TRACTABILITY OF QUANTIFIED TEMPORAL CONSTRAINTS TO THE MAX

MANUEL BODIRSKY, HUBIE CHEN, AND MICHAL WRONA

Abstract. A temporal constraint language is a set of relations that are first-order definable over \((\mathbb{Q}; <)\). We show that several temporal constraint languages whose constraint satisfaction problem is maximally tractable are also maximally tractable for the more expressive quantified constraint satisfaction problem. These constraint languages are defined in terms of preservation under certain binary polymorphisms. We also present syntactic characterizations of the relations in these languages.

Highly Set-Transitive Structures, Constraint Satisfaction Problems, Temporal Constraint Languages, Max-Closed Constraints, Quantified Constraint Satisfaction.

1. Introduction

For a fixed structure \(\Gamma\) with finite relational signature \(\tau\), the constraint satisfaction problem of \(\Gamma\), denoted by CSP(\(\Gamma\)), is the problem of deciding whether a given sentence of the form \(\exists x_1 \ldots \exists x_n (\psi_1 \land \cdots \land \psi_m)\) is true in \(\Gamma\); here, each \(\psi_i\) is an atomic \(\tau\)-sentence, that is, of the form \(R(y_1, \ldots, y_k)\) for \(R \in \tau\) and \(y_1, \ldots, y_k \in \{x_1, \ldots, x_n\}\). The given sentence is called an instance of CSP(\(\Gamma\)), and the formulas \(\psi_i\) are called the constraints of the instance. The set of relations of \(\Gamma\) is called the constraint language of \(\Gamma\).

A very active research program attempts to classify the computational complexity of CSP(\(\Gamma\)) for all finite structures \(\Gamma\); see e.g. the collection of survey articles [16]. Particularly interesting are structures \(\Gamma\) that are maximally tractable: the idea is that those structures \(\Gamma\) are the structures that sit at the frontier between the easy and the hard CSPs. The notion of maximal tractability also applies to structures \(\Gamma\) with an infinite relational signature. We say that \(\Gamma\) is maximally tractable for the CSP with respect to a class \(\mathcal{C}\) of relations over the same domain (and in this case we also say that the constraint language of \(\Gamma\) is maximally tractable) if

- for every structure \(\Gamma'\) whose constraint language is finite and contained in the constraint language of \(\Gamma\), the problem CSP(\(\Gamma'\)) is in P, and
- for every set of relations \(\mathcal{R} \subseteq \mathcal{C}\) that properly contains the constraint language of \(\Gamma\) there exists a structure \(\Gamma''\) with a finite constraint language that is contained in \(\mathcal{R}\) such that CSP(\(\Gamma''\)) is NP-hard.

Surprisingly often, when \(\Gamma\) is maximally tractable for the CSP with respect to a class of relations, the more expressive quantified constraint satisfaction problem for \(\Gamma\), denoted by QCSP(\(\Gamma\)), can be solved in polynomial time as well. The problem

The research leading to the results presented here has also received funding from the European Research Council under the European Community’s Seventh Framework Program (FP7/2007-2013 Grant Agreement no. 257039.

The author is supported by the Swedish Research Council (VR) under grant 621-2012-3239.
QCSP(\(\Gamma\)) is defined analogously to CSP(\(\Gamma\)), with the difference that, in addition
to existential quantification, universal quantification is also permitted in the input.
That is, an instance of QCSP(\(\Gamma\)) is a sentence of the form
\[ Q_1 v_1 \cdots Q_n v_n (\psi_1 \land \cdots \land \psi_m), \]
for \(Q_1, \ldots, Q_n \in \{\exists, \forall\}\), to be evaluated in \(\Gamma\).

While one might expect that QCSP(\(\Gamma\)) is typically harder than CSP(\(\Gamma\)), for the
following structures \(\Gamma\) both CSP(\(\Gamma\)) and QCSP(\(\Gamma\)) are polynomial-time decidable:

1. 2-element structures \(\Gamma\) preserved by min or by max \([20]\);
2. 2-element structures \(\Gamma\) preserved by the majority operation \([3]\);
3. 2-element structures \(\Gamma\) preserved by the minority operation \([15]\);
4. structures \(\Gamma\) preserved by a near-unanimity operation \([13]\);
5. structures \(\Gamma\) preserved by a Maltsev operation \([9]\);
6. structures \(\Gamma\) that contain a unary relation \(U\) for every subset of its domain,
   and CSP(\(\Gamma\)) is tractable \([5]\).

See \([14]\) for a survey on the state of the art of complexity classification for QCSP(\(\Gamma\))
for finite structures \(\Gamma\).

Maximal tractability of a structure \(\Gamma\) for the QCSP is defined analogously to
maximal tractability of \(\Gamma\) for the CSP. Note that if \(\Gamma\) is maximally tractable for
the CSP with respect to some class of relations \(\mathcal{C}\), and if for all structures \(\Gamma'\)
whose constraint language is finite and contained in the constraint language of \(\Gamma\) the
problem QCSP(\(\Gamma'\)) is in P as well, then \(\Gamma'\) is maximally tractable for the QCSP
with respect to \(\mathcal{C}\) as well. This is for instance the case for all the maximally tractable
structures \(\Gamma\) that appear in (1)-(6) above (each of those 6 items contains structures
\(\Gamma\) that are maximally tractable for the CSP with respect to the class of all relations
over the same domain). If structures that are maximally tractable for the CSP are
also maximally tractable for the QCSP, then this is interesting because it tells us
that in this case the restriction to only existential quantifiers does not make a hard
problem easy.

In this article we study this transfer of maximal tractability for the CSP to max-
imal tractability for the QCSP for structures \(\Gamma\) with an infinite domain. One of the
best-understood classes of infinite structures is the class of all highly set-transitive
structures; these are the structures \(\Gamma\) with the property that for all finite subsets
\(A, B\) of equal size, there exists an automorphism of \(\Gamma\) which sends \(A\) to \(B\). By a
theorem due to Cameron \([10]\), this is the case if and only if \(\Gamma\) is isomorphic to a
structure with domain \(\mathbb{Q}\) whose relations are first-order definable over \((\mathbb{Q}; <)\), i.e.,
for each relation \(R\) of \(\Gamma\) there is a first-order formula that defines \(R\) over \((\mathbb{Q}; <)\)
(without parameters). In the context of constraint satisfaction, the constraint lan-
geuages of highly set-transitive structures have also been called temporal constraint
languages, since many problems in qualitative temporal reasoning can be formu-
lated as CSP(\(\Gamma\)) for such structures \(\Gamma\) (see \([7, 8]\) for examples). We refer to a
relation having a first-order definition in \((\mathbb{Q}; <)\) as a temporal relation. From now
on, maximal tractability will be with respect to the class of all temporal relations
over \(\mathbb{Q}\).

