Guarded Ord-Horn: A Tractable Fragment of Quantified Constraint Satisfaction

Hubie Chen
Departament de Tecnologies
Universitat Pompeu Fabra
Barcelona, Spain
hubie.chen@upf.edu

Michał Wrona
Department of Computer and Information Science
Linköpings universitet
SE-581 83 Linköping, Sweden
michal.wrona@liu.se

***Abstract*—The first-order theory of dense linear orders without endpoints is well-known to be PSPACE-complete. We present polynomial-time tractability results for fragments of this theory which are defined by syntactic restriction; in particular, our fragments can be described using the framework of quantified constraint satisfaction over *Ord-Horn* clauses.**

I. Introduction

The first-order theory of dense linear orders without endpoints is a classical object of study in model theory; it is well-known that this theory is exactly equivalent to the concrete theory of the ordered rationals $(\mathbb{Q}; <)$. The corresponding computational problem of deciding if a sentence belongs to this theory is PSPACE-complete [1, Lecture 21]. This decision problem appears naturally in various areas of computer science: for instance, it forms part of the first-order theories of the rationals and of the reals, which are of interest in computational algebra [2], and many temporal reasoning and related problems studied in scheduling [3] and artificial intelligence [4] admit formulation in this problem. This theory can also be viewed as the base of and a prototype for (not necessarily linear) orders possessing further structure, such as time intervals, branching time, and spatial regions under the subset ordering.

A natural way to syntactically restrict first-order theories that has proven fruitful in recent complexity studies, and which we pursue in the present work, is the following. First, *expand* the structure(s) of interest by first-order definable relations to obtain a structure $\mathbf{B}$; as an example, in the case of the ordered rationals $(\mathbb{Q}; <)$, first-order definable relations include those defined by $x \leq y$ and $(x \neq y) \vee (y < z)$. Then, consider sentences over the expanded structure, but *restrict* their syntax so that conjunction ($\wedge$) is the only permitted boolean connective. In the case where only existential quantification is permitted, this problem is the *constraint satisfaction problem (CSP)* on $\mathbf{B}$: an instance can be essentially viewed as a prenex formula with only existential quantification, followed by a conjunction of atomic $\mathbf{B}$-formulas, and the problem is to decide if there is an assignment to the variables satisfying all of the atomic formulas, which can be viewed as constraints on the variables. When both quantifiers are present, this problem is known as the *quantified constraint satisfaction problem (QCSP)*. A structure over which the CSP or the QCSP is studied is typically referred to as a *constraint language*.

Classical examples of problems that follow this paradigm are the 2-SAT and Horn-SAT fragments of propositional logic, and their quantified analogs, Quantified 2-SAT [5] and Quantified Horn-SAT [6]; all of these problems are polynomial-time tractable. One obtains these problems within the presented framework in particular by taking the constraint language to contain all relations definable by *2-clauses* or, respectively, all relations definable by *Horn clauses*. Another example of this paradigm is the positive first-order theory of equality [7], which can be obtained by taking the quantified constraint satisfaction problem on the constraint language containing all positively definable relations over equality.

As examples of this paradigm with respect to dense linear orders $<$ without endpoints, consider the following.

- The constraint language with relations $\{<, \leq, \neq\}$ is known as the *point algebra* in temporal reasoning; it and extensions thereof have been studied intensely (see for example [8], [4], [9] and the references therein), and both the CSP and QCSP over it are known to be polynomial-time tractable [10].
- Constraints on relations of the form $(x < y_1) \vee \ldots \vee (x < y_k)$ are known as AND/OR precedence constraints, and arise naturally in scheduling applications [3].
- Constraint solving over *Ord-Horn clauses* is of outstanding importance in temporal and spatial reasoning, as several tractability results there are based into translations into such clauses. (As a starting point for information on Ord-Horn, we suggest the references [11], [12], [9]).
  Ord-Horn clauses are defined as clauses of the form
  $$(x_1 \neq y_1 \vee \cdots \vee x_k \neq y_k \vee xRy),$$
  where $R \in \{<, \leq, \neq, =\}$ and it is permitted that $k = 0$ or that xRy is not present.

