j \in \{1,2\}} x_i \neq x_{j+2})$ [20], and PSPACE-complete if ϕ is the empty conjunction [22].

The class of *Guarded Ord-Horn* (GOH) formulas [12] is defined inductively. In the base case we are allowed to take OH formulas of the form $(x \leq y)$, $(x_1 \neq y_1 \vee \cdots \vee x_k \neq y_k)$, or $(x \neq x_1 \vee \cdots \vee x \neq x_k) \vee (x < y) \vee (y \neq y_1 \vee \cdots \vee y \neq y_\ell)$. In the induction step we can form formulas of the form $\psi_1 \wedge \psi_2$ or $(x_1 \leq y_1 \vee \cdots \vee x_k \leq y_k) \wedge (x_1 \neq y_1 \vee \cdots \vee x_k \neq y_k \vee \psi)$, where ψ, ψ_1, ψ_2 are GOH formulas. Thus, newly added disequalities are guarded by atomic $\{\leq\}$-formulas. A GOH structure may only contain temporal relations definable by GOH formulas. Observe that the tractable template from the previous paragraph is GOH.

▶ **Theorem 1** ([12]). *Let $\mathfrak{B}$ be a GOH structure. Then $\mathrm{QCSP}(\mathfrak{B})$ is in* PTIME*.*

The tractability result from [12] is conceptually simple and based on pebble games generalizing local consistency methods. At the same time, all quantified OH constraints outside of GOH are coNP-hard or admit a LOGSPACE reduction from $\mathrm{QCSP}(\mathbb{Q}; \mathrm{M}^+)$.

▶ **Theorem 2** ([18]). *Let $\mathfrak{B}$ be an OH structure. Then one of the following holds.*

- *$\mathfrak{B}$ is GOH.*
- *$\mathrm{QCSP}(\mathfrak{B})$ is coNP-hard.*
- *$\mathfrak{B}$ primitively positively defines M^+ or M^-.*

There was a prospect that $\mathrm{QCSP}(\mathbb{Q}; \mathrm{M}^+)$ would be PSPACE-hard, because the PSPACE-hardness proof from [22] for $\mathrm{QCSP}(\mathbb{Q}; \mathrm{D})$, when adjusted appropriately, almost yields a proof of PSPACE-hardness for this QCSP. In that case, Theorems 1 and 2 would immediately yield a dichotomy between P and coNP-hardness for quantified OH constraints. However, it turns out that the situation is more complicated, as we explain below.

1.2 Contributions

On the one hand, we prove tractability for $\mathrm{QCSP}(\mathbb{Q}; \mathrm{M}^+)$, and thereby provide a positive answer to an open question from [4]. This is the main technical contribution of the present paper, and the proof stretches over the entirety of Section 3.

In a certain sense, the presented algorithm generalizes local consistency methods. We iteratively expand a given instance Φ of $\mathrm{QCSP}(\mathfrak{B})$ by constraints associated to relations whose arity is bounded by the size of Φ and which have short primitive positive definitions in $\mathfrak{B}$, until a fixed-point is reached. The condition for the expansion by these constraints is tested using an oracle for $\mathrm{CSP}(\mathfrak{B}, <)$. The algorithm is thus not very far from the well-known framework of Datalog with existential rules [1, 9].

▶ **Theorem 3.** $\mathrm{QCSP}(\mathbb{Q}; \mathrm{M}^+)$ *is in* PTIME*.*

Our tractability result naturally extends to the QCSPs of those OH structures which can be expressed in $(\mathbb{Q}; \mathrm{M}^+, \neq)$ using primitive positive definitions (see Proposition 6). We show that the set of these structures coincides with the intersection of the OH fragment with the $\pi\pi$ fragment. Here by $\pi\pi$ [1] we refer to the "*projection-projection*" operation from [7], which played an important role in identifying the maximally tractable temporal CSP languages covered by *min*, *mi* and *mx*. In the present paper, we introduce the $\pi\pi$ fragment using the syntactic description obtained in [4]. For a definition of the operation $\pi\pi$, see [7]. The $\pi\pi$ fragment consists of all temporal relations definable by a conjunction of clauses of the form

$$(x \neq y_1 \vee \cdots \vee x \neq y_k \vee x \geq z_1 \vee \cdots \vee x \geq z_\ell) \tag{2}$$

for $k, l \geq 0$. The *dual* $\pi\pi$ fragment is obtained by replacing every instance of $\geq$ in (2) by $\leq$.

▶ **Corollary 4.** $\mathrm{QCSP}(\mathfrak{B})$ *is in* PTIME *if* $\mathfrak{B}$ *is an OH structure in which every relation is definable by a conjunction of clauses of the form*

$$(x \neq y_1 \vee \cdots \vee x \neq y_k \vee x \geq z) \tag{3}$$

for $k \geq 0$ *and where the last disjunct* $(x \geq z)$ *may be omitted. The above condition is satisfied if and only if* $\mathfrak{B}$ *is contained in the intersection of the OH fragment and the* $\pi\pi$ *fragment.*

On the other hand, we confirm that $\mathrm{QCSP}(\mathbb{Q}; \mathrm{M}^+)$ indeed walks a very fine line between tractability and hardness. We show that, if M^+ is combined with any OH relation R that is not contained in the $\pi\pi$ fragment, then the resulting QCSP becomes coNP-hard, even if $\mathrm{QCSP}(\mathbb{Q}; R)$ is tractable. Intuitively, either $(\mathbb{Q}; \mathrm{M}^+, R)$ already primitively positively defines D and we use the PSPACE-hardnees proof from [22] directly, or we replace each constraint of the form $\mathrm{D}(x, y, z)$ in the proof by $\mathrm{M}^+(x, y, z) \wedge \mathrm{M}^+(z, z, x)$. The latter, however, is not entirely conditional, and certain issues arise due to the transitivity of $\leq$. These issues can be partially (but not entirely) resolved using constraints associated to

$$\check{\mathrm{Z}} \coloneqq \{(x_1, y_1, x_2, y_2) \in \mathbb{Q}^4 \mid (x_1 \neq y_1 \vee x_2 \neq y_2) \wedge (y_1 < y_2)\},$$

which is quantified primitively positively definable in $(\mathbb{Q}; \mathrm{M}^+, R)$, ultimately leaving us with a proof of coNP-hardness.

By a careful combination of syntactic pruning arguments, Theorem 2, and a new coNP-hardness proof inspired by the PSPACE-hardness proof from [22], we prove coNP-hardness in all cases for which tractability does not follow from Theorem 1, Corollary 4 or its analogue for *dual* $\pi\pi$, i.e., where $\geq$ is replaced with $\leq$ in (3). This leads to the following dichotomy for quantified OH constraints.

▶ **Theorem 5.** *Let* $\mathfrak{B}$ *be an OH structure. Then* $\mathrm{QCSP}(\mathfrak{B})$ *is solvable in polynomial time if* $\mathfrak{B}$ *is GOH, contained in the* $\pi\pi$ *fragment, or in the dual* $\pi\pi$ *fragment. Otherwise,* $\mathrm{QCSP}(\mathfrak{B})$ *is* coNP*-hard.*

[1] In contrast to previous literature on temporal CSPs, we deviate from the notation pp from [7] that clashes with the shortcut for "primitive positive" and use $\pi\pi$ instead.

We believe that the methods used in this paper will also prove useful in identifying the complexity of quantified temporal constraints outside of OH, e.g., in the case of *mi* or $\pi\pi$. Omitted proofs can be found in the long version of the article available on arXiv [17], where we also provide more details on the algebraic approach and relevant operations on $\mathbb{Q}$.

2 Preliminaries

2.1 First-order structures

The set $\{1, \ldots, n\}$ is denoted by $[n]$. In the present paper, we consider structures $\mathfrak{A} = (A; R_1, \ldots, R_k)$ over a finite relational signature τ. For the sake of simplicity, we often use the same symbol R for both the relation $R^{\mathfrak{A}}$ and the relational symbol R. An *expansion* of $\mathfrak{A}$ is a σ-structure $\mathfrak{B}$ with $A = B$ such that $\tau \subseteq \sigma$ and $R^{\mathfrak{B}} = R^{\mathfrak{A}}$ for each $R \in \tau$. We write $(\mathfrak{A}, R)$ for the expansion of $\mathfrak{A}$ by the relation R over A.

We assume that the reader is familiar with classical first-order logic; we allow the first-order formulas $x = y$ and $\bot$ (the nullary falsity predicate). Let T be a set of first-order τ-sentences over a common signature τ and ϕ, ψ τ-formulas whose free variables are among $\bar{x}$. We say that ϕ *entails* ψ w.r.t. T if $\mathfrak{A} \models \forall \bar{x}(\phi \Rightarrow \psi)$ holds for all models $\mathfrak{A}$ of T. We do not explicitly mention T if it is clear from the context, e.g., the theory of linear orders. A first-order τ-formula ϕ is *primitive positive* (pp) if it is of the form $\exists x_1, \ldots, x_m(\phi_1 \wedge \cdots \wedge \phi_n)$, where each ϕ_i is *atomic*, i.e., of the form $\bot$, $(x_i = x_j)$, or $R(x_{i_1}, \ldots, x_{i_\ell})$ for some $R \in \tau$. *Quantified primitive positive* (qpp) formulas generalize pp-formulas by allowing both existential *and* universal quantification. If ϕ and ψ are (q)pp-formulas, then $\phi \wedge \psi$ can be rewritten into an equivalent (q)pp-formula, so we treat such formulas as (q)pp-formulas as well.

The *(quantified) constraint satisfaction problem* for a structure $\mathfrak{B}$, denoted by (Q)CSP($\mathfrak{B}$), is the computational problem of deciding whether a given (q)pp τ-sentence holds in $\mathfrak{B}$. By *constraints*, we refer to the conjuncts in the quantifier-free part of a given (Q)CSP instance of (Q)CSP($\mathfrak{B}$). In the QCSP framework, we usually think of an instance as a game between two players: an *existential player* (EP) and a *universal player* (UP) who assign values to the existentially and universally quantified variables, respectively. To every moment of the game we associate a partial function $[\![\cdot]\!]$ from the variables into the domain of the parametrizing structure describing values assigned to the variables by either of the players. The instance is true if and only if the EP has a winning strategy in this game, i.e., can respond to all moves of the UP while keeping all constraints satisfied. Otherwise, the instance is false and the UP has a winning strategy, i.e., can violate a constraint regardless of the moves of the EP.

If $\mathfrak{A}$ is a τ-structure and $\phi(x_1, \ldots, x_n)$ is a τ-formula with free variables $x_1, \ldots, x_n$, then the relation $\{(a_1, \ldots, a_n) \in A^n \mid \mathfrak{A} \models \phi(a_1 \ldots, a_n)\}$ is the *relation defined by* ϕ *in* $\mathfrak{A}$, and denoted by $\phi^{\mathfrak{A}}$. Let S be a set of τ-formulas. We say that a relation R has a S*-definition* in $\mathfrak{A}$, or that $\mathfrak{A}$ S*-defines* R, if R equals $\phi^{\mathfrak{A}}$ for some $\phi \in S$. For instance, S can be the set of all quantifier-free or primitive positive formulas over τ. We might also say that a relation R S-defines another relation R' if the structure $(A; R)$ S-defines R'. The next proposition is folklore in the constraint satisfaction literature.