The class of temporal constraint languages is very rich, and often provides ex-
amples and counterexamples for CSPs of infinite structures \(\Gamma\). For example, for
finite structures \(\Gamma\) it is known that CSP(\(\Gamma\)) can be solved by Datalog if and only
if the core of \(\Gamma\) expanded by constants does not interpret primitively positively
the structure \((\mathbb{Z}; p; \{(x, y, z) \mid x - y + z = 1\})\); this follows from the main re-
result in \([2]\) in combination with \([17]\). The same statement is false already for
the simple structure \((\mathbb{Q}; \{(x, y, z) \mid x > y \lor x > z\})\), which does not interpret \((\mathbb{Z}_p; \{(x, y, z) \mid x - y + z = 1\})\) primitively positively with constant\(^1\) but which cannot be solved by Datalog; see [8]. The list of phenomena that are specific to infinite-domain constraint satisfaction and that are exemplified already for temporal constraint languages can be prolonged easily; we refer to [8] for more detail.

The complexity of CSP(\(\Gamma\)) has been completely classified for structures \(\Gamma\) with a temporal constraint language (due to [7]; also see [3]). In this paper we visit some of the CSP maximally tractable temporal constraint languages and study their QCSP. For the languages that we study, we show that even the QCSP is polynomial-time tractable. Our results are as follows.

- When all relations of \(\Gamma\) are preserved by the operation \((x, y) \mapsto \min(x, y)\), then QCSP(\(\Gamma\)) is in P.
- When all relations in \(\Gamma\) are preserved by the operation \(mx\) (introduced in [7]), then QCSP(\(\Gamma\)) is in P. As with min, the operation \(mx\) is binary and commutative; we repeat the definition in Section 3.3.

We complement those results by giving intuitive syntactic descriptions of the temporal relations that are preserved by min, and likewise of those that are preserved by mx. That is, we present a class of syntactically restricted first-order formulas with the property that a temporal relation is preserved by min if and only if it has a definition by a formula from the class, and likewise for mx. Those characterizations are important since they give a more explicit description of the allowed constraints in the respective QCSPs. A similar syntactic description of relations over finite domains preserved by min can be found in [19].

The results that we present for temporal constraint languages preserved by min also hold in analogous form for those that are preserved by max. The operations min and max are dual in the following sense: \(\max(x, y) = -\min(-x, -y)\). Similarly, the operation \(mx\) has the dual operation \((x, y) \mapsto -mx(-x, -y)\), and again the results that we have obtained for \(mx\) also hold analogously for the dual of \(mx\).

2. Temporal Constraint Languages

Let \(f: D^k \to D\) be a function. When \(t^1 = (t^1_1, \ldots, t^1_m), \ldots, t^k = (t^k_1, \ldots, t^k_m) \in D^m\), then \(f(t^1, \ldots, t^k)\) denotes the tuple \((f(t^1_1, \ldots, t^1_m), \ldots, f(t^k_1, \ldots, t^k_m))\) from \(D^m\) that we obtain by applying \(f\) componentwise. A function \(f: D^k \to D\) preserves a relation \(R \subseteq D^m\) if for all \(t^1, \ldots, t^k \in R\) it holds that \(f(t^1, \ldots, t^k) \in R\). If \(f\) preserves all relations of a structure \(\Gamma\), we say that \(f\) is a polymorphism of \(\Gamma\).

**Example.** The operation min: \(\mathbb{Q}^2 \to \mathbb{Q}\) that maps two rational numbers to the minimum of the two numbers is a polymorphism of the temporal constraint language \((\mathbb{Q}; \leq, <)\), but not of the temporal constraint language \((\mathbb{Q}; \neq)\), since it maps for instance the tuples \((1, 0) \notin \neq\) and \((0, 1) \notin \neq\) to \((0, 0) \notin \neq\).

See Figure 1 for an illustration of the operation min. In diagrams for binary operations \(f\) as in Figure 1 we draw a directed edge from \((a, b)\) to \((a', b')\) if \(f(a, b) < f(a', b')\). Unoriented lines in rows and columns of picture for an operation \(f\) relate pairs of values that get the same value under \(f\).

\(^1\)Since it has the polymorphism \(f(x, y) = \min(x, y)\) satisfying \(\forall x, y (f(x, y) = f(y, x))\); see [9] for an introduction to primitive positive interpretations and the related universal algebra.
A non-trivial relation that is preserved by min is the ternary relation defined by the formula $x_1 > x_2 \lor x_1 > x_3$, and the ternary relation $U$ defined as follows.

$$
U(x, y, z) \equiv (x = y \land y < z) \\
\lor (x = z \land z < y) \\
\lor (x = y \land y = z)
$$

Another important operation for temporal constraint satisfaction is the binary operation ‘mx’, which is defined as follows [7].

$$
mx(x, y) :=\begin{cases} 
\alpha(\min(x, y)) & \text{if } x \neq y \\
\beta(x) & \text{if } x = y
\end{cases}
$$

where $\alpha$ and $\beta$ are unary operations that preserve $<$ such that $\alpha(x) < \beta(x) < \alpha(x + \varepsilon)$ for all $x \in \mathbb{Q}$ and all $0 < \varepsilon \in \mathbb{Q}$ (see Figure 2). It is not difficult to show that such operations $\alpha, \beta$ do exist; see [7].

Note that mx does not preserve the relation $\leq$. It also does not preserve the relation $U$ introduced above. An example of a relation that is preserved by mx is the ternary relation $X$ defined as follows.

$$
X(x, y, z) \equiv (x = y \land y < z) \\
\lor (x = z \land z < y) \\
\lor (y = z \land y < x)
$$

Finally, let $pp$ be an arbitrary binary operation on $\mathbb{Q}$ such that $pp(a, b) \leq pp(a', b')$ iff one of the following cases applies:
TRACTABILITY OF QUANTIFIED TEMPORAL CONSTRAINTS TO THE MAX 5

- \( a \leq 0 \) and \( a \leq a' \)
- \( 0 < a, 0 < a' \), and \( b \leq b' \).

Clearly, such an operation exists. For an illustration, see the left diagram in Figure 3. The right diagram of Figure 3 is an illustration of the dual-pp operation. The name of the operation pp is derived from the word ‘projection-projection’, since the operation behaves as a projection to the first argument for negative first argument, and a projection to the second argument for positive first argument.

![Figure 3. A visualization of pp (left) and dual-pp (right).]

The following two propositions have been shown in [7].