A structure whose relations are first-order definable over the ordered rationals $(\mathbb{Q}; <)$ will here be called a *temporal constraint language*. A complete CSP complexity classification of temporal constraint languages was presented by Bodirsky and Kára [13]. This classification result gives a description

1530-1311/12 $26.00 © 2012 IEEE
DOI 10.1109/TIME.2012.19

of the constraint languages whose CSP is polynomial-time tractable, and shows that the remaining languages have an NP-complete CSP. The tractable languages include those examples just described.

In contrast to the CSP, where Ord-Horn is tractable, it is known that for the QCSP, there are simple Ord-Horn clauses that (viewed as constraint languages) are intractable [14]; an example is the clause $(x \neq y) \vee (y = z)$. In this paper, we present polynomial-time tractability results for the QCSP on temporal constraint languages defined by Ord-Horn clauses, **thus giving extremely positive complexity results on the first-order theory of dense linear orders without endpoints.**

We begin by presenting a class of Ord-Horn clauses called *Basic Ord-Horn Formulas*. It has been shown algebraically that constraint languages defined by such formulas enjoy the so-called *local-to-global property* [15]. This is an algorithmically desirable property which (essentially) holds that performing *local consistency*, a iterative, tractable method of performing local inferences on a conjunction of constraints, results in a computationally wieldy characterization of the global solution space of the constraints. This property has been intensely studied and is robust in that it admits formulations from multiple viewpoints (see the discussion in [10]); in the finite, this property is exactly the one for which the classical Baker-Pixley theorem gives logical and algebraic characterizations.

We then define a class of formulas called Guarded Ord-Horn (GOH) formulas; these formulas generalize the class of Basic Ord-Horn formulas, and each such formula can be written as the conjunction of Ord-Horn clauses. The formulas in this class are described using a recursive syntactic definition with Basic Ord-Horn at the base. Our main tractability result is that constraint languages definable by GOH formulas have a polynomial-time decidable QCSP; in developing this result, we make crucial use of an algebraic characterization of the local-to-global property for Basic Ord-Horn. An attractive aspect of this tractability result is that the polynomial-time algorithm that we present is conceptually simple and based on novel pebble games. These pebble games generalize local consistency methods, which have recently received focused attention for the CSP (see [16] for a state-of-the-art result); we believe that this paper will help to bring the study of local consistency methods into focus for the QCSP.

Related work: Quantified constraint satisfaction problems of temporal constraint languages augmented with relations defined by linear inequalities have been studied in [17]. One example of such a problem is the quantified version of the feasibility problem for linear programs. It is shown that this QCSP is coNP-hard, and contained in PSPACE, but the precise complexity remains open. The paper presents also a tractable QCSP; the corresponding constraint language does not contain any of the tractable languages presented here.

Charatonik and Wrona [18], [19] studied the QCSP for temporal constraint languages where all relations have a *positive* first-order definition over the *reflexive* dense linear order $\leq$. They showed the complete complexity characterization for the QCSP on these constraint languages and organized them into LOGSPACE, NLOGSPACE-complete, P-complete, NP-complete and PSPACE-complete problems.

II. Preliminaries

A. Quantified Constraint Satisfaction

Throughout, we assume that each signature is a finite set of relation symbols. Let τ be such a signature. A first-order τ-sentence is a *quantified constraint sentence* if it has the form $Q_1 v_1 \dots Q_n v_n (\psi_1 \wedge \dots \wedge \psi_m)$, where each Q_i is a quantifier from $\{\forall, \exists\}$, and each ψ_i is an atomic τ-formula of the form $R(x_1, \dots, x_k)$ where $R \in \tau$.

Let $\mathbf{B}$ be a τ-structure with domain B; that is, $\mathbf{B}$ has for each relation symbol R in τ a relation $R^{\mathbf{B}} \subseteq B^k$ where k is the *arity* of R; see for example [20]. The *quantified constraint satisfaction problem* for $\mathbf{B}$, denoted by QCSP($\mathbf{B}$), is the problem of deciding, given a quantified constraint τ-sentence Φ, whether or not $\mathbf{B} \models \Phi$.