▶ **Proposition 6** ([2, 3]). *Let* $\mathfrak{A}, \mathfrak{B}$ *be structures with the same domain. If every relation of* $\mathfrak{B}$ *is (q)pp-definable in* $\mathfrak{A}$*, then* (Q)CSP($\mathfrak{B}$) *reduces to* (Q)CSP($\mathfrak{A}$) *in* LOGSPACE*.*

2.2 Temporal structures

Since $(\mathbb{Q}; <)$ has quantifier-elimination [13], every temporal relation is in fact quantifier-free-definable in $(\mathbb{Q}; <)$. We may further assume that every quantifier-free definition is in *conjunctive normal form* (CNF). We might sometimes refer to temporal relations directly using their CNF-definitions. Also, it will sometimes be convenient to work with formulas over the structure $(\mathbb{Q}; \leq, \neq)$ instead of the structure $(\mathbb{Q}; <)$, e.g., in the definitions of OH or $\pi\pi$ from the introduction. The following lemma is folklore.

▶ **Lemma 7** ([2]). *The OH and the (dual) $\pi\pi$ fragments are closed under expansions by pp-definable relations.*

From the syntactic descriptions (1) and (2) it is apparent that the fragments OH and $\pi\pi$ are incomparable. Their intersection consists of all temporal relations definable by a conjunction of clauses of the form (3). This can be shown using the following lemma.

▶ **Lemma 8.** *Let R be an OH relation defined by a quantifier-free formula ϕ in CNF over the signature $\{\leq, \neq\}$ containing a clause $\psi_1 \vee \psi_2$, where ψ_1 is equivalent to $(x \geq z_1 \vee \cdots \vee x \geq z_\ell)$ for some variables x and $z_1, \ldots, z_\ell$. Then we may replace ψ_1 in ϕ by $(x \geq z_i)$ for some $i \in [\ell]$ so that the resulting formula still defines R.*

In the present article, the intersection of OH and the (*dual*) $\pi\pi$ fragment is the sole source of all newly identified tractable QCSPs. It is convenient to work with a finite relational basis. Recall the relations M^+ and M^- from the introduction. By Lemma 8 and Lemma 9 below, a temporal relation is OH and contained in the $\pi\pi$ fragment if and only if it is pp-definable in $(\mathbb{Q}; \mathrm{M}^+, \neq)$. An analogous statement holds for *dual* $\pi\pi$ and $(\mathbb{Q}; \mathrm{M}^-, \neq)$.

▶ **Lemma 9.** *The $(k+2)$-ary temporal relation defined by (3) has the pp-definition*

$$\exists h_1, \ldots, h_{k+1}\Big((x = h_1) \wedge \big(\bigwedge\nolimits_{i\in[k]} \mathrm{M}^+(h_i, h_i, x) \wedge \mathrm{M}^+(h_i, y_i, h_{i+1})\big) \wedge (h_{k+1} = z)\Big).$$

Note that the length of the above pp-definition is linear in k, this will be relevant later in the proof of Theorem 3.

3 $\mathrm{QCSP}(\mathbb{Q}; \mathrm{M}^+)$ is in PTIME

In this section, we prove that $\mathrm{QCSP}(\mathbb{Q}; \mathrm{M}^+)$ can be solved in polynomial time using Algorithm 1. In the formulation of the algorithm, we view instances of $\mathrm{QCSP}(\mathbb{Q}; \mathrm{M}^+)$ as sentences over $\{\geq, \neq\}$ in prenex normal form whose quantifier-free part is in CNF.

We first need to fix some terminology. For the remainder of Section 3, Φ always denotes an arbitrary or explicitly specified instance of $\mathrm{QCSP}(\mathbb{Q}; \mathrm{M}^+)$, ϕ its quantifier-free part, and V the set if its variables. Furthermore, we denote the universal variables by $\mathrm{V}_\forall$ and the existential variables by $\mathrm{V}_\exists$. Let $\prec$ be the linear order on all variables of Φ in which they appear in the quantifier prefix of Φ. When we write $A \prec B$ for $A, B \subseteq V$, we mean $x \prec y$ for all $x \in A, y \in B$. In particular, this condition is trivially true if one of the two sets is empty.

▶ **Definition 10.** *For $x, z \in \mathrm{V}$, we define*
- *$x \equiv z$ if and only if x, z refer to the same variable,*
- *$x \preceq z$ if and only if $x \equiv z$ or $x \prec z$,*
- *$x \preceq_\forall z$ if and only if $x \equiv z$, or $x \prec z$ and $z \in \mathrm{V}_\forall$.*

For $u \in V$ and $A \subseteq V$, we define
- *$\uparrow_u \coloneqq \{y \in \mathrm{V}_\forall \mid u \preceq y\}$,*
- *$\uparrow_A \coloneqq \bigcup_{u\in A} \uparrow_u$ (recall that the empty union is empty).*

Note that the three binary relations in Definition 10 are transitive.

▶ **Definition 11.** *For every pair $x, z \in \mathrm{V}$, we define x-z-cut $\coloneqq \{u \in \mathrm{V}_\forall \mid \mathrm{V}_\exists \cap \{x, z\} \prec u\} \setminus \{z\}$.*

Observe that the definition of the x-z-cut depends on how x and z are quantified. The idea is that x-z-cut represents the universal variables that the UP can always make equal to x to trigger the condition $(x \geq z)$ via an entailed constraint of the form $\big((\bigwedge_{v\in A} x = v) \Rightarrow x \geq z\big)$. Since the UP has full control over the values of these variables with respect to x and z, they can be removed from the clauses added in the second last line in Algorithm 1.

Note that, by Lemma 9, the constraints added by Algorithm 1 correspond to relations which have pp-definitions in $(\mathbb{Q}; \mathrm{M}^+)$ of length linear in their arity. This means that satisfiability in Algorithm 1 can be tested using an oracle for $\mathrm{CSP}(\mathbb{Q}; \mathrm{M}^+, <)$, because we can simply replace each constraint by its pp-definition in $\mathrm{CSP}(\mathbb{Q}; \mathrm{M}^+)$ while only changing the size of Φ by a polynomial factor.

Algorithm 1 An algorithm for $\mathrm{QCSP}(\mathbb{Q}; \mathrm{M}^+)$.

Input: An instance Φ of $\mathrm{QCSP}(\mathbb{Q}; \mathrm{M}^+)$ with the quantifier-free part ϕ
Output: *true* or *false*
while *ϕ changes* **do**
  for $x, z, u \in \mathrm{V}$ **do**
    if *ϕ contains the clause $(x \geq z)$ or $(z \geq x)$, where $x \prec z$ and $z \in \mathrm{V}_\forall$* **then**
      return *false*;
    if *$\phi \wedge (\bigwedge_{v\in\uparrow_u\setminus\{x,z\}} x = v) \wedge (x < z)$ is unsatisfiable* **then**
      expand ϕ by the clause $\big((\bigwedge_{v\in\uparrow_u\setminus(\{x,z\}\cup x\text{-}z\text{-cut})} x = v) \Rightarrow x \geq z\big)$;
return *true*;

▶ **Example 12.** Consider the instance Φ of $\mathrm{QCSP}(\mathbb{Q}; \mathrm{M}^+)$ defined by

$$\exists x_1 \forall x_2 \exists x_3 \forall x_4 \exists x_5 \big((x_1 = x_2 \Rightarrow x_1 \geq x_5) \wedge (x_3 = x_2 \Rightarrow x_3 \geq x_4) \\ \wedge\ (x_5 = x_4 \Rightarrow x_5 \geq x_3) \wedge (x_3 \geq x_1) \wedge (x_5 \geq x_1)\big).$$

We claim that Algorithm 1 derives $(x_1 \geq x_4)$, and thereby rejects on Φ. We first observe that the formula $\phi \wedge (\bigwedge_{v\in\uparrow_u\setminus\{x_1,x_4\}} x_1 = v) \wedge (x_1 < x_4)$ is satisfiable for every $u \in \{x_1, \dots, x_5\}$. Since $x_3, x_5 \in \mathrm{V}_\exists$, it is enough to show that $\phi \wedge (x_1 = x_2) \wedge (x_1 < x_4)$ is satisfiable, which is witnessed by any assignment satisfying $(x_5 = x_1 = x_2 < x_3 < x_4)$. On the other hand, $\phi \wedge (\bigwedge_{v\in\uparrow_{x_2}\setminus\{x_1,x_3\}} x_1 = v) \wedge (x_1 < x_3)$ is not satisfiable. Therefore, the algorithm expands ϕ by $(x_1 = x_2 \Rightarrow x_1 \geq x_3)$, because $x_4 \in x_1$-x_3-cut. But now $\phi \wedge (\bigwedge_{v\in\uparrow_{x_2}\setminus\{x_1,x_4\}} x_1 = v) \wedge (x_1 < x_4)$ is not satisfiable anymore. Since $x_2 \in x_1$-x_4-cut, the algorithm expands ϕ by $(x_1 \geq x_4)$.

As mentioned below Definition 11, Algorithm 1 rejects correctly because all constraints added during the run of the algorithm are entailed by Φ, see Lemma 13.

▶ **Lemma 13.** *Suppose that Algorithm 1 derives from Φ a constraint ψ. Then Φ is true if and only if Φ expanded by ψ is true.*

Proof. Denote by Ψ and Ψ' the sentences obtained from Φ by replacing ϕ with $\phi \wedge \psi$ and $\phi \wedge \psi'$, respectively, where

$$\psi := \Big(\bigvee_{v\in\uparrow_u\setminus(\{x,z\}\cup x\text{-}z\text{-cut})} x \neq v\Big) \vee (x \geq z) \quad \text{and} \quad \psi' := \Big(\bigvee_{v\in\uparrow_u\setminus\{x,z\}} x \neq v\Big) \vee (x \geq z).$$

Since $\phi \wedge \neg\psi'$ is unsatisfiable, we have that ϕ entails ψ'. It follows that Φ is true iff Ψ' is true. To complete the proof, we have to show that if Ψ' is true, then Ψ is true. We prove the contraposition and assume that the UP has a winning strategy on Ψ. If the UP wins on Ψ by falsifying any clause different from ψ, then the very same choices lead the UP to falsifying the same clause in Ψ'. Otherwise, the UP falsifies ψ while playing on Ψ. Then the UP can play in the same way on Ψ' when it comes to the variables that occur both in ψ and ψ' and set all variables in x-z-cut to the same value as x. Note that this is possible since either x is universal or x precedes all variables in x-z-cut. It remains to show that ψ' is falsified. Clearly, all $\{\neq\}$-disjuncts are falsified. Since z is either universal or precedes all variables in x-z-cut, the disjunct $(x \geq z)$ is falsified as well, because it is falsified in ψ. ◀

Example 14 showcases how a winning strategy of the UP, obtainable implicitly from Lemma 13, might in fact be uniquely determined.

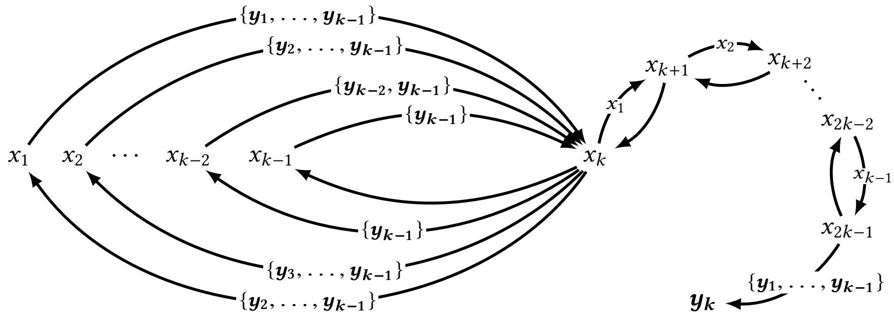

Figure 1 The quantifier-free part of Φ from Example 14.