**Proposition 2.1 (7).** Let \( R_1, \ldots, R_k \) be temporal relations that are preserved by min (respectively, mx). Then \( \text{CSP}(\mathbb{Q}; R_1, \ldots, R_k) \) is in \( P \).

**Proposition 2.2 (7).** Let \( R \) be a relation with a first-order definition in \((\mathbb{Q}; <)\) that is not preserved by min (respectively, mx). Then there exists a finite set of temporal relations \( R_1, \ldots, R_k \) preserved by min (respectively, mx) such that \( \text{CSP}(\mathbb{Q}; R, R_1, \ldots, R_k) \) is NP-hard.

A straightforward corollary of Proposition 2.1 and Proposition 2.2 is that the sets of temporal relations preserved by min and by mx, respectively, are maximally tractable constraint languages.

The set of temporal relations that is preserved by min is a proper subset of the set of temporal relations that is preserved by pp (see [7]): e.g., the relation \( \{(x, y, z) \in \mathbb{Q}^3 \mid (x = y \land y < z) \lor (x > y \land y = z)\} \) is preserved by pp, but not by min and not by mx. Indeed, the CSP for the temporal constraint language that just contains this ternary relation is NP-hard.

### 3. Syntactic Characterizations

It is known that the structure \((\mathbb{Q}; <)\) has quantifier-elimination [18], that is, for every first-order formula \( \phi \) there exists a quantifier-free formula that is equivalent to \( \phi \) over \((\mathbb{Q}; <)\) (we always allow equality = in our formulas). We will call such a quantifier-free formula over the signature \{\( < \)\} a temporal formula. For the sake of presentation, we write temporal formulas using all symbols in \{\( <, >, \leq, \geq, =, \neq \}\). Let \( \phi_R \) be a temporal formula that defines a temporal relation \( R \). For a tuple \( t \in R \) and a variable \( v \) occurring in \( \phi_R \), we write \( t(v) \) to indicate the value in \( t \) corresponding to the variable \( v \).

It is also well-known that \((\mathbb{Q}; <)\) is a homogeneous structure [21], that is, every isomorphism between finite substructures extends to an automorphism. The set
of automorphisms of \((Q; <)\) will be denoted by \(\text{Aut}(Q; <)\). A set of the form 
\(\{\alpha(t_1), \ldots, \alpha(t_k) \mid \alpha \in \text{Aut}(Q; <)\}\) is called an orbit of \(k\)-tuples (the orbit of \((t_1, \ldots, t_k)\)). It is well known that every \(k\)-ary temporal relation is a union of orbits of \(k\)-tuples.

Let \(\phi_1\) and \(\phi_2\) be two temporal formulas. We say that \(\phi_1\) \textit{entails} \(\phi_2\) if and only if \((Q; <) \models \forall v_1, \ldots, v_n(\phi_1 \Rightarrow \phi_2)\). We also say that an \(n\)-ary temporal relation \(R\) \textit{entails} \(\phi\) if and only if \((Q; <) \models \forall v_1, \ldots, v_n(\phi_R(v_1, \ldots, v_n) \Rightarrow \phi)\), where \(\phi_R\) is a temporal formula defining \(R\). We will use the following fact, which follows directly from the definition of entailment.

**Observation 3.1.** Let \(\phi_R := \Phi \land \psi\) be a temporal formula defining a temporal relation \(R\) and \(\psi'\) a temporal formula entailed by \(R\) which entails \(\psi\). Then \(\phi'_R := \Phi \land \psi'\) also defines \(R\).

### 3.1. Characterization of pp-closed Relations.

**Theorem 3.2.** A temporal relation \(R\) is preserved by pp if and only if it can be defined as a conjunction of clauses of the form

\[
(1) \quad x \neq y_1 \lor \cdots \lor x \neq y_k \lor x \geq z_1 \lor \cdots \lor x \geq z_l,
\]

where it is permitted that \(l = 0\) or \(k = 0\), in which cases the clause is a disjunction of disequalities or contains no disequalities, respectively.

To provide the proof we fix \(n\), and consider a temporal relation of arity \(n\). All the temporal formulas considered in the proof of Theorem 3.2 are meant to have all variables in \(\{v_1, \ldots, v_n\}\). In particular we take advantage of temporal formulas of the form:

\[
(2) \quad \neg(x_1 \circ_1 x_2 \land \cdots \land x_k \circ_{k-1} x_k),
\]

where \(x_1, \ldots, x_k \in \{v_1, \ldots, v_n\}\) are pairwise different and every \(\circ_i\) is in \(\{<, =\}\).

We focus on formulas of the form \((2)\) that satisfy an extra condition that we call the \textit{unambiguity condition} which states that if \((v_i = v_j)\) occurs in \(\phi\), then \(i < j\).

Let \(\Phi_R\) be the set containing all temporal formulas \(\phi\) over variables \(\{v_1, \ldots, v_n\}\) of the form \((2)\) entailed by \(R\) that satisfy the unambiguity condition. We will now argue that \(\bigwedge \Phi_R\) defines \(R\). Observe first that for every orbit of \(n\)-tuples in \(\text{Aut}(Q; <)\), there is the negation \(\phi\) of a formula of the form \((2)\) that defines this orbit. We can certainly assume that \(\phi\) is over variables \(\{v_1, \ldots, v_n\}\) and satisfies the unambiguity condition. On the other hand, to see that every such \(\phi\) defines exactly one orbit observe that any two tuples \(t := (t_1, \ldots, t_n)\) and \(t' := (t'_1, \ldots, t'_n)\) that are in the relation defined by \(\phi\) induce isomorphic substructures in \((Q; <)\). By the homogeneity of \((Q; <)\), an isomorphism between such substructures can be extended to an automorphism of \((Q; <)\). It follows that every orbit may be defined by the negation \(\phi\) of a formula of the form \((2)\). We are now ready to prove that \(R'\) defined by \(\bigwedge \Phi_R\) is equal to \(R\). By the definition of \(R'\) we have that \(R \subseteq R'\). On the other hand if there is an orbit in \(\text{Aut}(Q; <)\) which is not in \(R\), then an appropriate formula of the form \((2)\) is entailed by \(R\). Thus \(R'\) does not contain the orbit as well. Hence we obtain that \(\bigwedge \Phi_R\) defines \(R\).