In this paper, we use $(\mathbb{Q}; <)$ as a concrete model whose theory is the one of interest. A *temporal relation* [13] is a relation $R \subseteq \mathbb{Q}^k$ with a first-order definition in $(\mathbb{Q}; <)$; that is, there exists a first-order formula $\phi(x_1, \dots, x_k)$ with free variables $x_1, \dots, x_k$ such that $(a_1, \dots, a_k) \in R$ if and only if $\phi(a_1, \dots, a_k)$ is true in $(\mathbb{Q}; <)$. A *temporal constraint language* is a relational structure $\mathbf{B}$ with domain $\mathbb{Q}$ where each relation $R^{\mathbf{B}}$ of $\mathbf{B}$ is a temporal relation. In this paper we study QCSP($\mathbf{B}$) only for structures $\mathbf{B}$ with a *finite* signature. It is well-known and follows from the theorem of Ryll-Nardzewski (see [20]) that all temporal constraint languages are ω-categorical.

We say that a k-ary function (also called *operation*) $f : B^k \to B$ *preserves* an m-ary relation $R \subseteq B^m$ if whenever $R(a_1^i, \dots, a_m^i)$ holds for all $1 \leq i \leq k$, then $R\big(f(a_1^1, \dots, a_1^k), \dots, f(a_m^1, \dots, a_m^k)\big)$ holds as well. Let $\mathbf{B}$ be a structure, and let ϕ be a first-order formula with free variables $x_1, \dots, x_k$. Then we say that f preserves ϕ (over $\mathbf{B}$) if f preserves the k-ary relation defined by ϕ over $\mathbf{B}$. If f preserves all relations of a relational structure $\mathbf{B}$, we say that f is a *polymorphism* of $\mathbf{B}$ (see e.g. [21] for background on the role of polymorphisms in the study of CSPs). Unary polymorphisms are called the *endomorphisms* of $\mathbf{B}$; bijective endomorphisms of $\mathbf{B}$ whose inverse is also an endomorphism are called *automorphisms* of $\mathbf{B}$. In this paper, use of the term automorphism is assumed to refer to automorphisms of $(\mathbb{Q}; <)$, unless otherwise specified. An *orbit* is a subset of $\mathbb{Q}$-tuples that is equal to the closure of a tuple $t \in \mathbb{Q}^k$ under all automorphisms.

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 $(\mathbb{Q}; \leq, <)$. It is not a polymorphism of the temporal constraint language $(\mathbb{Q}; \neq)$, since it maps for instance the tuples $(1, 0) \in \neq$ and $(0, 1) \in \neq$ to $(0, 0) \notin \neq$. An interesting relation that is preserved by $\min$ is the ternary relation defined by the formula $x_1 > x_2 \vee x_1 > x_3$, and the ternary relation U with $U(x, y, z)$ defined by

$$(x = y \wedge y < z) \vee (x = z \wedge z < y) \vee (x = y \wedge y = z).$$

Definition 2.1: Let B be any set. A k-ary operation $f : B^k \to B$ is called a *quasi near-unanimity* function (or QNUF) if for all $x, y \in B$ we have

$$f(x, x, \ldots, x, y) = f(x, x, \ldots, y, x) = \ldots = f(y, x, \ldots, x, x) = f(x, \ldots, x) \ .$$

□

A polymorphism $f : B^k \to B$ of $\mathbf{B}$ is called *oligopotent* (wrt $\mathbf{B}$) if the mapping $f^* : B \to B$ defined by $f^*(x) := f(x, \ldots, x)$ preserves all first-order definable relation of $\mathbf{B}$. This definition is different from the definition in [22], but equivalent to it (see Proposition 12 in that paper), and more convenient for our purposes.

Theorem 2.2 (follows from Theorem 19 in [22]): Let $\mathbf{B}$ be an ω-categorical relational structure with an oligopotent $k+1$-ary polymorphism that is a quasi near-unanimity operation, for some $k \geq 2$. Then every n-ary relation R in $\mathbf{B}$ contains all tuples t such that for every subset I of $\{1, \ldots, n\}$ with $|I| \leq k$ there is a tuple $s \in R$ such that $t[i] = s[i]$ for all $i \in I$.