▶ **Example 14.** Consider the instance $\Phi \coloneqq \exists x_1 \forall y_1 \exists x_2 \forall y_2 \cdots \exists x_k \forall y_k \exists x_{k+1} \ldots \exists x_{2k-1}\ \phi$ with ϕ described by Figure 1, where an edge from x to z labeled with y stands for $\mathrm{M}^+(x, y, z)$. An edge from x to z labeled with some subset A of the universal variables stands for a constraint of the form $\big((\bigwedge_{v\in A} x = v) \Rightarrow x \geq z\big)$ already derived by Algorithm 1. Using Lemma 9, these edges can be appropriately replaced with pp-definitions, and thus Φ is well-defined.

We claim that the UP has the unique winning strategy on Φ of playing $[\![y_i]\!]$ equal to an arbitrary number $> [\![x_i]\!]$ if $i = k$ and $[\![x_1]\!] = \cdots = [\![x_k]\!]$, and equal to $\mathit{min}\{[\![x_1]\!], \ldots, [\![x_i]\!]\}$ otherwise. We start by showing that this is a winning strategy.

Suppose, on the contrary, that there exists an assignment $[\![\cdot]\!] \colon \mathrm{V} \to \mathbb{Q}$ of values to the variables witnessing that the EP has a counter-strategy to the strategy of the UP from above. First, consider the case where $[\![x_1]\!], \ldots, [\![x_k]\!]$ are not all equal. Suppose that $[\![x_k]\!] = \mathit{min}\{[\![x_1]\!], \ldots, [\![x_k]\!]\}$ and let $j \in [k]$ be the largest index such that $[\![x_j]\!] > [\![x_k]\!]$. Recall that the algorithm already derived the constraint $\psi_1 \coloneqq \big((\bigwedge_{v\in\{y_{j+1},\ldots,y_{k-1}\}} x_k = v) \Rightarrow x_k \geq x_j\big)$ on Φ. By the strategy of the UP, we have $[\![y_{j+1}]\!] = \cdots = [\![y_{k-1}]\!] = [\![x_{j+1}]\!] = [\![x_k]\!]$. But then ψ_1 is clearly not satisfied by $[\![\cdot]\!]$, a contradiction. Suppose now that $[\![x_k]\!] > \mathit{min}\{[\![x_1]\!], \ldots, [\![x_k]\!]\}$. Let $j \in [k]$ be the largest index such that $[\![x_j]\!] = \mathit{min}\{[\![x_1]\!], \ldots, [\![x_k]\!]\}$. Recall that the algorithm already derived the constraint $\psi_2 \coloneqq \big((\bigwedge_{v\in\{y_j,\ldots,y_{k-1}\}} x_j = v) \Rightarrow x_j \geq x_k\big)$. By the strategy of the UP, we have $[\![y_j]\!] = \cdots = [\![y_{k-1}]\!] = [\![x_j]\!] < [\![x_k]\!]$. But then ψ_2 is clearly not satisfied by $[\![\cdot]\!]$, a contradiction. We conclude that $[\![x_1]\!] = \cdots = [\![x_k]\!]$. In this case, the UP played $[\![y_k]\!] > [\![x_k]\!]$. Since $[\![\cdot]\!]$ is a satisfying assignment, we must have $[\![x_k]\!] = [\![x_{k+1}]\!] = \cdots = [\![x_{2k-1}]\!]$. But then $[\![\cdot]\!]$ does not satisfy $\big((\bigwedge_{v\in\{y_1,\ldots,y_{k-1}\}} x_{2k-1} = v) \Rightarrow x_{2k-1} \geq y_k\big)$ because $[\![y_k]\!] > [\![x_k]\!] = [\![x_{2k-1}]\!] = [\![y_1]\!] = \cdots = [\![y_{k-1}]\!]$, a contradiction. We conclude that the strategy of the UP from above is a winning strategy.

The strategy of the UP is unique in the sense that, no matter what values the EP played for $x_1, \ldots, x_i$, if the UP deviates from his strategy at y_i, then the EP wins by playing $[\![x_{i+1}]\!] = \cdots = [\![x_{2k-1}]\!]$ equal to an arbitrary number $> \mathit{max}([\![x_i]\!], [\![y_i]\!])$ if $[\![x_i]\!] \neq [\![y_i]\!]$ and equal to $\mathit{min}\{[\![x_1]\!], \ldots, [\![x_i]\!]\}$ otherwise.

Table 1 The inference rules of the proof system $\mathcal{P}$. Here I stands for "initialize," S for "simplify," T for "transitivity," R for "reject," A for "alternative transitivity," and C for "constraint".

§I	$\mathcal{P}(x,x;\emptyset)$:− $x\in V$
§S	$\mathcal{P}(x,z;A\setminus x\text{-}z\text{-cut})$:− $\mathcal{P}(x,z;A)$
§T	$\mathcal{P}(x,z;A)$:− $\mathcal{P}(x,y;A)\wedge\mathcal{P}(y,z;\emptyset)$
§R	$\bot$:− $\begin{cases}1.\ \mathcal{P}(x,z;\emptyset)\\ 2.\ x\prec z \text{ and } z\in \mathrm{V}_\forall\text{, or } z\prec x \text{ and } x\in\mathrm{V}_\forall\end{cases}$
§A	$\mathcal{P}(x_i,z;A\cup B\cup(\{x_1,x_2\}\setminus\{x_i\}))$:− $\begin{cases}1.\ \mathcal{P}(x_1,y;A)\wedge\mathcal{P}(y,x_2;\emptyset)\wedge\mathcal{P}(y,z;B)\\ 2.\ (\{x_1,x_2\}\setminus\{x_i\})\subseteq\mathrm{V}_\forall\ (i\in\{1,2\})\end{cases}$
§C	$\mathcal{P}(x_i,z;A\cup B\cup(\{x_1,x_2,x_3,x_4\}\setminus\{x_i\}))$:− $\begin{cases}1.\ \mathcal{P}(x_1,u;A)\wedge\mathcal{P}(u,x_2;\emptyset)\\ 2.\ \mathcal{P}(x_3,v;B)\wedge\mathcal{P}(v,x_4;\emptyset)\\ 3.\ (\{x_1,x_2,x_3,x_4\}\setminus\{x_i\})\subseteq\mathrm{V}_\forall\ (i\in\{1,2,3,4\})\\ 4.\ (u{=}v\Rightarrow u\geq z) \text{ or } (v{=}u\Rightarrow v\geq z) \text{ in } \phi\end{cases}$

Finally, we show that the algorithm derives *false*. In the first run of the main loop, we get $(x_{i+1} \geq x_i)$ for all $i \in \{1, \dots, k-2\}$. Assuming previously derived constraints, we get $(x_k \geq x_i)$ for all $i \in \{1, \dots, k-1\}$ (purely by transitivity). Now it is possible to derive $(x_i = y_i \Rightarrow x_i \geq x_{i+1})$ for all $i \in \{1, \dots, k-1\}$. In the final step, we get $(x_1 \geq y_k)$, again, simply by invoking an oracle for $\mathrm{CSP}(\mathbb{Q}; \mathrm{M}^+, <)$, which makes the algorithm reject.

3.1 False instances

The goal of this subsection is to reformulate the condition for rejection by Algorithm 1 within a certain proof system $\mathcal{P}$ operating on Φ, whose rules are given in Table 1 using a Datalog-style syntax. The proof system syntactically derives predicates of the form $\mathcal{P}(x, y; A)$ with $x, y \in V$ and $A \subseteq \mathrm{V}_\forall$ (on the left hand side of $:-$) from other predicates of this form derived earlier and the information encoded in Φ (on the right hand side of $:-$).

We shall now provide some intuition behind the formulation of the proof system. The idea is that an expression $\mathcal{P}(x, z; A)$ should capture a constraint $\big((\bigwedge_{v\in A} x = v) \Rightarrow x \geq z\big)$ entailed by Φ, where A only consists of universal variables. Assuming the adopted semantics, §T and §A just describe natural properties of such expressions, and §S and §R witness consequences of the quantification over the variables. The combination of §I and §C captures precisely the situations where the UP can indirectly enforce the identification of two (potentially existential) variables within a constraint in ϕ. In particular, it can be used to introduce $\mathcal{P}(u, z; \{v\})$ for conjuncts $(u = v \Rightarrow v \geq z)$ in ϕ with v universally quantified as follows. The proof system first derives $\mathcal{P}(u, u; \emptyset)$ and $\mathcal{P}(v, v; \emptyset)$ using §I. Then it uses §C to derive $\mathcal{P}(u, z; \{v\})$ by identifying x_1, x_2 with u and x_3, x_4 with v.

The reader might naturally ask why we cannot obtain a polynomial-time algorithm by just closing Φ under the rules of the proof system with a suitable form of fixed-point semantics. The reason is that, already under the least fixed-point semantics, the proof system might derive exponentially many expressions of the form $\mathcal{P}(x, z; A)$. Such a situation occurs, e.g., in Example 18 and in the case of the constraint paths in ϕ as defined in the proof of Lemma 25.

The precise connection between the proof system and Algorithm 1 is captured by Lemma 15. Note that Lemma 15 in particular implies that Algorithm 1 rejects whenever the proof system derives $\bot$. When combined with Lemma 13 and Lemma 19 (proved later in Section 3.2), we get that this is in fact the only situation in which Algorithm 1 rejects. Lemma 15 can be proved by a straightforward induction on the length of the derivation sequences within $\mathcal{P}$.

▶ **Lemma 15.** *Suppose that $\mathcal{P}(x, z; A)$ is derived by the proof system and $z \notin A$. Then Algorithm 1 expands ϕ by the clause*

$$\Big(\big(\bigwedge_{v \in \uparrow_A \setminus (\{x,z\} \cup x\text{-}z\text{-}cut)} x = v \big) \Rightarrow x \geq z \Big).$$

In particular, it expands ϕ by the clause $(x \geq z)$ for every derived $\mathcal{P}(x, z; \emptyset)$.

In the proof of Lemma 15, we use the following simple observation.

▷ Claim 16. For every pair $(x, z) \in V^2$, and every $A \subseteq V_\forall$, there exists $u \in V$ such that $A \subseteq \uparrow_u$ and $\uparrow_A \setminus (\{x, z\} \cup x\text{-}z\text{-cut}) = \uparrow_u \setminus (\{x, z\} \cup x\text{-}z\text{-cut})$.

Proof. It is easy to see that if $A \neq \emptyset$, then we may choose u to be the variable in A that satisfies $u \preceq y$ for all $y \in A$.

If $A = \emptyset$, then we choose u as the last variable in the quantifier-prefix of Φ. Indeed, if u is existential, then we are done. Otherwise, u is universal. If $u \in \{x, z\}$, then this variable is removed from $\uparrow_A$ and we are done. If $u \notin \{x, z\}$, then $u \in x\text{-}z\text{-cut}$. This completes the proof of the observation. ◁

Proof sketch (Lemma 15). We assume that ϕ is expanded by all derived clauses from the run of Algorithm 1, and show that $\big((\bigwedge_{v \in \uparrow_A \setminus (\{x,z\} \cup x\text{-}z\text{-cut})} x = v) \Rightarrow x \geq z \big)$ is among these clauses.