Finally, we define an order on \(\Phi_R\). For \(\phi_1, \phi_2 \in \Phi_R\) of the form \(\neg(x_1 \circ_1 x_2 \land \cdots \land x_k \circ_{k-1} x_k)\) and \(\neg(y_1 \circ_1 y_2 \land \cdots \land y_l \circ_{l-1} y_l)\), respectively, we will say that \(\phi_1\) is less than \(\phi_2\), in symbols \(\phi_1 \preceq \phi_2\), if \(x_1, \ldots, x_k\) is a subsequence of \(y_1, \ldots, y_l\) and \(\phi_1\) entails \(\phi_2\). It is easy to see that this relation is reflexive and transitive. By
the definition of $\Phi_R$, we have that two formulas $\phi_1, \phi_2 \in \Phi_R$ with the same set of variables entail each other if and only if they are the same formula. Hence, $(\Phi_R; \preceq)$ is also antisymmetric, and we have defined a partial order on $\Phi_R$. Let $\Phi_R^m$ be the set of minimal elements of $(\Phi_R; \preceq)$. By Observation 3.1 it holds that $\bigwedge \Phi_R^m$ defines $R$.

Proof. To prove the right-to-left implication, it is certainly enough to show that every relation $R_C$ defined by a single clause $C$ of the form (1) is preserved by $\text{pp}$. Assume on the contrary that there are $t_1, t_2 \in R_C$ but $t_3 = \text{pp}(t_1, t_2)$ is not in $R_C$. It implies that $t_3(x) = t_3(y_1) = \cdots = t_3(y_k)$, and hence by the definition of $\text{pp}$ it holds that $t_1(x) = t_1(y_1) = \cdots = t_1(y_k)$ or $t_2(x) = t_2(y_1) = \cdots = t_2(y_k)$. In the first case, $t_1(x) \leq 0$ and since for at least one $z_j$ we have that $t_1(z_j) \leq t_1(x)$, it follows that $t_3(z_j) \leq t_3(x)$. Otherwise, we have that $t_1(x) > 0$. Again, for at least one $z_j$ it holds that $t_2(z_j) \leq t_2(x)$. It follows that $t_3(z_j) \leq t_3(x)$. Thus, every clause of the form (1) is preserved by $\text{pp}$.

We now show that every temporal relation $R$ preserved by $\text{pp}$ may be defined as a conjunction of clauses of the desired form. Recall that $\bigwedge \Phi_R^m$ defines $R$. We will obtain the desired definition of $R$ by replacing every member $\phi$ of $\Phi_R^m$, which is of the form (2), with a clause of the form (1). If $\phi = (x = y_1 \wedge \cdots \wedge y_{k-1} = y_k)$, then we replace it by $(x \neq y_1 \lor \cdots \lor x \neq y_k)$. Since these formulas are equivalent, the transformation does not change the defined relation.

Otherwise, $\phi$ contains at least one occurrence of $<$ and hence we can assume that $\phi$ is of the form:

$$\neg((x = y_1 \wedge \cdots \wedge y_{k-1} = y_k) \land y_k < z_1 \land z_1 \circ_1 z_2 \land \cdots \land z_{l-1} \circ_{l-1} z_l),$$

where every $\circ_i$ is in $\{=, <\}$. Here we will consider two cases. The first is where $R$ contains no tuple $t$ satisfying $t(x) = t(y_1) = \cdots = t(y_k) < t(z_j)$ for all $i \in [l]$. Observe that in this case $R$ entails $\psi$ equal to $(x \neq y_1 \lor \cdots \lor x \neq y_k \lor x \geq z_1 \lor \cdots \lor x \geq z_l)$. It is easy to verify that $\psi$ entails $\phi$. Hence, by Observation 3.1 it follows that we can safely replace $\phi$ by $\psi$ in $\Phi_R^m$.

If $R$ contains a tuple $t$ satisfying $t(x) = t(y_1) = \cdots = t(y_k) < t(z_i)$ for all $i \in [l]$, then, as we show, $R$ is not preserved by $\text{pp}$. This leads to a contradiction with the assumption. Consider a formula $\theta$ equal to $\neg(z_1 \circ_1 z_2 \land \cdots \land z_{l-1} \circ_{l-1} z_l)$. We have $\theta \preceq \phi$. Since $\phi$ is in $\Phi_R^m$, it follows that $\theta$ is not in $\Phi_R$, and hence not entailed by $R$. It implies that $R$ contains a tuple $t_1$ satisfying $(z_1 \circ_1 z_2 \land \cdots \land z_{l-1} \circ_{l-1} z_l)$. Let $\alpha$ be automorphisms of $(\mathbb{Q}; <)$ such that $\alpha(t(x)) < 0 < \alpha(t(z_i))$ for all $i \in [l]$. Then $t_2 = \text{pp}(\alpha(t), t_1)$ satisfies $\neg\phi$. Since $\phi$ is entailed by $R$, it follows $t_2 \notin R$, and thus that $R$ is not preserved by $\text{pp}$.

3.2. Characterization of min-closed Relations.

**Theorem 3.3.** A temporal relation $R$ is preserved by $\min$ if and only if it can be defined as a conjunction of clauses of the form

$$x \circ_i z_1 \lor \cdots \lor x \circ_i z_l,$$

where for all $i \in [l]$, we have that $\circ_i$ is in $\{\geq, >\}$.

**Proof.** Observe that to prove the left-to-right implication, it is enough to show that every relation $R_C$ defined by a single clause $C$ of the form (3) is preserved by $\min$. Let $t_1, t_2$ be some tuples in $R$ and $t_3 = \min(t_1, t_2)$. Then $t_1, t_2$ satisfy $x \circ_i z_i$, and $x \circ_j z_j$ for some $i, j \in [l]$, respectively. We consider two cases. Assume first that
$t_1(z_i) = t_2(z_j)$. Here, it is easy to see that $t_3(x) ≥ t_3(z_i)$ and $t_3(x) ≥ t_3(z_j)$. Thus if either $o_i$ or $o_j$ is $≥$, then we are done. Otherwise, it holds that $t_1(x) > t_1(z_i)$ and $t_2(x) > t_2(z_j)$. It follows that $t_3(x)$ is greater than both $t_3(z_i)$ and $t_3(z_j)$. This settles the case where $t_1(z_i) = t_2(z_j)$. Assume now without loss of generality that $t_1(z_i) < t_2(z_j)$. Observe that $t_3(z_i) ≤ t_3(x)$. Thus we are done when $o_i$ is $≤$. Otherwise we have that $t_1(z_i) < t_1(x)$ and that $t_1(z_i) < t_2(z_j) ≤ t_2(x)$. It follows that $t_3(z_i) < t_3(x)$. It completes the proof of the left-to-right implication.