A simple example of an oligopotent QNUF is the ternary median operation median: $\mathbb{Q}^3 \to \mathbb{Q}$ which maps its three arguments to the middle value of those arguments; if two or more arguments are equal to a value c, then the median operation returns c. In particular, $\text{median}(x, x, x) = x$ for all $x \in \mathbb{Q}$, and hence median is oligopotent.

Let $\mathbf{B}$ be a structure with domain B over signature τ. If $E \subseteq B^2$ is an equivalence relation, we write $\mathbf{B}/E$ for the *factor structure* $\mathbf{A}$ with signature τ, defined as follows. The domain of $\mathbf{A}$ consists of the equivalence classes of E, and for each relation symbol $R \in \tau$ we have $R^{\mathbf{A}} = \{([c_1], \ldots, [c_k]) \mid (c_1, \ldots, c_k) \in R^{\mathbf{B}}\}$, where $[c]$ denotes the E-equivalence class of $c \in D$.

B. Basic Ord-Horn

Throughout, we use *OH* as short for Ord-Horn.

Definition 2.3: The set of *basic OH formulas* over a variable set V is the set containing the following formulas:

- $x = y$,
- $x \leq y$,
- $(x_1 \neq y_1 \vee \ldots \vee x_p \neq y_p)$,
- $(x_1 \neq x_2 \vee \ldots x_1 \neq x_q) \vee (x_1 < y_1) \vee (y_1 \neq y_2 \vee \ldots y_1 \neq y_{q'})$.

It is assumed that the variables x, y, x_i, y_i are in V. □

It has been demonstrated that an oligopotent QNUF with certain desirable properties can be constructed for basic OH formulas. We describe the result by making use of the following notions from [10, Section 4], adapted to the current context. Let $t \in \mathbb{Q}^k$ be a tuple, where $k \geq 3$, and let $\pi : [k] \to [k]$ be a permutation such that $t_{\pi(1)} \leq \cdots \leq t_{\pi(k)}$. If it holds that $t_{\pi(2)} = \cdots = t_{\pi(k-1)}$, we say that this value is the *main value* of t. Not every tuple $t \in \mathbb{Q}^k$ has a main value, but when a tuple has a main value, it is unique. We define the equivalence relation $\equiv_m$ on $\mathbb{Q}^k$ as follows: $t \equiv_m t'$ if and only if $t = t'$ or t, t' have the same main value. We say that an operation $h : \mathbb{Q}^k \to \mathbb{Q}$ is *main-injective* if for all $t, u \in \mathbb{Q}^k$, it holds that $h(t) = h(u)$ if and only if $t \equiv_m u$.

Theorem 2.4: [15] For any finite set F of basic OH formulas, there exists a main-injective surjective oligopotent QNUF preserving all formulas in F.

III. Tractability of Guarded Ord-Horn

A. Definitions

Throughout, we will use GOH as short for *Guarded Ord-Horn*.

Definition 3.1: The set of *GOH formulas* over a variable set V is defined inductively as follows.

1) Basic OH formulas are GOH.
2) If ψ_1 and ψ_2 are GOH formulas, then $\psi_1 \wedge \psi_2$ is a GOH formula.
3) If ψ is a GOH formula, then
$$(x_1 \leq y_1) \wedge \ldots \wedge (x_m \leq y_m) \wedge (x_1 \neq y_1 \vee \ldots \vee x_m \neq y_m \vee \psi)$$
is a GOH formula. Here, it is assumed that the variables x_i, y_i are contained in V.

We refer to GOH formulas formed according to these rules as formulas of type 1, 2, and 3, respectively. □

Definition 3.2: A relation $R \subseteq \mathbb{Q}^n$ is GOH if there exists a GOH formula ϕ over the variables $\{v_1, \ldots, v_n\}$ such that $R(v_1, \ldots, v_n) \equiv \phi(v_1, \ldots, v_n)$. A structure $\mathbf{B}$ is GOH if it is over a finite signature and each of its relations is GOH, that is, if for every relation symbol S from the signature of $\mathbf{B}$, it holds that $S^{\mathbf{B}}$ is GOH. □

We now introduce a normal form for GOH formulas.