We prove the lemma by induction on the length of the derivation of $\mathcal{P}(x, z; A)$. Observe that it is enough to show that, if $\mathcal{P}(x, z; A)$ is derived, where $z \notin A$, then $\phi \wedge (\bigwedge_{v \in A} x = v) \wedge (x < z)$ is not satisfiable. Then indeed, by Claim 16, we may choose $u \in V$ such that $A \subseteq \uparrow_u$ and $\uparrow_A \setminus (\{x, z\} \cup x\text{-}z\text{-cut}) = \uparrow_u \setminus (\{x, z\} \cup x\text{-}z\text{-cut})$. Since $z \notin A$ and $(x = x)$ is always satisfied, if $\phi \wedge (\bigwedge_{v \in A} x = v) \wedge (x < z)$ is not satisfiable, then neither is $\phi \wedge (\bigwedge_{v \in \uparrow_u \setminus \{x,z\}} x = v) \wedge (x < z)$, and therefore the algorithm expands ϕ by the desired clause. The rest of the proof consists of a straightforward verification of the base case for §I and the induction step for the remaining rules of $\mathcal{P}$. ◀

We conclude this subsection with two examples, the first one showcasing how the run of Algorithm 1 can be represented within the proof system, and the second one demonstrating that, in general, the proof system cannot be used to verify true instances in polynomial time.

▶ **Example 17.** Consider the instance Φ from Example 12. We show that the proof system derives $\perp$. First, we can derive $\mathcal{P}(x_i, x_i; \emptyset)$ for every $i \in [5]$ using §I. With §C (and suitable identifications of variables), we get $\mathcal{P}(x_3, x_1; \emptyset), \mathcal{P}(x_5, x_1; \emptyset)$, $\mathcal{P}(x_1, x_5; \{x_2\})$, $\mathcal{P}(x_3, x_4; \{x_2\})$, and $\mathcal{P}(x_5, x_3; \{x_4\})$. Next, a single application of §A yields $\mathcal{P}(x_1, x_3; \{x_2, x_4\})$. We can use §S to simplify the latter to $\mathcal{P}(x_1, x_3; \{x_2\})$. Using §A again, we get $\mathcal{P}(x_1, x_4; \{x_2\})$, and finally, §S simplifies the latter to $\mathcal{P}(x_1, x_4; \emptyset)$. Now an application of §R yields $\perp$.

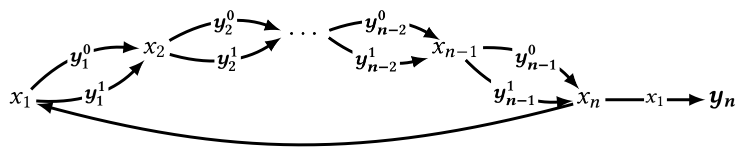

Figure 2 The quantifier-free part of Φ from Example 18.

▶ **Example 18.** Consider $\Phi \coloneqq \exists x_1 \forall y_1^0 \forall y_1^1 \exists x_2 \forall y_2^0 \forall y_2^1 \cdots \exists x_{n-1} \forall y_{n-1}^0 \forall y_{n-1}^1 \exists x_n \forall y_n \ \phi$ with ϕ described by Figure 2, where an edge from x to z labeled with y stands for $\mathrm{M}^+(x, y, z)$. Note that the proof system derives $\mathcal{P}(x_1, x_n; \{y_1^{i_1}, \dots, y_{n-1}^{i_{n-1}}\})$ for all $i_1, \dots, i_{n-1} \in \{0, 1\}$. Indeed, this is because it can follow the shortest derivation sequences, of which there are exponentially many. In contrast, Algorithm 1 derives the constraints $(x_{n-1} \geq x_1), \dots, (x_2 \geq x_1)$, $(x_1 \geq y_n)$ in this order, which leads to rejection. Interestingly enough, constraint paths as in Figure 2 were previously used in [22] to prove PSPACE-hardness of $\mathrm{QCSP}(\mathbb{Q}; \mathrm{D})$.

3.2 True instances

In this subsection we prove Lemma 19, which states that the refutation condition §R from Table 1 is not only sufficient, but also necessary.

▶ **Lemma 19.** *If the proof system does not derive $\perp$ from Φ, then Φ is true.*

Proof. Suppose that the proof system cannot derive $\perp$ from Φ. Consider the following strategy for the EP. Let $x \in \mathrm{V}_\exists$ be such that $[\![x]\!]$ is not yet defined, but $[\![z]\!]$ is defined for every $z \prec x$. Then the EP selects any value for x such that, for every $z \prec x$:

- $[\![x]\!] \geq [\![z]\!]$ *if and only if* there exists $y \prec x$ with $[\![y]\!] \geq [\![z]\!]$ and $\mathcal{P}(x, y; \emptyset)$;
- $[\![x]\!] = [\![z]\!]$ *if and only if* there exist $y_1, y_2 \prec x$ and $y_2 \prec A \prec x$ such that

$$\mathcal{P}(x, y_1; \emptyset) \wedge \mathcal{P}(y_2, x; A) \quad \text{and} \quad [\![z]\!] = [\![y_1]\!] = [\![y_2]\!] = [\![A]\!].$$

We remark that some naïve simplifications of the above strategy fail already on small instances. For example, it is not enough for the EP to set $[\![x]\!] \geq [\![z]\!]$ if and only if $\mathcal{P}(x, z; \emptyset)$. To see this, consider $\Phi = \exists y \forall z \exists x \, \mathrm{M}^+(x, x, y)$. If the UP sets $[\![z]\!] = [\![y]\!]$, then the EP has to respect $(x \geq z)$ even though $\mathcal{P}(x, z; \emptyset)$ is not derived.

▷ Claim 20. The strategy of the EP is well-defined.

Proof. Suppose, on the contrary, that it is not. Let $x \in \mathrm{V}_\exists$ be the smallest variable w.r.t. $\prec$ for which the strategy of the EP is not well-defined. Then it must be the case that there exist $y, y_1, y_2 \prec x$ and $y_2 \prec A \prec x$ such that

$$\mathcal{P}(x, y; \emptyset) \wedge \mathcal{P}(x, y_1; \emptyset) \wedge \mathcal{P}(y_2, x; A) \quad \text{and} \quad [\![y]\!] > [\![y_1]\!] = [\![y_2]\!] = [\![A]\!]. \tag{4}$$

In particular, $y \notin A$. We choose the smallest possible y w.r.t. $\prec$ witnessing a condition of the form (4). By §T, we have $\mathcal{P}(y_2, y; A)$.

Case 1: $y \prec y_2$. Then, by §S, we have $\mathcal{P}(y_2, y; \emptyset)$.

Case 1.1: $y_2 \in \mathrm{V}_\forall$. Then $\perp$ can be derived using §R, a contradiction.

Case 1.2: $y_2 \in \mathrm{V}_\exists$. Then the EP did not follow his strategy because $[\![y]\!] > [\![y_2]\!]$ and we have $\mathcal{P}(y_2, y; \emptyset)$, a contradiction.

Case 2: $y_2 \prec y$.

Case 2.1: $y \in \mathrm{V}_\forall$. Then, by §S, we have $\mathcal{P}(y_2, y; \emptyset)$. But then $\perp$ can be derived using §R, a contradiction.

Case 2.2: $y \in \mathrm{V}_\exists$. Then, by §S, we have $\mathcal{P}(y_2, y; A \setminus y_2\text{-}y\text{-cut})$. Since $[\![y]\!] \geq [\![y_2]\!]$, by the strategy of the EP, there exists a variable $y' \prec y$ such that $[\![y']\!] \geq [\![y_2]\!]$ and $\mathcal{P}(y, y'; \emptyset)$. If $[\![y']\!] = [\![y_2]\!]$, then the EP did not follow his strategy, because he played $[\![y]\!] > [\![y']\!]$ while $y' \prec y$, $y_2 \prec A \setminus y_2\text{-}y\text{-cut} \prec y$, $\mathcal{P}(y, y'; \emptyset) \wedge \mathcal{P}(y_2, y; A \setminus y_2\text{-}y\text{-cut})$, and $[\![y']\!] = [\![y_2]\!] = [\![A \setminus y_2\text{-}y\text{-cut}]\!]$. a contradiction. So it must be the case that $[\![y']\!] > [\![y_2]\!]$. By §T, we have $\mathcal{P}(y_2, y'; A)$. But now y' can assume the role of y in (4), a contradiction to the minimality of y w.r.t. $\prec$. ◁

The next claim characterizes the equality of values for pairs of variables under $[\![\cdot]\!]$ in terms of properties of previously quantified variables, assuming that the EP has followed the strategy above. In particular, we show that if $[\![x]\!] = [\![z]\!]$ if and only if there exists a variable $y \preceq \{x, z\}$ so that $[\![x]\!] = [\![y]\!]$ and $[\![y]\!] = [\![z]\!]$ are enforced by the identifications of values of universal variables with y by the UP. Recall the comparison relations $\preceq$ and $\preceq_\forall$ from Definition 10.

▷ Claim 21. Suppose that the EP follows the strategy above. Then, for all $x, z \in \mathrm{V}$, we have $[\![x]\!] = [\![z]\!]$ if and only if there exist $x_1, x_2, z_1, z_2 \in V$ and $A_{x_2,x}, A_{z_2,z} \subseteq \mathrm{V}_\forall$ such that

1. $\{x_1, x_2\} \preceq x$ and $\{z_1, z_2\} \preceq z$
2. $x_2 \prec A_{x_2,x} \preceq x$ and $z_2 \prec A_{z_2,z} \preceq z$,
3. $y \preceq_\forall \{x_1, x_2, z_1, z_2\}$ for some $y \in \{x_2, z_2\}$,
4. $\mathcal{P}(x_2, x; A_{x_2,x}) \wedge \mathcal{P}(x, x_1; \emptyset) \wedge \mathcal{P}(z_2, z; A_{z_2,z}) \wedge \mathcal{P}(z, z_1; \emptyset)$,
5. $[\![A_{x_2,x}]\!] = [\![x_1]\!] = [\![x_2]\!] = [\![z_1]\!] = [\![z_2]\!] = [\![A_{z_2,z}]\!]$.

Whenever the right-hand side of the equivalence holds, we also have $[\![x]\!] = [\![x_2]\!]$ and $[\![z]\!] = [\![z_2]\!]$.

Proof sketch (Claim 21). "$\Leftarrow$" We show that $[\![x]\!] = [\![x_2]\!]$ and $[\![z]\!] = [\![z_2]\!]$. If $x \equiv x_2$, then clearly $[\![x]\!] = [\![x_2]\!]$. So, w.l.o.g., $x_2 \prec x$. If $x \in \mathrm{V}_\forall$, then §S yields $\mathcal{P}(x_2, x; \emptyset)$ and hence §R produces $\bot$, a contradiction. So we must have $x \in \mathrm{V}_\exists$. Then either $x_1 \equiv x$ or $x_1 \prec x$, and it follows from the strategy of the EP that $[\![x]\!] = [\![x_2]\!]$. Analogously we obtain that $[\![z]\!] = [\![z_2]\!]$. The rest follows by the transitivity of the equality.