We now turn to prove the reverse implication. Assume that there is some relation $R$ preserved by $\min$ that cannot be defined by a conjunction of clauses of the form $(4)$. By Lemma 23 in [7], we have that every relation preserved by $\min$ is also preserved by $\pp$. Hence, by Theorem 3.2, the relation $R$ may be defined as a conjunction of clauses of the form

$$(x \neq y_1 \lor \cdots \lor x \neq y_k \lor x \circ_1 z_1 \lor \cdots \lor x \circ_l z_l),$$

where $o_i \in \{>, ≥\}$ for every $i \in [l]$. Consider the set of such definitions of $R$ with a minimal number of disequalities and from this set choose one with a minimal number of literals. Denote that formula by $\phi_R$.

If $\phi_R$ does not have any disequalities, then we are done. Otherwise it contains a clause $C$ of the form (4) with at least one disequality $(x \neq y_1)$. Since $\phi_R$ does not contain any unnecessary literals, there is a tuple in $R$ that falsifies all literals in $C$ except for $(x \neq y_1)$. Suppose that $R$ contains both tuples $t_1, t_2$ that falsify all literals in $C$ except for $(x \neq y_1)$ and satisfy $(x < y_1)$ and $(x > y_1)$, respectively. Let $\alpha_1$ be an automorphism of $(\mathbb{Q}; <)$ that sends $(t_1(x), t_1(y_1))$ to $(0, 1)$, and $\alpha_2$ an automorphism of $(\mathbb{Q}; <)$ that sends $(t_2(x), t_2(y_1))$ to $(1, 0)$. Observe that $t_3 = \min(\alpha_1(t_1), \alpha_2(t_2))$ satisfies $(x = y_1)$. Since $\min$ preserves $=, ≤$, and $<$, we have that $t_3$ falsifies all literals in $C$, hence we have a contradiction with the fact that $R$ is preserved by $\min$. It follows that $R$ does not have either $t_1$ or $t_2$. If $R$ does not contain $t_1$, then in $\phi_R$ we can replace $C$ by $(x > y_1 \lor x \neq y_2 \lor \cdots \lor x \neq y_k \lor x \circ_1 z_1 \lor \cdots \lor x \circ_l z_l)$ obtaining a definition of $R$ guaranteed by Theorem 3.2 with a smaller number of disequalities. From now on we can therefore assume that $R$ contains $t_1$.

Consider now the formula $\phi'_R$ obtained from $\phi_R$ by replacing the clause $C$ by the clause $C' := (y_1 \neq x \lor y_1 \neq y_2 \lor \cdots \lor y_1 \neq y_k \lor y_1 \circ_1 z_1 \lor \cdots \lor y_1 \circ_l z_l)$. Observe that $C$ and $C'$ entail each other. Hence $\phi'_R$ also defines $R$. Consider $C'$ as a part of $\phi'_R$ and observe that no literal from $C'$ can be removed. Indeed, the new definition would have less disequalities or the same number of disequalities and less literals than $\phi_R$. Thus $R$ contains a tuple $t'$ that satisfies all disjuncts of $C'$ except for $(y_1 \neq x)$. As in the previous paragraph, we argue that $R$ cannot have both tuples $t_1$ and $t'_1$ that falsify all literals in $C'$ except for $(x \neq y_1)$ and satisfy $(y_1 < x)$ and $(y_1 > x)$, respectively. If $R$ does not contain $t'_1$, then in $\phi'_R$ we can replace $C'$ by $(y_1 > x \lor y_1 \neq y_2 \lor \cdots \lor y_1 \neq y_k \lor y_1 \circ_1 z_1 \lor \cdots \lor y_1 \circ_l z_l)$ obtaining a definition of $R$ in the form guaranteed by Theorem 3.2 with a smaller number of disequalities. Thus, we can assume that $R$ contains $t'_1$.

Now, suppose towards the contradiction that $R$ has both: $t_1$ that falsifies all disjuncts of $C$ except for $(x \neq y_1)$ and satisfies $(x < y_1)$; and $t'_1$ that falsifies all disjuncts of $C'$ except for $(y_1 \neq x)$ and satisfies $(y_1 < x)$. Let $\alpha, \alpha'$ be automorphisms of $\text{Aut}((\mathbb{Q}; <))$ such that $\alpha$ sends $(t_1(x), t_1(y_1))$ to $(0, 1)$ and $\alpha'$ sends $(t'_1(x), t'_1(y_1))$ to $(1, 0)$. To complete the proof we will show that $t_4 = \min(\alpha(t_1), \alpha'(t'_1))$ falsifies all disjuncts of $C$. Since both $t_1$ and $t'_1$ are in $R$, we obtain a contradiction to
the assumption that $R$ is preserved by min. Since $\alpha(t_1(x)) = \alpha(t_1(y_2)) = \cdots = \alpha(t_1(y_k)) = 0$ and $\alpha'(t'_1(y_1)) = \alpha'(t'_1(y_2)) = \cdots = \alpha'(t'_1(y_k)) = 0$, it follows that $t_4(x) = t_4(y_1) = \cdots = t_4(y_k) = 0$ and hence $t_4$ falsifies all disequalities in $C$. Now, the clause $C$ contains a disjunct $(x \circ_i z_i)$ with $\circ_i \in \{>, \geq\}$ and $i \in [k]$ if and only if $C'$ contains $(y_1 \circ_i z_i)$. Assume that $\circ_i$ is $>$. The same argument will work for $\geq$. Observe that $\alpha(t_1)$ satisfies $(x \leq z_i)$ and sends $x$ to $0$; and that $\alpha'(t'_1)$ satisfies $(y_1 \leq z_i)$ and sends $y_1$ to $0$. It follows that $t_4(x) \leq 0 \leq t_4(z_i)$. Thus $t_4$ falsifies $(x < z_i)$ and we are done. \hfill \square

Example. According to Theorem 3.3, the relation $U$ from Section 2 can be defined by a conjunction of clauses of the form $\Box$. Indeed, observe that $((x \geq y \lor x \geq z) \land y \geq x \land z \geq x)$ defines $U$.

3.3. Characterization of $\text{mx}$-closed relations. Let us say that a boolean relation $S \subseteq \{0,1\}^n$ is near-affine if it holds that $S \cup \{(1,\ldots,1)\}$ is preserved by the operation $a(x,y) = x \oplus y + 1$. Let us say that a formula $\phi$ is min-affine if there exists a near-affine relation $T \subseteq \{0,1\}^n$ such that $\phi$ is of the form

$$\bigvee_{t \in T} \left( \left( \bigwedge_{i,j:t_i = t_j = 0} x_i = x_j \right) \land \left( \bigwedge_{i,j:t_i = 0,t_j = 1} x_i < x_j \right) \right)$$

where $x_1,\ldots,x_n$ are variables; here, the formula $\phi$ is intended to be interpreted over the ordered rationals.