Definition 3.3: A GOH formula ϕ is *normal* if for each subformula of type 3, it holds that, for each $i \in [m]$, the variables x_i and y_i are different. □

Proposition 3.4: Each GOH formula ϕ is logically equivalent to a normal GOH formula $N(\phi)$.

Proof. We show how to construct inductively, from any GOH formula ϕ, a normal GOH formula $N(\phi)$. We consider the three types of GOH formulas in turn.

1) $N(\phi) = \phi$
2) $N(\psi_1 \wedge \psi_2) = N(\psi_1) \wedge N(\psi_2)$
3) When $\phi = (x_1 \leq y_1) \wedge \ldots \wedge (x_m \leq y_m) \wedge (x_1 \neq y_1 \vee \ldots \vee x_m \neq y_m \vee \psi)$, let $i_1, \ldots, i_n$ be a list of the elements of $D = \{i \mid x_i, y_i \text{ are different variables }\}$. Let $N(\phi)$ be the formula $(x_{i_1} \leq y_{i_1}) \wedge \ldots \wedge (x_{i_n} \leq y_{i_n}) \wedge (x_{i_1} \neq y_{i_1} \vee \ldots \vee x_{i_n} \neq y_{i_n} \vee \psi)$, where it is understood that $N(\phi) = \psi$ if $D = \emptyset$.

It is clear that any GOH formula ϕ is equivalent to a normal GOH formula $N(\phi)$, over $(\mathbb{Q}, <)$. □

Definition 3.5: Let ϕ be a normal GOH formula. We define the *guards formula* of ϕ, denoted by $G(\phi)$, inductively as follows. We consider the three types of GOH formulas in turn.

1) For a basic OH formula ϕ: $G(\phi) = \phi$
2) $G(\psi_1 \wedge \psi_2) = G(\psi_1) \wedge G(\psi_2)$
3) $G((x_1 \leq y_1) \wedge \ldots \wedge (x_m \leq y_m) \wedge (x_1 \neq y_1 \vee \ldots \vee x_m \neq y_m \vee \psi)) = (x_1 \leq y_1) \wedge \ldots \wedge (x_m \leq y_m)$

For an arbitrary GOH formula ϕ, we define the *guards formula* of ϕ as $G(\phi) = G(N(\phi))$, where $N(\phi)$ is the operator from Proposition 3.4. □

Clearly, every guards formula is the conjunction of basic OH formulas.

Definition 3.6: Let $\mathbf{B}$ be a GOH structure over signature σ, and let $R(w_1,\ldots,w_n)$ be an atomic formula over σ (where the variables $w_1,\ldots,w_n$ are not necessarily distinct). We define $\overline{R(w_1,\ldots,w_n)}$ as the formula $G(\phi(w_1,\ldots,w_n))$, where ϕ is a GOH formula for $R^{\mathbf{B}}$ (as in Definition 3.2). We extend this notation to quantified constraint sentences by defining, for such a sentence $\Phi = Q_{v_1}v_1\ldots Q_{v_n}v_n \wedge_{i=1}^m \psi_i$, where the ψ_i are atomic formulas, $\overline{\Phi} = Q_{v_1}v_1\ldots Q_{v_n}v_n \wedge_{i=1}^m \overline{\psi_i}$. □

We observe that if the sentence Φ is true with respect to a GOH structure $\mathbf{B}$, then the sentence $\overline{\Phi}$ is also true: one can conceive of $\overline{\Phi}$ as a relaxation of Φ.

We will make use of the following consequence of Theorem 2.4.

Corollary 3.7: Let $\mathbf{B}$ be a GOH structure over signature σ. There exists a main-injective surjective oligopotent QNUF preserving all formulas of the form $\overline{R(w_1,\ldots,w_n)}$, where $R(w_1,\ldots,w_n)$ is an atomic formula over σ.