"$\Rightarrow$" Whenever the right-hand side of the equivalence in Claim 21 is satisfied, we call $(x, x_1, x_2; A_{x_2,x})$ and $(z, z_1, z_2; A_{z_2,z})$ *witnessing quadruples* for $[\![x]\!] = [\![z]\!]$. If $x \equiv z$, then the statement trivially follows using §I, the witnessing quadruples are $(x, x, x; \emptyset)$ and $(z, z, z; \emptyset)$. So, w.l.o.g., $z \prec x$. If $x \in \mathrm{V}_\forall$, then the claim follows using §I, the witnessing quadruples are again $(x, x, x; \emptyset)$ and $(z, z, z; \emptyset)$. So suppose that $x \in \mathrm{V}_\exists$ and that the claim holds for all pairs of variables preceding x. Since $[\![x]\!] = [\![z]\!]$, by the strategy of the EP, there exist $x_1, x_2 \prec x$ and $x_2 \prec A \prec x$ such that $\mathcal{P}(x, x_1; \emptyset) \wedge \mathcal{P}(x_2, x; A)$ and $[\![z]\!] = [\![x_1]\!] = [\![x_2]\!] = [\![A]\!]$.

Since $[\![x_2]\!] = [\![z]\!]$ and $x_2, z \prec x$, we can apply the induction hypothesis for the pair x_2, z to obtain the witnessing quadruples $(x_2, x_{2_1}, x_{2_2}; A_{x_{2_2},x_2})$ and $(z, z_1, z_2; A_{z_2,z})$. By assumption, there exists $y \in \{z_2, x_{2_2}\}$ such that $y \preceq_\forall \{z_1, z_2, x_{2_1}, x_{2_2}\}$. Note that $[\![y]\!] = [\![x_1]\!]$. Thus, we can apply the induction hypothesis for the pair x_1, y to obtain the witnessing quadruples $(x_1, x_{1_1}, x_{1_2}; A_{x_{1_2},x_1})$ and $(y, y_1, y_2; A_{y_2,y})$. By assumption, there exists $y' \in \{y_2, x_{1_2}\}$ such that $y' \preceq_\forall \{y_1, y_2, x_{1_1}, x_{1_2}\}$. The two cases $y \equiv z_2$ and $y \equiv x_{2_2}$ are illustrated in Figure 3.

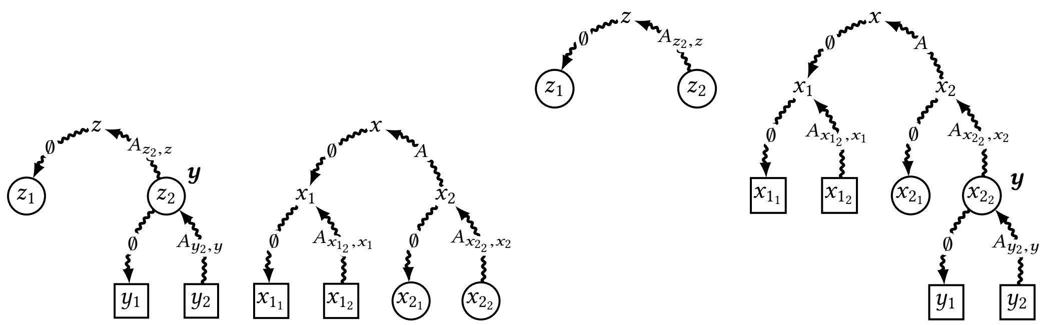

Figure 3 Cases $y \equiv z_2$ and $y \equiv x_{2_2}$ in the proof of Claim 21. The squiggly arrows represent inferences of $\mathcal{P}$.

Our goal is to find witnesses x_1', x_2', z_1', z_2' for the main statement of the claim, i.e., the witnessing quadruples will be of the form $(x, x_1', x_2'; A_{x_2',x})$ and $(z, z_1', z_2'; A_{z_2',z})$. The idea is that we want to choose x_1', x_2', z_1', z_2' from the variables introduced above, which all evaluate to the value $[\![x]\!] = [\![z]\!]$ in $[\![\cdot]\!]$. To obtain the property in item 3, we want to choose variables that are small enough with respect to the order $\prec$, so that one of them can be compared to the others with respect to $\preceq_\forall$, assuming the properties of y and y' from above.

One suitable choice of witnesses is as follows. First, we choose $x_1' := x_{1_1}$. As visible in Figure 3, we can apply §T to $\mathcal{P}(x, x_1; \emptyset) \wedge \mathcal{P}(x_1, x_{1_1}; \emptyset)$ to derive $\mathcal{P}(x, x_{1_1}; \emptyset)$. Second, we choose $z_1' := z_1$ if $y \not\equiv z_1$ and $z_1' := y_1$ otherwise. Note that we have $\mathcal{P}(z, z_1; \emptyset)$ by assumption and, if $y \equiv z_1$, then we can use §T to derive $\mathcal{P}(z, y_1; \emptyset)$ from $\mathcal{P}(z, z_1; \emptyset) \wedge \mathcal{P}(y, y_1; \emptyset)$. Next, we choose $x_2' := x_{2_2}$ if $y \not\equiv x_{2_2}$, and $x_2' := y_2$ otherwise. A short argument shows that $x_{2_2} \preceq_\forall x_{2_1}$, which allows us to apply §A to $\mathcal{P}(x_{2_2}, x_2; A_{x_{2_2}, x_2}) \wedge \mathcal{P}(x_2, x_{2_1}; \emptyset) \wedge \mathcal{P}(x_2, x; A)$ to obtain an expression of the form $\mathcal{P}(x_{2_2}, x; A_{x_{2_2}, x})$. If $y \equiv x_{2_2}$, then it is necessary to apply §A a second time to obtain the expression of the form $\mathcal{P}(y_2, x; A_{y_2, x})$. Finally, the choice for z_2' that we need will be $z_2' := z_2$ if $y \not\equiv z_2$ or $z_2' := y_2$ otherwise.

With the above witnessing quadruples, one can verify that items 1, 2, 4, and 5 will be satisfied. Thanks to choosing "small enough candidates" with respect to $\prec$ for each of x_1', x_2', z_1', z_2', item 3 can be verified as well. A full proof of Claim 21 with a verification of these properties can be found in Appendix A. $\lhd$

$\rhd$ Claim 22. The strategy of the EP is a winning strategy.

Proof. Suppose, on the contrary, that this is not the case. Then there has to be a violated constraint of the form $(x = z \Rightarrow x \geq w)$, i.e., $[\![x]\!] = [\![z]\!] < [\![w]\!]$. Since $[\![x]\!] = [\![z]\!]$, by Claim 21, there exist $x_1, x_2, z_1, z_2 \in V$ and $A_{x_2,x}, A_{z_2,z} \subseteq \mathrm{V}_\forall$ such that

- $\{x_1, x_2\} \preceq x$ and $\{z_1, z_2\} \preceq z$
- $x_2 \prec A_{x_2,x} \preceq x$ and $z_2 \prec A_{z_2,z} \preceq z$,
- $y \preceq_\forall \{x_1, x_2, z_1, z_2\}$ for some $y \in \{x_2, z_2\}$,
- $\mathcal{P}(x_2, x; A_{x_2,x}) \wedge \mathcal{P}(x, x_1; \emptyset) \wedge \mathcal{P}(z_2, z; A_{z_2,z}) \wedge \mathcal{P}(z, z_1; \emptyset)$,
- $[\![A_{x_2,x}]\!] = [\![x_1]\!] = [\![x_2]\!] = [\![z_1]\!] = [\![z_2]\!] = [\![A_{z_2,z}]\!]$.

Moreover, we have $[\![x]\!] = [\![y]\!] = [\![z]\!]$. Let $A := A_{x_2,x} \cup A_{z_2,z} \cup (\{x_1, x_2, z_1, z_2\} \setminus \{y\})$. By a single application of §C, we get $\mathcal{P}(y, w; A)$. Since $[\![z]\!] < [\![w]\!]$, clearly w is different from all variables which share the value with z. We choose the smallest possible w w.r.t. $\prec$ for which $\mathcal{P}(y, w; A)$ can be derived and such that $[\![z]\!] < [\![w]\!]$. Since $[\![w]\!] \neq [\![z]\!]$, we have $w \notin A$. Now we consider the following cases.

Case 1: $w \prec y$**.** By §S, we have $\mathcal{P}(y, w; \emptyset)$.

Case 1.1: $y \in \mathrm{V}_\forall$**.** In this case $\bot$ can be derived using §R, a contradiction.

Case 1.2: $y \in \mathrm{V}_\exists$**.** Then the EP was supposed to set $[\![y]\!] \geq [\![w]\!]$, however, we have $[\![y]\!] < [\![w]\!]$. Hence, the EP did not follow his strategy, a contradiction.

Case 2: $y \prec w$**.**

Case 2.1: $w \in \mathrm{V}_\forall$**.** By §S, we get $\mathcal{P}(y, w; \emptyset)$. But then $\bot$ can be derived using §R, a contradiction.

Case 2.2: $w \in \mathrm{V}_\exists$**.** Since $[\![w]\!] > [\![y]\!]$, by the strategy of the EP, there must exist $w' \prec w$ with $[\![w']\!] \geq [\![y]\!]$ such that $\mathcal{P}(w, w'; \emptyset)$. By §S, we have $\mathcal{P}(y, w; A \setminus y\text{-}w\text{-cut})$. If $[\![w']\!] = [\![y]\!]$, then the EP did not follow his strategy because he played $[\![w]\!] > [\![w']\!]$ while $y, w' \prec w$ and $y \prec A \setminus y\text{-}w\text{-cut} \prec w$, $[\![w']\!] = [\![y]\!] = [\![A \setminus y\text{-}w\text{-cut}]\!]$, and $\mathcal{P}(y, w; A \setminus y\text{-}w\text{-cut}) \wedge \mathcal{P}(w, w'; \emptyset)$, a contradiction. So it must be the case that $[\![w']\!] > [\![y]\!]$. By an application of §T to $\mathcal{P}(y, w; A) \wedge \mathcal{P}(w, w'; \emptyset)$, we have $\mathcal{P}(y, w'; A)$. But note that now w' can assume the role of w, a contradiction to the minimality of w w.r.t. $\prec$. $\lhd$

This concludes the proof of Lemma 19. ◀

3.3 Putting everything together

Proof (Theorem 3). We show that Algorithm 1 solves $\mathrm{QCSP}(\mathbb{Q}; \mathrm{M}^+)$ in polynomial time. Observe that Algorithm 1 runs in polynomial time with respect to the length of Φ. Indeed, it expands Φ by at most V^3-many constraints, all of which have pp-definitions in $(\mathbb{Q}; \mathrm{M}^+, <)$ of linear length due to Lemma 9, and $\mathrm{CSP}(\mathbb{Q}; \mathrm{M}^+, <)$ is solvable in polynomial time [8]. Note that, if Φ contains a clause $(x \geq z)$ or $(z \geq x)$ such that $x \prec z$ and $z \in \mathrm{V}_\forall$, then Φ is false. Therefore, by Lemma 13, Φ is false whenever the algorithm rejects. Suppose that the algorithm accepts an instance Φ. By Lemma 15, $\bot$ cannot be derived from Φ using the proof system and hence, by Lemma 19, Φ is true. This completes the proof. ◀

4 The Complexity Dichotomy

This section is devoted to the proof of Theorem 5. We start by explaining how coNP-hardness is obtained in the cases that are not covered by Theorem 1, Corollary 4, or its analogue for *dual* $\pi\pi$. Recall Theorem 2 that can be used as a black box.

First, we use a syntactical argument to reduce the arity of the relations that need to considered to 4.

▶ **Lemma 23.** *Let $\mathfrak{B}$ be an OH structure that is not contained in the $\pi\pi$ fragment. Then $\mathfrak{B}$ pp-defines a relation of arity at most 4 that is not contained in the $\pi\pi$ fragment.*

Second, we perform an "educated brute-force" search through all relations of arity at most 4 that are not contained in the $\pi\pi$ fragment in order to classify them. Recall the relations D and $\check{\mathrm{Z}}$ defined in the introduction.