Theorem 3.4. A temporal relation $R \subseteq Q^k$ is preserved by $\text{mx}$ if and only if it is definable by a formula $\phi(v_1,\ldots,v_k)$ that is the conjunction of min-affine formulas, each of which is over a subset of $\{v_1,\ldots,v_k\}$.

For a tuple $b = (b_1,\ldots,b_n) \in Q^n$, we define the min-tuple of $b$ to be the tuple $t = (t_1,\ldots,t_n) \in \{0,1\}^n$ such that $t_i = 0$ if and only if $b_i$ is the minimum value in $\{b_1,\ldots,b_n\}$; note that the min-tuple of a $Q$-tuple of nontrivial arity always contains at least one entry equal to $0$. Observe also that a tuple $b \in Q^n$ is in the relation defined by a min-affine formula if and only if the min-tuple of $b$ is in $T$.

We remark that we permit the empty conjunction of min-affine formulas, which we consider to be true, and we also permit min-affine formulas where the near-affine relation $T$ is empty, which formulas are considered to be false.

Proof. For the right-to-left direction, it suffices to show that the claim holds in the case that $\phi$ is defined by a single min-affine formula. So suppose that $\phi$ is of the form described above, with respect to $T \subseteq \{0,1\}^n$. Let $b = (b_1,\ldots,b_n)$, $b' = (b'_1,\ldots,b'_n)$ be two tuples in $Q^n$ that satisfy $\phi$. Let $t, t' \in T$ be the min-tuples of $b, b'$, respectively; so, $t, t'$ witness that $b, b'$ satisfy $\phi$. Let $m, m'$ denote the minimum values in the tuples $b, b'$, respectively.

We consider two cases.

Case $m = m'$. If $t = t'$, then the tuple $\text{mx}(b, b')$ is witnessed to satisfy $\phi$ via the tuple $t = t'$; this can be verified by the definition of $\text{mx}$. If $t \neq t'$, then we claim that the tuple $\text{mx}(b, b')$ is witnessed to satisfy $\phi$ via the tuple $t \oplus t' + 1$. First, observe that $t \oplus t' + 1$ is contained in $T \cup \{(1,\ldots,1)\}$ by the assumption that this relation is preserved by the operation $a$; since $t \neq t'$, it holds that $t \oplus t' + 1$ is not equal to $(1,\ldots,1)$ and is thus contained in $T$. Next, it is straightforwardly verified from the definition of $\text{mx}$ that $\text{mx}(b, b')$ takes on its minimum value at exactly the
coordinates where one of \(b, b'\) takes on its minimum value and the other does not, which are precisely the coordinates where \(t \oplus t' \oplus 1\) is equal to 0.

Case \(m \neq m'\). We assume for the sake of notation that \(m < m'\). In this case, the minimum entries among all entries in \(b\) and \(b'\) are the entries in \(b\) that have value \(m\). It follows from the definition of \(\text{mx}\) that \(\text{mx}(b, b')\) takes on its minimum value at exactly the coordinates where \(b\) does so, and hence the tuple \(\text{mx}(b, b')\) satisfies \(\phi\) via the tuple \(t \in T\).

For the left-to-right direction, we proceed by induction on the arity \(k\) of the relation \(R \subseteq \mathbb{Q}^k\). The case \(k = 0\) is verified as follows: if \(R\) is empty, then take the empty conjunction; if \(R\) is non-empty, then take the conjunction consisting of the single min-affine formula on no variables with \(T = \emptyset\). In what follows, we assume that \(k > 0\). Let \(\phi\) be the conjunction of all min-affine formulas that are entailed by \(R(v_1, \ldots, v_k)\). Suppose that \(b = (b_1, \ldots, b_k) \in \mathbb{Q}^k\) satisfies \(\phi\). Our goal is to show \(b \in R\). Let \(t \in \{0, 1\}^k\) be the min-tuple of \(b\).

We claim that there is a tuple \(c \in R\) with min-tuple \(t\). Let \(T \in \{0, 1\}^k\) be the set of min-tuples of tuples in \(R\). It follows directly from (Lemma 28 of [7]) that if \(s, s' \in T\), then \(s \oplus s' \oplus 1 \in T \cup \{(1, \ldots, 1)\}\). From this implication, it is straightforward to verify that \(T\) is near-affine. Hence, the conjunction \(\phi\) contains as a conjunct a min-affine formula \(\alpha\) with relation \(T\). Since \(b\) satisfies \(\alpha\), it follows that \(t \in T\), from which the claim follows.

If \(t = (0, \ldots, 0)\), then \(c \in R\) and \(b\) are both constant tuples. Since \(b\) is equal to \(c\) under an automorphism of \((\mathbb{Q}; <)\), we have \(b \in R\). So, we suppose that \(t\) contains 1 as an entry; for the sake of notation, we assume that \(t\) has the form 
\((1, \ldots, 1, 0, \ldots, 0)\), where the first \(m\) entries are 1; here, \(0 < m < k\). We have that \((b_1, \ldots, b_m)\) satisfies all min-affine formulas on variables from \(v_1, \ldots, v_m\) which are entailed by \(R(v_1, \ldots, v_k)\). Hence, by induction, it holds that \((b_1, \ldots, b_m) \in \pi_{1, \ldots, m}R\), and that there exists a tuple of the form \((b_1, \ldots, b_m, d)\) in \(R\), where \(d\) is a tuple of length \((k - m)\). We can apply an automorphism to the tuple \((b_1, \ldots, b_m, d)\) to obtain a tuple \((b'_1, \ldots, b'_m, d')\) in \(R\) where all entries are positive. Also, by applying an automorphism to \(c \in R\), we can obtain a tuple of the form 
\((c'_1, \ldots, c'_m, 0, \ldots, 0)\) where, for all coordinates \(i\) from 1 to \(m\), it holds that \(c'_i > b'_i\).

Applying \(\text{mx}\) to the tuples \((b'_1, \ldots, b'_m, d')\) and \((c'_1, \ldots, c'_m, 0, \ldots, 0)\), one obtains the tuple \((\text{mx}(b'_1, c'_1), \ldots, \text{mx}(b'_m, c'_m), \text{mx}(d', 0))\), where \(0\) is a tuple of length \((k - m)\) with all entries equal to 0, which is equivalent to \(b\) under an automorphism. \(\square\)

**Example.** Observe that the relation \(X\) closed under \(\text{mx}\) presented in Section 2 is defined there as a conjunction of min-affine formulas.

4. **Tractability Results**

In this section, we prove the following theorem.

**Theorem 4.1.** Let \(\Gamma\) be a temporal constraint language preserved by an operation \(h \in \{\min, \text{mx}\}\). The problem \(\text{QCSP}(\Gamma)\) is polynomial-time decidable.