Proof. There are clearly a finite number of such formulas of the described form, up to renaming of variables. Each such formula is the conjunction of basic OH formulas; letting F be the set of all basic formulas that can appear in such a conjunction, the result follows directly from Theorem 2.4. □

B. Pebble games

Throughout this subsection, Φ denotes a quantified constraint sentence $\Phi = Q_{v_1}v_1\ldots Q_{v_n}v_n\phi$ in prenex form. We use V_Φ to denote the set of variables $\{v_1,\ldots,v_n\}$ of Φ, and for each variable $v \in V_\Phi$, we use Q_v to denote its corresponding quantifier. We write $u <_\Phi v$ to indicate that u occurs strictly before v in the quantifier prefix of Φ; we write $u \leq_\Phi v$ to indicate that $u <_\Phi v$ or $u = v$. We extend this notation to sets of variables U, W; we write, for instance, that $U <_\Phi W$ if $u <_\Phi w$ for all $u \in U, w \in W$, and that $U <_\Phi w$ for a variable w if $U <_\Phi \{w\}$. We assume that $\mathbf{B}$ is a relational structure and that Φ is a quantified constraint sentence that is an instance of QCSP($\mathbf{B}$). Much of the development is relative to the structure $\mathbf{B}$, but we will generally suppress explicitly mentioning the structure $\mathbf{B}$.

In order to establish our tractability result, we will consider a number of pebble games (for more on pebble games in the context of constraint satisfaction, see e.g. [23]). The most basic pebble game that we will consider is here referred to simply as the k-pebble game, and is based on the pebble game for quantified constraint sentences defined by Chen and Dalmau [24]. The k-pebble game is played between two players, Spoiler and Duplicator. Spoiler can use up to k pebbles, and can perform two actions. First, Spoiler can place a pebble on a variable of the sentence that comes after all variables having pebbles; if it is an existentially quantified variable, Duplicator has to respond by placing a pebble on an element b of the structure $\mathbf{B}$, and if it is a universally quantified variable, the Spoiler also places a pebble on an element b of $\mathbf{B}$ of his choosing. Second, Spoiler can remove a pebble that is on a variable of the sentence, in which case the corresponding structure pebble is removed. At the start of the game, no pebbles are in play. The Duplicator wins if he can play forever in such a way that the mapping from variables to $\mathbf{B}$-elements determined by pebbles is always a partial solution.

The notion of partial solution that we use in formalizing the pebble games used is that of *projective homomorphism*. We say that a function f defined on a subset of V_Φ is a *projective homomorphism* of Φ if for each atomic formula $R(v_1,\ldots,v_k)$ of Φ, there exists an extension of f that satisfies $R(v_1,\ldots,v_k)$ over $\mathbf{B}$. Equivalently, such a function f is a projective homomorphism of Φ if for each atomic formula $R(v_1,\ldots,v_k)$, there exists a tuple $(b_1,\ldots,b_k) \in R^{\mathbf{B}}$ such that for all $v_i \in \mathsf{dom}(f)$, it holds that $f(v_i) = b_i$. We use $\mathsf{dom}(f)$ to denote the domain of a function f. Similarly, we say that a function f defined on a subset of V_Φ is a *projective homomorphism* of $\overline{\Phi}$ if for each formula $\overline{R(v_1,\ldots,v_k)}$ in $\overline{\Phi}$ corresponding to an atomic formula of Φ, there exists an extension of f that satisfies $\overline{R(v_1,\ldots,v_k)}$ over $\mathbf{B}$.

The notion of a (Duplicator) winning strategy for the k-pebble game can be formalized in the following way.

Definition 3.8: A winning strategy for the k-pebble game (on Φ) is a set S of projective homomorphisms, each having domain of size less than or equal to k, that satisfies the following conditions.

(1) The partial function with empty domain is an element of S,
(2) If $f \in S$ and g is a restriction of f, then $g \in S$.
(3) If $f \in S$, $|\mathsf{dom}(f)| < k$, $v \in V_\Phi$, and $\mathsf{dom}(f) <_\Phi v$, then f can be (v, Q_v)-*extended* in S.

We say that an operation f can be (v, Q)-extended in a set of operations S if:

- when $Q = \exists$, there exists an extension $g \in S$ of f with domain $\mathsf{dom}(g) = \mathsf{dom}(f) \cup \{v\}$, and
- when $Q = \forall$, every extension g of f with domain $\mathsf{dom}(g) = \mathsf{dom}(f) \cup \{v\}$ is in S.