▶ **Lemma 24.** *Let $\mathfrak{B}$ be an OH structure that is not contained in the $\pi\pi$ fragment. Then $\mathfrak{B}$ pp-defines* D *or* $(\mathfrak{B}; \mathrm{M}^+)$ *qpp-defines* $\check{\mathrm{Z}}$.

Third, we show coNP-hardness of the QCSP for said relations combined with M^+.

▶ **Lemma 25.** *Let $\mathfrak{B}$ be an OH structure that is not contained in the $\pi\pi$ fragment and pp-defines M^+. Then $\mathrm{QCSP}(\mathfrak{B})$ is* coNP*-hard.*

The proof of Lemma 25 below relies almost entirely on constraint paths built using M^+. In Figure 4, edges relate to constraints over M^+, e.g. the two leftmost arrows in the lower chain represent $\mathrm{M}^+(f, y_1^i, z) \wedge \mathrm{M}^+(z, z, f)$ for $i \in \{1, 2\}$ where z corresponds to an unlabelled vertex. These constraint paths are used to generate exponentially many incomparable expressions within the proof system $\mathcal{P}$, but M^+ itself has no mechanism for turning them into a working gadget. This is why such constraint paths can be handled by Algorithm 1. The situation changes already when we add a single constraint associated to the relation $\check{\mathrm{Z}}$.

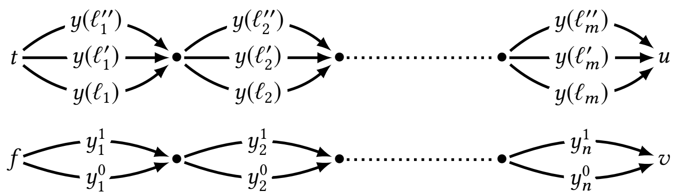

Figure 4 A gadget for the proof of Lemma 25.

Proof (Lemma 25). In the case where $\mathfrak{B}$ pp-defines D, we have that QCSP($\mathfrak{B}$) is PSPACE-hard by Corollary 6 in [22] and Proposition 6. So suppose that $\mathfrak{B}$ does not pp-define D. By Lemma 24, we have that $\mathfrak{B}$ qpp-defines Ž.

We reduce from the complement of the satisfiability problem for propositional 3-CNF. Consider an arbitrary propositional 3-CNF formula ψ, i.e., a conjunction of clauses of the form $\ell_i \vee \ell_i' \vee \ell_i''$ for $i \in [m]$, where $\ell_i, \ell_i', \ell_i''$ are potentially negated propositional variables from $\{x_1, \dots, x_n\}$. We set $\Phi := \exists t \exists f \forall y_1^0 \forall y_1^1 \dots \forall y_n^0 \forall y_n^1 \exists \dots \exists u \exists v \; \phi \wedge \check{\mathrm{Z}}(v, f, u, t)$, where $\exists \dots$ are additional unlabelled existentially quantified variables, and ϕ is defined as in Figure 4. Here $y(x_i) := y_i^1$, $y(\neg x_i) := y_i^0$, a directed edge from x to z labeled with y stands for $\mathrm{M}^+(x, y, z) \wedge \mathrm{M}^+(z, z, x)$,[2] and unlabelled dots correspond to unlabelled existential variables.

"$\Rightarrow$" Suppose that ψ is not satisfiable. We show that the EP has a winning strategy on Φ. First, the EP chooses $[\![f]\!] < [\![t]\!]$. If there exists $i \in [n]$, such that the UP chose $[\![y_i^0]\!] \neq [\![f]\!]$ and $[\![y_i^1]\!] \neq [\![f]\!]$, then the EP chooses the values for the remaining existential variables as follows: equal $[\![f]\!]$ if they appear in the lower chain in Figure 4 before y_i^0 and y_i^1, and equal $[\![t]\!]$ otherwise. Since $[\![f]\!] < [\![t]\!]$, this choice satisfies $\phi \wedge (v \neq f)$. We may therefore assume that $[\![f]\!] \in \{[\![y_i^0]\!], [\![y_i^1]\!]\}$ for every $i \in [n]$.

We claim that there exists $j \in [m]$ such that $[\![t]\!] \notin \{[\![y(\ell_j)]\!], [\![y(\ell_j')]\!], [\![y(\ell_j'')]\!]\}$. Suppose, on the contrary, that this is not the case. Let $[\![\cdot]\!]'$ be any map from $\{x_1, \dots, x_n\}$ to $\{0, 1\}$ such that, for every $i \in [n]$, $[\![x_i]\!]' = 0$ if $[\![y_i^0]\!] = t$ and $[\![x_i]\!]' = 1$ if $[\![y_i^1]\!] = t$. Recall that $[\![f]\!] \in \{[\![y_i^0]\!], [\![y_i^1]\!]\}$ for every $i \in [n]$ and thus $[\![\cdot]\!]'$ is well-defined. Observe that $[\![\cdot]\!]'$ is a satisfying assignment to ψ, contradicting our assumption. Hence the claim holds.

The EP can choose the values for the remaining existential variables as follows: equal $[\![t]\!]$ if they appear in the upper chain in Figure 4 before the j-th column, equal to an arbitrary number $q > [\![t]\!]$ if they appear in the upper chain after the j-th column, and equal $[\![f]\!]$ otherwise. Such assignment satisfies $\phi \wedge (t \neq u)$.

"$\Leftarrow$" Suppose that there exists a satisfying assignment $[\![\cdot]\!]'$ to ψ. We show that then the UP has a winning strategy on Φ. If the EP chooses $[\![f]\!] \geq [\![t]\!]$, the UP wins on Φ, suppose therefore that $[\![f]\!] < [\![t]\!]$. If $[\![x_i]\!]' = 0$, the UP plays $[\![y_i^0]\!] = t$ and $[\![y_i^1]\!] = f$, and if $[\![x_i]\!]' = 1$, the UP plays $[\![y_i^0]\!] = f$ and $[\![y_i^1]\!] = t$. It follows from the lower chain in Figure 4 that the EP loses unless $[\![v]\!] = [\![f]\!]$. Moreover, since $[\![\cdot]\!]'$ is a satisfying assignment to ψ, it follows from the upper chain that the EP loses unless $[\![u]\!] = [\![t]\!]$. But then $[\![\cdot]\!]$ violates $(v \neq f \vee u \neq t)$ and the UP wins again. ◀

We are now ready to prove Theorem 5. As an intermediate step, we prove Corollary 4, which extends the tractability result for M^+ to the whole $\pi\pi$ fragment.

Proof (Corollary 4). By Lemma 9, every relation definable by a clause of the form (3) is pp-definable in $(\mathbb{Q}; \mathrm{M}^+)$. For clauses of the form (3) where the last disjunct $(x \geq z)$ is not present, we may use the pp-definition from Lemma 9 and universally quantify over z, which yields a qpp-definition. Hence, if $\mathfrak{B}$ is as in Corollary 4, then every relation of $\mathfrak{B}$ is qpp-definable in $(\mathbb{Q}; \mathrm{M}^+)$. In this case, QCSP($\mathfrak{B}$) reduces in LOGSPACE to QCSP$(\mathbb{Q}; \mathrm{M}^+)$ due to Proposition 6 and, by Theorem 3, is in PTIME.

For the second part, note that the forward direction immediately follows from the definition of the OH and $\pi\pi$ fragment, since the syntactic form in (3) is a special case of both (2) and (1). For the backward direction, suppose that $\mathfrak{B}$ is an OH structure contained in the $\pi\pi$ fragment. Then every relation has a CNF-definition ϕ over $\{\neq, \geq\}$ where each conjunct is of the form (2) for $k, \ell \geq 0$. By Lemma 8, we can choose ϕ so that every conjunct is of the form (3), possibly with the last disjunct omitted. ◀

[2] Note the difference from the previous interpretation of labeled directed edges, e.g., in Examples 14 and 18. The current interpretation entails $D(x, y, z)$.

Proof (Theorem 5). If $\mathfrak{B}$ is GOH, then QCSP($\mathfrak{B}$) is in PTIME by Theorem 3. If $\mathfrak{B}$ is contained in the $\pi\pi$ fragment, then QCSP($\mathfrak{B}$) is in PTIME by Corollary 4. If $\mathfrak{B}$ is contained in the *dual* $\pi\pi$ fragment, then we reach the same conclusion as in the previous case using the dual version of Corollary 4, which can be obtained by reversing the order in each individual statement used in its proof. Finally, suppose that $\mathfrak{B}$ is not GOH, and also not contained in the $\pi\pi$ or *dual* $\pi\pi$ fragment. Then it either follows directly from Theorem 2 that QCSP($\mathfrak{B}$) is coNP-hard, in which case we are done, or $\mathfrak{B}$ pp-defines M^+ or M^-. If $\mathfrak{B}$ pp-defines M^+, then QCSP($\mathfrak{B}$) is coNP-hard by Lemma 25, because $\mathfrak{B}$ is not contained in the $\pi\pi$ fragment. Otherwise $\mathfrak{B}$ pp-defines M^-. Then we reach the same conclusion as in the previous case using the dual version of Lemma 25, which can again be obtained by reversing the order. ◀

5 The Meta-Problem

For a classification as the one in Theorem 5, one is often interested in the complexity of the following meta-problem: "Does a given structure satisfy the condition for tractability provided by the classification?" In the present section, we give a nondeterministic single-exponential upper bound on the complexity of this meta-problem for Theorem 5.

▶ **Theorem 26.** *The question whether* QCSP($\mathfrak{B}$) *is tractable for a given OH structure* $\mathfrak{B}$ *is decidable in* NEXPTIME*.*

The most natural way to finitely represent a temporal relation R of arity k is by the set of all weak linear orderings on k variables that are witnessed by the tuples in R. Since the first-order theory of $(\mathbb{Q}; <)$ has quantifier-elimination [13], every temporal relation has such a representation. To see this, note that the atomic formula $R(x_1, \ldots, x_k)$ has a quantifier-free definition ψ over $\{<\}$, then it is enough to list all weak linear orderings that are compatible with ψ. For instance, the temporal relation defined by $(x_1 \neq x_2 \vee x_1 < x_3) \wedge (x_1 \leq x_2) \wedge (x_2 \leq x_3)$ admits the following three weak linear orderings:

$$(x_1 = x_2 < x_3), \quad (x_1 < x_2 < x_3), \quad \text{and} \quad (x_1 < x_2 = x_3).$$

Proof (Theorem 26). For every relation R of $\mathfrak{B}$, let ϕ_R be the first-order formula over $\{<\}$ defining R that is provided as an input to the meta-problem. Let ψ_R be the disjunction of all k-ary formulas specifying a weak linear order compatible with R. We claim that ψ_R can be computed from ϕ_R in exponential time. Indeed, to test whether a k-ary formula θ specifying a weak linear ordering is compatible with R, we must only test whether $(\mathbb{Q}; <) \models \forall \bar{x} \big(\theta(\bar{x}) \Rightarrow \phi_R(\bar{x})\big)$. This can be done in PSPACE (Theorem 21.2 in [14]). Clearly, ψ_R defines R. Now, to test whether R is contained in the (*dual*) $\pi\pi$ fragment, we simply guess a conjunction ϕ of clauses of the form (2) that defines R. Note that ϕ might be of size exponential in the size of the input. To verify whether ϕ defines R, we test whether $(\mathbb{Q}; <) \models \forall \bar{x} \big(\phi(\bar{x}) \Leftrightarrow \psi_R(\bar{x})\big)$. This can be done in NEXPTIME because the satisfiability of quantifier-free formulas over $(\mathbb{Q}; <)$, such as $\neg(\phi(\bar{x}) \Leftrightarrow \psi_R(\bar{x}))$, can be decided in NP. Whether R is contained in the GOH fragment can be tested similarly. ◀

6 Open Questions

For quantified OH constraints, we leave the following question open:

Question 1: Do OH QCSPs exhibit a dichotomy between coNP and PSPACE-hardness?