Let \(\Gamma\) be a relational \(\tau\)-structure. Formulas of the form

\[
Q_1v_1 \ldots Q_nv_n(\psi_1 \land \cdots \land \psi_m)
\]
where each $\psi_i$ is of the form $R(y_1, \ldots, y_k)$ for $R \in \tau$, and where $Q_1, \ldots, Q_n \in \{\exists, \forall\}$, will be called quantified constraint formulas (over $\Gamma$). Existentially quantified variables are typically (but not exclusively) denoted by $x, x_1, x_2, \ldots$, and universally quantified variables by $y, y_1, y_2, \ldots$.

**Lemma 4.2.** Let $\Gamma$ be a temporal constraint language preserved by an operation $h \in \{\min, \max\}$. Let $\Psi(z_1, \ldots, z_m) = \forall y \exists x_1 \ldots \exists x_n \phi$ be a quantified constraint formula over $\Gamma$ having free variables $\{z_1, \ldots, z_m\}$. Let $\Phi'(z_1, \ldots, z_n)$ be the formula $\exists y \exists x_1 \ldots \exists x_n (\phi \land \bigwedge_{i \in [m]} (z_i < y))$. If the formula $\Phi'$ is satisfiable, then the formulas $\Phi, \Psi'$ have the same satisfying assignments over $\Gamma$.

**Proof.** Observe that for every $q \in \mathbb{Q}$ there exists a value $\alpha(q)$ such that $h(q, p) = h(p, q) = \alpha(q)$ for all $p > q$. Notice that $\alpha(q)$ is injective and preserves $<$. Let $\Psi$ be the formula $\exists y \exists x_1 \ldots \exists x_n \phi$ with free variables $\{z_1, \ldots, z_m, y\}$. Assume that $\Phi$ is satisfiable; then, there exists a tuple $(g_1, \ldots, g_m, y) \in \mathbb{Q}^m$ such that, for all $d \in \mathbb{Q}$, the tuple $(g_1, \ldots, g_m, d)$ satisfies $\Psi$. Clearly, all satisfying assignments of $\Phi$ are satisfying assignments of $\Phi'$. We thus need to show that each satisfying assignment of $\Phi'$ is a satisfying assignment of $\Phi$.

Let $(f_1, \ldots, f_m) \in \mathbb{Q}^m$ be a satisfying assignment of $\Phi'$. By the definitions of the formulas, there exists a value $b \in \mathbb{Q}$ with $b > f_i$ for all $i \in [m]$ such that $(f_1, \ldots, f_m, b)$ satisfies $\Psi$. By applying a suitable automorphism to $(g_1, \ldots, g_m)$, we may assume that $f_i < g_i$ for all $i \in [m]$. Now consider $h((f_1, \ldots, f_m, b), (g_1, \ldots, g_m, d))$ where $d \in \mathbb{Q}$. In the first $m$ coordinates, this tuple is equal to $(\alpha(f_1), \ldots, \alpha(f_m))$. In the last coordinate, this tuple is equal to $h(b, d)$; by varying $d$, the value $h(b, d)$ can take on values from every orbit, where here we understand orbit to be with respect to the group consisting of the automorphisms of $(\mathbb{Q}, <)$ that fix the values $\alpha(f_i)$. It follows that $(\alpha(f_1), \ldots, \alpha(f_m))$ satisfies $\Phi$; since $\alpha^{-1}$ preserves all first-order definable relations, we conclude that $(f_1, \ldots, f_m)$ satisfies $\Phi$. \hfill $\square$

With this lemma in hand, we now prove the theorem. Let $\Gamma$ be a structure that is preserved by min or by max. We now present an algorithm for QCSP($\Gamma$). For the sake of notation, we assume that the input instance is of the form $\forall y_1 \exists x_1 \ldots \forall y_n \exists x_n \phi$, that is, it exhibits a strict alternation between the two quantifier types. The result for the general case can be obtained, for example, by considering an algorithm that inserts dummy variables/quantifiers into an arbitrary instance of QCSP($\Gamma$) to massage it into the described form, and then passes to the algorithm that we describe. We make use of the fact that QCSP($\Gamma$) can be decided by the algorithms given in [7]; indeed, those algorithms describe how to compute satisfying assignments in the event that they exist.

We now discuss the correctness of this algorithm. The algorithm maintains the invariant that, at the beginning of each loop, the formula $\Psi_i(y_1, x_1, \ldots, y_i, x_i)$ is equivalent to the formula $\forall y_{i+1} \exists x_{i+1} \ldots \forall y_n \exists x_n \phi$. (By equivalent, we mean that the formulas have the same satisfying assignments over $\Gamma$.) This is certainly true for $i = n$ by the initialization of $\Psi_n$.

We now consider the behavior of the algorithm for an execution of the loop. If the formula $\Phi'_{i+1}$ is not satisfiable, it follows that the formula $\Phi_i$ is not satisfiable (as any satisfying assignment for $\Phi_i$ is clearly one for $\Phi'_{i+1}$), and hence that the original sentence is false; the algorithm hence reports “False” correctly.

In the case that $\Phi'_{i+1}$ is found to be satisfiable, consider first the case that the satisfying assignment $w$ for $\Phi'_{i+1}$ does not satisfy $\Phi_i$. It follows by Lemma 4.2 that $\Phi_i$
Algorithm for QCSP(Γ) where Γ is preserved by min/mx.

Input: an instance \(\forall y_1 \exists x_1 \ldots \forall y_n \exists x_n \phi\) of QCSP(Γ).

Set \(\Psi_n(y_1, x_1, \ldots, y_n, x_n) = \phi\).

Loop \(i = n\) to 1
  Let \(\Phi_i = \forall y_i \exists x_i \Psi_i\)
  Let \(\Phi'_i = \exists y_i \exists x_i (\Psi_i \land (x_1 < y_i \land \cdots \land x_{i-1} < y_i) \land (y_1 < y_i \land \cdots \land y_{i-1} < y_i))\)
  If \(\Phi'_i\) is satisfiable
    Let \(w\) be a satisfying assignment
  Else
    Return False
  If \(w\) does not satisfy \(\Phi_i\),
    Return False
  [Comment: \(\Phi_i, \Phi'_i\) are equivalent and both satisfiable]
  Let \(\Psi_{i-1} = \Phi'_i\)
End Loop

Return True

is not satisfiable, and hence that the original sentence is false; the algorithm hence reports “False” correctly.