□

We next introduce an extension of the k-pebble game where the Spoiler can perform "backmoves" or "backpebbling": he can place a pebble on a variable that does not come after all pebbled variables, and the Duplicator must respond.

Definition 3.9: A winning strategy for the k-pebble game with backmoves (on Φ) is a set S of projective homomorphisms, each having domain of size less than or equal to k, that is a winning strategy for the k-pebble game (Definition 3.8) and in addition satisfy the following condition.

(4) If $f \in S$, $|\mathsf{dom}(f)| < k$, $v \in V_\phi$, and $\mathsf{dom}(f) \not<_\Phi v$, then there exists an extension $g \in S$ of f with domain $\mathsf{dom}(g) = \mathsf{dom}(f) \cup \{v\}$.

□

The definition of the k-pebble game with backmoves naturally gives rise to an algorithm for deciding whether or not there is a winning strategy for this game. The algorithm is as follows.

It is clear that, for each fixed k, and any finite structure $\mathbf{B}$, the algorithm runs in polynomial time, measured with

Algorithm for k-pebble game with backmoves.

Input: a quantified constraint sentence Φ

1. Let S be the set of all projective homomorphisms of Φ having domain of size less than or equal to k, relative to the structure $\mathbf{B}$ of interest.

2. Repeat until no changes are possible:

- If there exists $f \in S$ having a restriction g not in S, then remove f from S.
- If there exists $f \in S$ and $v \in V_\Phi$ with $|\mathsf{dom}(f)| < k$ and $\mathsf{dom}(f) <_\Phi v$ such that f cannot be (v, Q_v)-extended in S, then remove f from S.
- If there exists $f \in S$ and $v \in V_\Phi$ with $|\mathsf{dom}(f)| < k$ and $\mathsf{dom}(f) \not<_\Phi v$ such that there is no extension $g \in S$ of f with domain $\mathsf{dom}(g) = \mathsf{dom}(f) \cup \{v\}$, then remove f from S.

3. Return S.

respect to the input Φ. This algorithm can be applied to temporal constraint languages $\mathbf{B}$ by maintaining, at all times, an orbit representative for each orbit that is present in S, and then performing the described computations using the orbit representatives; this is justified by the fact that the set of all projective homomorphisms is closed under automorphism. Again, for each fixed k and any temporal constraint language $\mathbf{B}$, the algorithm runs in polynomial time in the input Φ.

In the case that, when the algorithm terminates, the set S is nonempty, we call this resulting strategy S the *maximal winning strategy*. The following proposition is straightforward to verify.

Proposition 3.10: Let S be the set returned by the algorithm for the k-pebble game with backmoves on a quantified constraint sentence Φ.

- For any winning strategy T for this game (on Φ), it holds that $T \subseteq S$.
- If $S = \emptyset$, then there is no winning strategy for this game (on Φ).

We consider one more pebble game, the *truth pebble game*, which characterizes the truth of a quantified constraint sentence.

Definition 3.11: A winning strategy for the truth pebble game (on $\Phi = Q_{v_1} v_1 \dots Q_{v_n} v_n \phi$) is a set S of projective homomorphisms, each having domain of the form $\{v_1, v_2, \dots, v_i\}$ for some $i \geq 0$, that satisfy the following conditions.

(1) The partial function with empty domain is an element of S.

(2') If $f \in S$ and g is a restriction of f to a set of the form $\{v_1, v_2, \dots, v_i\}$, then $g \in S$.

(3') If $f \in S$, $\mathsf{dom}(f) = \{v_1, v_2, \dots, v_i\}$, and $i < n$, then f can be $(v_{i+1}, Q_{v_{i+1}})$-extended in S.

□

Proposition 3.12: A quantified constraint sentence Φ is true if and only if there exists a winning strategy for the truth pebble game on Φ.

We now show that the maximal winning strategy for the k-pebble game with backmoves fully supports any winning strategy for the truth pebble game, in the following sense.

Proposition 3.13: Let Φ be a quantified constraint sentence. Let S be the strategy computed by the algorithm for the k-pebble game with backmoves (on Φ). If there exists a winning strategy T for the truth pebble game (on Φ), then S is nonempty and contains all restrictions of operations in T to domains of size less than or equal to k.