We also ask the following questions regarding open cases outside of OH:

Question 2: Is QCSP($\mathfrak{B}$) in P whenever $\mathfrak{B}$ is a temporal structure contained in the *mi* fragment [7]? It is enough to consider QCSP($\mathbb{Q}; x \neq y \vee x \geq z \vee x > w$) [2].

Question 3: Is QCSP($\mathfrak{B}$) in NP whenever $\mathfrak{B}$ is a temporal structure contained in the $\pi\pi$ fragment? It is enough to consider QCSP($\mathbb{Q}; x \neq y \vee x \geq z_1 \vee x \geq z_2$) [2].

References

1 Serge Abiteboul, Richard Hull, and Victor Vianu. *Foundations of databases*, volume 8. Addison-Wesley, Boston, 1995. URL: `http://webdam.inria.fr/Alice/`.

2 Manuel Bodirsky. *Complexity of Infinite-Domain Constraint Satisfaction*. Cambridge University Press, Cambridge, 2021. `doi:10.1017/9781107337534`.

3 Manuel Bodirsky and Hubie Chen. Quantified equality constraints. *SIAM J. Comput.*, 39(8):3682–3699, 2010. A preliminary version of the paper appeared in the proceedings of LICS'07. `doi:10.1137/080725209`.

4 Manuel Bodirsky, Hubie Chen, and Michał Wrona. Tractability of quantified temporal constraints to the max. *Int. J. Algebra Comput.*, 24(8):1141–1156, 2014. `doi:10.1142/S0218196714500507`.

5 Manuel Bodirsky and Víctor Dalmau. Datalog and constraint satisfaction with infinite templates. *J. Comput. Syst. Sci.*, 79(1):79–100, 2013. A preliminary version appeared in the proceedings of the Symposium on Theoretical Aspects of Computer Science (STACS'05). `doi:10.1016/j.jcss.2012.05.012`.

6 Manuel Bodirsky and Jan Kára. The complexity of equality constraint languages. *Theory Comput. Syst.*, 43(2):136–158, 2008. `doi:10.1007/s00224-007-9083-9`.

7 Manuel Bodirsky and Jan Kára. The complexity of temporal constraint satisfaction problems. *J. ACM*, 57(2):9:1–9:41, 2010. `doi:10.1145/1667053.1667058`.

8 Manuel Bodirsky and Jakub Rydval. On the descriptive complexity of temporal constraint satisfaction problems. *J. ACM*, 70(1):2:1–2:58, 2023. `doi:10.1145/3566051`.

9 Stefano Ceri, Georg Gottlob, and Letizia Tanca. *Logic programming and databases*. Surveys in computer science. Springer, Berlin, Heidelberg, 1990. URL: `https://www.worldcat.org/oclc/20595273`.

10 Witold Charatonik and Michał Wrona. Quantified positive temporal constraints. In Michael Kaminski and Simone Martini, editors, *Computer Science Logic, 22nd International Workshop, CSL 2008, 17th Annual Conference of the EACSL, Bertinoro, Italy, September 16-19, 2008. Proceedings*, volume 5213 of *Lecture Notes in Computer Science*, pages 94–108, Berlin, Heidelberg, 2008. Springer. `doi:10.1007/978-3-540-87531-4_9`.

11 Witold Charatonik and Michał Wrona. Tractable quantified constraint satisfaction problems over positive temporal templates. In Iliano Cervesato, Helmut Veith, and Andrei Voronkov, editors, *Logic for Programming, Artificial Intelligence, and Reasoning, 15th International Conference, LPAR 2008, Doha, Qatar, November 22-27, 2008. Proceedings*, volume 5330 of *Lecture Notes in Computer Science*, pages 543–557, Berlin, Heidelberg, 2008. Springer, Springer. `doi:10.1007/978-3-540-89439-1_38`.

12 Hubie Chen and Michał Wrona. Guarded ord-horn: A tractable fragment of quantified constraint satisfaction. In Ben C. Moszkowski, Mark Reynolds, and Paolo Terenziani, editors, *19th International Symposium on Temporal Representation and Reasoning, TIME 2012, Leicester, United Kingdom, September 12-14, 2012*, pages 99–106, Leicester, UK, 2012. IEEE Computer Society. `doi:10.1109/TIME.2012.19`.

13 Wilfrid Hodges. *A shorter model theory*. Cambridge University Press, Cambridge, 1997.

14 Dexter Kozen. *Theory of computation*, volume 170 of *Texts in Computer Science*. Springer, 2006. `doi:10.1007/1-84628-477-5`.

15 Andrei A. Krokhin, Peter Jeavons, and Peter Jonsson. Reasoning about temporal relations: The tractable subalgebras of allen's interval algebra. *J. ACM*, 50(5):591–640, 2003. `doi:10.1145/876638.876639`.

16 Bernhard Nebel and Hans-Jürgen Bürckert. Reasoning about temporal relations: A maximal tractable subclass of allen's interval algebra. *J. ACM*, 42(1):43–66, 1995. doi:10.1145/200836.200848.

17 Jakub Rydval, Žaneta Semanišinová, and Michał Wrona. Identifying tractable quantified temporal constraints within ord-horn. *CoRR*, abs/2402.09187, 2024. doi:10.48550/arXiv.2402.09187.

18 Michał Wrona. Tractability frontier for dually-closed ord-horn quantified constraint satisfaction problems. In Erzsébet Csuhaj-Varjú, Martin Dietzfelbinger, and Zoltán Ésik, editors, *Mathematical Foundations of Computer Science 2014 - 39th International Symposium, MFCS 2014, Budapest, Hungary, August 25-29, 2014. Proceedings, Part I*, volume 8634 of *Lecture Notes in Computer Science*, pages 535–546, Berlin, Heidelberg, 2014. Springer. doi:10.1007/978-3-662-44522-8_45.

19 Dmitriy Zhuk. A proof of the CSP dichotomy conjecture. *J. ACM*, 67(5):30:1–30:78, 2020. doi:10.1145/3402029.

20 Dmitriy Zhuk. Personal communication, 2022.

21 Dmitriy Zhuk and Barnaby Martin. QCSP monsters and the demise of the chen conjecture. *J. ACM*, 69(5):35:1–35:44, 2022. doi:10.1145/3563820.

22 Dmitriy Zhuk, Barnaby Martin, and Michał Wrona. The complete classification for quantified equality constraints. In Nikhil Bansal and Viswanath Nagarajan, editors, *Proceedings of the 2023 ACM-SIAM Symposium on Discrete Algorithms, SODA 2023, Florence, Italy, January 22-25, 2023*, pages 2746–2760, Florence, Italy, 2023. SIAM. doi:10.1137/1.9781611977554.ch103.

A A Full Proof of Claim 21

▷ Claim 21. Suppose that the EP follows the strategy above. Then, for all $x, z \in \mathrm{V}$, we have $[\![x]\!] = [\![z]\!]$ if and only if there exist $x_1, x_2, z_1, z_2 \in V$ and $A_{x_2,x}, A_{z_2,z} \subseteq \mathrm{V}_\forall$ such that

1. $\{x_1, x_2\} \preceq x$ and $\{z_1, z_2\} \preceq z$
2. $x_2 \prec A_{x_2,x} \preceq x$ and $z_2 \prec A_{z_2,z} \preceq z$,
3. $y \preceq_\forall \{x_1, x_2, z_1, z_2\}$ for some $y \in \{x_2, z_2\}$,
4. $\mathcal{P}(x_2, x; A_{x_2,x}) \wedge \mathcal{P}(x, x_1; \emptyset) \wedge \mathcal{P}(z_2, z; A_{z_2,z}) \wedge \mathcal{P}(z, z_1; \emptyset)$,
5. $[\![A_{x_2,x}]\!] = [\![x_1]\!] = [\![x_2]\!] = [\![z_1]\!] = [\![z_2]\!] = [\![A_{z_2,z}]\!]$.

Whenever the right-hand side of the equivalence holds, we also have $[\![x]\!] = [\![x_2]\!]$ and $[\![z]\!] = [\![z_2]\!]$.

Proof. "⇐" We show that $[\![x]\!] = [\![x_2]\!]$ and $[\![z]\!] = [\![z_2]\!]$. If $x \equiv x_2$, then clearly $[\![x]\!] = [\![x_2]\!]$. So, w.l.o.g., $x_2 \prec x$. If $x \in \mathrm{V}_\forall$, then §S yields $\mathcal{P}(x_2, x; \emptyset)$ and hence §R produces $\perp$, a contradiction. So we must have $x \in \mathrm{V}_\exists$. Then either $x_1 \equiv x$ or $x_1 \prec x$, and it follows from the strategy of the EP that $[\![x]\!] = [\![x_2]\!]$. Analogously we obtain that $[\![z]\!] = [\![z_2]\!]$. The rest follows by the transitivity of the equality.

"⇒" Whenever the right-hand side of the equivalence in Claim 21 is satisfied, we call $(x, x_1, x_2; A_{x_2,x})$ and $(z, z_1, z_2; A_{z_2,z})$ *witnessing quadruples* for $[\![x]\!] = [\![z]\!]$. If $x \equiv z$, then the statement trivially follows using §I, the witnessing quadruples are $(x, x, x; \emptyset)$ and $(z, z, z; \emptyset)$. So, w.l.o.g., $z \prec x$. If $x \in \mathrm{V}_\forall$, then the claim follows using §I, the witnessing quadruples are again $(x, x, x; \emptyset)$ and $(z, z, z; \emptyset)$. So suppose that $x \in \mathrm{V}_\exists$ and that the claim holds for all pairs of variables preceding x. Since $[\![x]\!] = [\![z]\!]$, by the strategy of the EP, there exist $x_1, x_2 \prec x$ and $x_2 \prec A \prec x$ such that $\mathcal{P}(x, x_1; \emptyset) \wedge \mathcal{P}(x_2, x; A)$ and $[\![z]\!] = [\![x_1]\!] = [\![x_2]\!] = [\![A]\!]$.

Since $[\![x_2]\!] = [\![z]\!]$ and $x_2, z \prec x$, we can apply the induction hypothesis for the pair x_2, z to obtain the witnessing quadruples $(x_2, x_{2_1}, x_{2_2}; A_{x_{2_2},x_2})$ and $(z, z_1, z_2; A_{z_2,z})$. By assumption, there exists $y \in \{z_2, x_{2_2}\}$ such that $y \preceq_\forall \{z_1, z_2, x_{2_1}, x_{2_2}\}$. Note that $[\![y]\!] = [\![x_1]\!]$. Thus, we can apply the induction hypothesis for the pair x_1, y to obtain the witnessing quadruples

$(x_1, x_{1_1}, x_{1_2}; A_{x_{1_2},x_1})$ and $(y, y_1, y_2; A_{y_2,y})$. By assumption, there exists $y' \in \{y_2, x_{1_2}\}$ such that $y' \preceq_\forall \{y_1, y_2, x_{1_1}, x_{1_2}\}$. The two cases w.r.t. the variable y that can occur are illustrated in Figure 3, see Case 1 and Case 2 below.