In the case that the assignment \(w\) does satisfy \(\Phi_i\), by Lemma 4.2 the formulas \(\Phi_i, \Phi'_i\) are equivalent. Consequently, when \(\Psi_{i-1}\) is set equal to \(\Phi'_i\), by the definition of \(\Phi_i\) and the fact that the invariant held at the beginning of the loop, the invariant will hold for the next execution of the loop. In the case that \(i = 1\), if the loop does not return false, by the given argumentation, we have that both \(\Phi_1\) and \(\Phi'_1\) are true, and hence that the original sentence is true.

5. Discussion

The polynomial-time algorithms for the QCSP of temporal constraint languages that are preserved by min or preserved by mx that we have seen in this paper properly strengthen previously known algorithmic results for the QCSP:

- In [12], it has been shown that instances of the QCSP where all constraints are of the form \(x \geq y_1 \lor \cdots \lor x \geq y_k\) can be solved in polynomial time. All such constraints are preserved by min (this follows immediately from the syntactic characterization that we gave for temporal relations that are preserved by min).
- In [4], it has been shown that the QCSP for all temporal constraint languages preserved by min is in NP.

There are (up to duality) two more temporal constraint languages whose CSP is maximally tractable [7]. One of them is characterized by a certain binary polymorphism called \(ll\). The tractability for the CSP for \(ll\)-closed languages has been shown in [8]. The QCSP for \(ll\)-closed languages is \(coNP\)-hard [6], so this is an instance where maximal tractability doesn’t transfer to the QCSP. There is the temporal constraint language that contains all relations preserved by the binary operation
TRACTABILITY OF QUANTIFIED TEMPORAL CONSTRAINTS TO THE MAX

mi (for a definition, see [7]). In this case, the tractability of the QCSP is open. In fact, already the following is open.

**Question 5.1.** Is QCSP((Q; \{(x, y, z) \mid x = y \Rightarrow y \leq z\})) in P?

The relation defined by \(x = y \Rightarrow y \leq z\) is indeed preserved by mi (see [7]). Finally, the temporal languages preserved by a constant operation have a tractable CSP [7]. As shown in previous work [12, 11], the QCSP for such languages may be contained in LOGSPACE, be NLOGSPACE-complete, be P-complete, be NP-complete, or be PSPACE-complete.

Another field of open problems is related to the question how bad the complexity of the QCSP becomes when we are outside the maximal tractable classes. From maximal tractability of the CSP we only obtain NP-hardness, but very often the problem might become even PSPACE-hard. Very often, questions in this context lead to the following notorious open problem.

**Question 5.2.** Is QCSP((Q; \{(x, y, z) \mid x = y \Rightarrow y = z\})) PSPACE-complete?

We only know that this QCSP is coNP-hard [6].

Postprint note: D. Zhuk and B. Martin solved Question 5.2 [22].

REFERENCES

[1] B. Aspvall, M. F. Plass, and R. E. Tarjan. A linear-time algorithm for testing the truth of certain quantified boolean formulas. *Information Processing Letters*, 8(3):121–123, 1979.

[2] L. Barto and M. Kozik. Constraint satisfaction problems of bounded width. In *Proceedings of the Annual Symposium on Foundations of Computer Science (FOCS)*, pages 595–603, 2009.

[3] M. Bodirsky. Complexity classification in infinite-domain constraint satisfaction. *Memoire d’habilitation à diriger des recherches, Université Diderot – Paris 7*. Available at arXiv:1201.0856, 2012.

[4] M. Bodirsky and H. Chen. Collapsibility in infinite-domain quantified constraint satisfaction. In *Proceedings of Computer Science Logic (CSL)*, Szeged, Hungary, 2006.

[5] M. Bodirsky and H. Chen. Relatively quantified constraint satisfaction. *Constraints*, 14(1):3–15, 2009.

[6] M. Bodirsky and H. Chen. Quantified equality constraints. *SIAM Journal on Computing*, 39(8):3682–3699, 2010. A preliminary version of the paper appeared in the proceedings of LICS’07.

[7] M. Bodirsky and J. Kára. The complexity of temporal constraint satisfaction problems. *Journal of the ACM*, 57(2):1–41, 2009. An extended abstract appeared in the Proceedings of the Symposium on Theory of Computing (STOC’08).

[8] M. Bodirsky and J. Kára. A fast algorithm and Datalog inexpressibility for temporal reasoning. *ACM Transactions on Computational Logic*, 11(3), 2010.

[9] F. Börner, A. A. Bulatov, H. Chen, P. Jeavons, and A. A. Krokhin. The complexity of constraint satisfaction games and QCSP. *Information and Computation*, 207(9):923–944, 2009.

[10] P. J. Cameron. The random graph. *R. L. Graham and J. Nešetřil, Editors, The Mathematics of Paul Erdős*, 1996.

[11] W. Charatonik and M. Wrona. Quantified positive temporal constraints. In *Proceedings of CSL*, pages 94–108, 2008.

[12] W. Charatonik and M. Wrona. Tractable quantified constraint satisfaction problems over positive temporal templates. In *LPAR*, pages 543–557, 2008.

[13] H. Chen. The computational complexity of quantified constraint satisfaction. Ph.D. thesis, Cornell University, August 2004.

[14] H. Chen. Meditations on quantified constraint satisfaction. In *Logic and Program Semantics - Essays Dedicated to Dexter Kozen on the Occasion of His 60th Birthday*, Lecture Notes in Computer Science 7230. Springer, 2012.
[15] N. Creignou, S. Khanna, and M. Sudan. *Complexity Classifications of Boolean Constraint Satisfaction Problems*. SIAM Monographs on Discrete Mathematics and Applications 7, 2001.

[16] N. Creignou, P. G. Kolaitis, and H. Vollmer, editors. *Complexity of Constraints - An Overview of Current Research Themes [Result of a Dagstuhl Seminar]*, volume 5250 of Lecture Notes in Computer Science. Springer, 2008.

[17] T. Feder and M. Y. Vardi. The computational structure of monotone monadic SNP and constraint satisfaction: a study through Datalog and group theory. *SIAM Journal on Computing*, 28:57–104, 1999.

[18] W. Hodges. *A shorter model theory*. Cambridge University Press, Cambridge, 1997.

[19] P. G. Jeavons and M. C. Cooper. Tractable constraints on ordered domains. *Artificial Intelligence*, 79(2):327–339, 1995.

[20] M. Karpinski, H. K. Bünning, and P. H. Schmitt. On the computational complexity of quantified Horn clauses. In *Proceedings of CSL*, pages 129–137, 1987.

[21] D. Macpherson. A survey of homogeneous structures. *Discrete Mathematics*, 311(15):1599–1634, 2011.

[22] D. Zhuk and B. Martin. The complete classification for quantified equality constraints. *CoRR*, abs/2104.00406, 2021.