Our goal is to find witnesses x'_1, x'_2, z'_1, z'_2 for the main statement of the claim, i.e., the witnessing quadruples will be of the form $(x, x'_1, x'_2; A_{x'_2,x})$ and $(z, z'_1, z'_2; A_{z'_2,z})$. For the sake of brevity, we will not explicitly write down the precise definitions of $A_{x'_2,x}$ and $A_{z'_2,z}$ as they will be clear from the context. We set $x'_1 := x_{1_1}$, and:

$$z'_1 := \begin{cases} y_1 & \text{if } y \equiv z_1, \\ z_1 & \text{otherwise,} \end{cases} \qquad z'_2 := \begin{cases} y_2 & \text{if } y \equiv z_2, \\ z_2 & \text{otherwise,} \end{cases} \qquad x'_2 := \begin{cases} y_2 & \text{if } y \equiv x_{2_2}, \\ x_{2_2} & \text{otherwise.} \end{cases}$$

By the induction hypothesis and the transitivity of $\prec$, x'_1, x'_2, z'_1, z'_2 clearly satisfy item 1. It will also be clear that our implicit choice of $A_{x'_2,x}$ and $A_{z'_2,z}$ leads to satisfaction of item 5. The remaining three items are proved in the case distinction below. In both Cases 1 and 2, we initially start proving items 2 and 4, and then proceed with item 3 in the finer subdivision into Cases 1.1, 1.2, 2.1, and 2.2. For the sake of conciseness, when applying rules of $\mathcal{P}$ to derive new expressions, we often do not state all necessary expressions for the inference, as long as they are clear from the rule and the resulting expression.

To justify the applications of the rule §A that follow, we observe that $y_2 \preceq_\forall y_1$. Indeed, suppose that $y_1 \not\equiv y_2$. By §T, we have $\mathcal{P}(y_2, y_1, A_{y_2,y})$. If $y_1 \in V_\exists$ or $y_1 \prec y_2$, then $y' \preceq_\forall \{y_1, y_2\}$ implies $y_1 \prec y_2$ and $y_2 \in V_\forall$. By §S, we obtain $\mathcal{P}(y_2, y_1, \emptyset)$, and then using §R we get $\bot$, a contradiction. By an analogous argument, we have $x_{2_2} \preceq_\forall x_{2_1}$. Now it immediately follows that, by §A, we have

$$\mathcal{P}(x_{2_2}, x; A \cup A_{x_{2_2},x_2} \cup (\{x_{2_1}\} \setminus \{x_{2_2}\}). \tag{5}$$

A subsequent double application of §T yields

$$\mathcal{P}(x_{2_2}, x_{1_1}; A \cup A_{x_{2_2},x_2} \cup (\{x_{2_1}\} \setminus \{x_{2_2}\})). \tag{6}$$

Case 1: $y \equiv z_2$. Then $z_2 \preceq_\forall \{z_1, x_{2_1}, x_{2_2}\}$. We now establish item 4 with a suitable choice of sets $A_{x'_2,x}$ and $A_{z'_2,z}$. It is easy to verify that item 2 is satisfied as well.

- If $y \equiv z_1$, then, by §T, we have $\mathcal{P}(z, z'_1; \emptyset)$ because $z'_1 \equiv y_1$. Otherwise $z_1 \in V_\forall$ and $y \prec z_1$; we have $\mathcal{P}(z, z'_1; \emptyset)$, because $z'_1 \equiv z_1$.
- Recall that $y_2 \preceq_\forall y_1$. Thus, by §A, we have

  $$\mathcal{P}(z'_2, z; A_{z_2,z} \cup A_{y_2,y} \cup (\{y_1\} \setminus \{y_2\})), \tag{7}$$

  because $z'_2 \equiv y_2$.
- By §T, we have $\mathcal{P}(x, x'_1; \emptyset)$, because $x'_1 = x_{1_1}$.
- By assumption, we have $y \preceq_\forall x_{2_2}$. Recall that we have (5). If $y \equiv x_{2_2}$, then an application of §A yields

  $$\mathcal{P}(x'_2, x; A \cup A_{x_{2_2},x_2} \cup (\{x_{2_1}\} \setminus \{x_{2_2}\}) \cup A_{y_2,y} \cup (\{y_1\} \setminus \{y_2\})), \tag{8}$$

  because $x'_2 \equiv y_2$. Otherwise $y \prec x_{2_2}$ and $x_{2_2} \in V_\forall$. Then $\mathcal{P}(x'_2, x; A \cup A_{x_{2_2},x_2} \cup (\{x_{2_1}\} \setminus \{x_{2_2}\}))$ follows directly from (5) because $x'_2 \equiv x_{2_2}$.

In the case distinction below, we verify item 3 for x'_1, x'_2, z'_1, z'_2.

Case 1.1: $y' \equiv y_2$. By the choice of y', we have $y_2 \equiv y' \preceq \{y_1, y_2, x_{1_1}\}$. Recall that $y_2 \preceq y \preceq_\forall \{z_1, z_2, x_{2_2}\}$. Hence, if $y \prec z_1$, then $y_2 \preceq_\forall z_1$. Similar applies to z_2 and x_{2_2}. It follows that, for all choices of x'_1, x'_2, z'_1, z'_2 above, we get $z'_2 \equiv y_2 \preceq_\forall \{z'_1, x'_1, x'_2\}$.

Case 1.2: $y' \equiv x_{1_2}$. We show that, as above, $z_2' \equiv y_2 \preceq_\forall \{z_1', x_1', x_2'\}$, starting with x_2'. If $y \equiv z_2 \prec x_{2_2}$ and $x_{2_2} \in V_\forall$, then we have $z_2' \equiv y_2 \preceq_\forall x_{2_2} \equiv x_2'$. Otherwise $x_{2_2} \equiv z_2 \equiv y$ and by our choice of x_2' and z_2', we have $z_2' \equiv y_2$ and $x_2' \equiv y_2$. In particular, $z_2' \preceq_\forall x_2'$. Next comes z_1'. If $y \prec z_1$ and $z_1 \in V_\forall$, then $y_2 \preceq_\forall z_1$ because $y_2 \preceq y$. Consequently, $z_2' \equiv y_2 \preceq_\forall z_1 \equiv z_1'$. Otherwise $y \equiv z_1$ and $z_1' \equiv y_1$. Recall that we always have $y_2 \preceq_\forall y_1$ and hence $z_2' \equiv y_2 \preceq_\forall y_1 \equiv z_1'$.
Finally x_1'. Recall that we have $y \equiv z_2 \preceq_\forall x_{2_2} \preceq_\forall x_{2_1}$ (the second $\preceq_\forall$ was derived above equation (5)). We consider the following two cases. First, suppose that $z_2 \prec x_{2_2}$ and $x_{2_2} \in V_\forall$. Since $x_{2_2} \in V_\forall$, it cannot be the case that $x_{1_1} \prec x_{2_2}$, otherwise §S applied on (6) yields $\mathcal{P}(x_{2_2}, x_{1_1}; \emptyset)$. Using §R, we obtain $\perp$, a contradiction. Hence $x_{2_2} \preceq x_{1_1}$. Now it follows that $y_2 \preceq z_2 \prec x_{2_2} \preceq x_{1_1}$. Since $y' \preceq_\forall \{y_2, x_{1_1}\}$, we even have $z_2' \equiv y_2 \preceq_\forall x_{1_1} \equiv x_1'$. Second, suppose that $x_{2_2} \equiv z_2 \equiv y$. Recall that $y_2 \preceq_\forall y_1$ (derived above (5)) and $\mathcal{P}(y_2, y; A_{y_2,y})$. Combining this with (6) and applying §A, we get

$$\mathcal{P}(y_2, x_{1_1}; A \cup A_{x_{2_2},x_2} \cup (\{x_{2_1}\} \setminus \{x_{2_2}\}) \cup A_{y_2,y} \cup (\{y_1\} \setminus \{y_2\})). \tag{9}$$

We cannot have $x_{1_1} \prec y_2$, otherwise $y_2 \in V_\forall$ in which case §S yields $\mathcal{P}(y_2, x_{1_1}; \emptyset)$ and then §R yields $\perp$, a contradiction. Hence, $y_2 \preceq x_{1_1}$. Since $y' \preceq_\forall \{y_2, x_{1_1}\}$, we get $z_2' \equiv y_2 \preceq_\forall x_{1_1} \equiv x_1'$.

Case 2: $y \equiv x_{2_2}$. Then $x_{2_2} \preceq_\forall \{z_1, z_2, x_{2_1}\}$. We now show that item 4 holds true with a suitable choice of sets $A_{x_2',x}$ and $A_{z_2',z}$; it will be clear that item 2 is satisfied as well.

- If $y \equiv z_1$, then, by §T, we get $\mathcal{P}(z, z_1'; \emptyset)$, because $z_1' \equiv y_1$. Otherwise $z_1 \in V_\forall$ and $y \prec z_1$; we have $\mathcal{P}(z, z_1'; \emptyset)$ because $z_1' \equiv z_1$.
- First, suppose that $y \equiv z_2$. Recall that we have $y_2 \preceq_\forall y_1$. By §A, we have (7) because $z_2' \equiv y_2$. Second, suppose that $y \prec z_2$ and $z_2 \in V_\forall$. Then we have $\mathcal{P}(z_2', z; A_{z_2,z})$ because $z_2' \equiv z_2$.
- By §T, we have $\mathcal{P}(x, x_1'; \emptyset)$ because $x_1' \equiv x_{1_1}$.
- Recall that we have (5) and $y_2 \preceq_\forall y_1$. By §A, we have (8) because $x_2' \equiv y_2$.

Finally, we verify item 3 of the claim.

Case 2.1: $y' \equiv y_2$. By the choice of y', we have $y_2 \equiv y' \preceq \{y_1, y_2, x_{1_1}\}$. Recall that $y_2 \preceq y \preceq_\forall \{z_1, z_2, x_{2_2}\}$. Hence, if $y \prec z_1$, then $y_2 \preceq_\forall z_1$. Similar applies to z_2 and x_{2_2}. It follows that, for all choices of x_1', x_2', z_1', z_2' above, we get $x_2' \equiv y_2 \preceq_\forall \{z_1', z_2', x_1'\}$.

Case 2.2: $y' \equiv x_{1_2}$. By a double application of §T on (8), we get (9). It cannot be that $x_{1_1} \prec y_2$, as this implies $y_2 \in V_\forall$, in which case §S yields $\mathcal{P}(y_2, x_{1_1}; \emptyset)$ and then §R yields $\perp$, a contradiction as in Case 1.2. Hence $x_2' \equiv y_2 \preceq_\forall x_{1_1} \equiv x_1'$. Recall that $y_2 \preceq_\forall y_1$. Either $x_{2_2} \prec z_1$ and $z_1 \in V_\forall$, in which case $x_2' \equiv y_2 \preceq_\forall z_1 \equiv z_1'$, or $x_{2_2} \equiv z_1$ in which case $x_2' \equiv y_2 \preceq_\forall y_1 \equiv z_1'$. Also either $x_{2_2} \prec z_2$ and $z_2 \in V_\forall$, in which case $x_2' \equiv y_2 \preceq_\forall z_2 \equiv z_2'$, or $x_{2_2} \equiv z_2$ in which case $x_2' \equiv y_2 \preceq_\forall y_2 \equiv z_2'$. Hence, $x_2' \preceq_\forall \{z_1', z_2', x_1'\}$. $\triangleleft$