Proof. We prove that, at each stage of the algorithm, all such restrictions of operations in T are contained in S. It is clear that, after step (1) is performed, this holds. We thus show that if this holds, then no such restriction of an operation in T can be removed from S by one of the removal rules in step (2), as follows.

In each of the following arguments, we suppose that $f \in S$ is the restriction of an operation $h \in T$.

- All restrictions of f are restrictions of h, and are hence all contained in S.
- Let $v_j \in V_\Phi$ with $|\mathsf{dom}(f)| < k$ and $\mathsf{dom}(f) <_\Phi v_j$. We make use of properties (2') and (3') of Definition 3.11. By property (2'), there exists a restriction $h_1 \in T$ of h such that $\mathsf{dom}(f) \subseteq \mathsf{dom}(h_1)$ but $v_j \notin \mathsf{dom}(h_1)$. By possibly repeated application of property (3'), one can obtain $h_2 \in T$ with domain $\{v_1, \dots, v_{j-1}\}$ such that h_2 extends h_1, and f is a restriction of h_2. Since h_2 can be (v_j, Q_{v_j})-extended in T, the operation f can be (v_j, Q_{v_j})-extended in S.
- Let $v \in V_\Phi$ with $|\mathsf{dom}(f)| < k$ and $\mathsf{dom}(f) \not<_\Phi v$. We have that h is defined on v, and hence the restriction of h to $\mathsf{dom}(f) \cup \{v\}$ is contained in S.

□

It is straightforward to verify that, for an instance Φ of QCSP($\mathbf{B}$) and a winning strategy S for one of the pebble games considered in this subsection, the closure of S under the surjective polymorphisms of $\mathbf{B}$ is also a winning strategy for the pebble game. In what follows, we will tacitly assume that winning strategies are closed under such polymorphisms.

C. Algorithm

In this subsection, we present our algorithm for Guarded Ord-Horn, and prove its correctness.

Before presenting our algorithm, we introduce a couple of notions. Let us say that a winning strategy S for the k-pebble game with backmoves (on Φ) *implies the equality* $u = v$, where u, v are distinct variables in V_Φ, if every operation $f \in$

S with domain $\mathsf{dom}(f) = \{u, v\}$ has the property that $f(u) = f(v)$. Our algorithm will detect implied equalities and simplify quantified sentences; the notion of simplification used is as follows. Let $u <_\Phi v$ be two variables in V_Φ. We use $\Phi[u \leftarrow v]$ to denote the sentence obtained from Φ by eliminating v and its quantifier from the quantifier prefix, and by replacing every instance of v in the quantifier-free part with u.

> Algorithm for QCSP($\mathbf{B}$) where $\mathbf{B}$ is a GOH-structure.
>
> Input: a quantified constraint sentence Φ
>
> 1. Run the algorithm for the k-pebble game with backmoves on $\overline{\Phi}$, where k is the arity of the QNUF from Corollary 3.7; let S be the result.
>
> 2. If $S = \emptyset$, return FALSE.
>
>   Otherwise, check to see if S implies any equalities. If it does, let $u = v$ be an implied equality, with $u <_\Phi v$; replace Φ with $\Phi[u \leftarrow v]$; and return to step 1.
>
> 3. Return TRUE.

Clearly, this algorithm runs in polynomial time: each time it performs a replacement of Φ, one variable is eliminated (and the sentence is shortened), and so it loops at most $|V_\Phi|$ times; and, as previously discussed, the algorithm for the k-pebble game with backmoves runs in time polynomial in the size of the input sentence. We thus turn to discuss the correctness of the algorithm.

When the algorithm detects an implied equality $u = v$, with $u <_\Phi v$, the variable v must be existentially quantified, since a function with domain $\{u\}$ can be (v, Q_v)-extended in a winning strategy for the pebble game. Also, in this case, by Proposition 3.13, all functions f contained in winning strategies for the truth pebble game on $\overline{\Phi}$ that are defined on $\{u, v\}$ set $f(u) = f(v)$. Since a winning strategy for the truth pebble game on Φ must be a winning strategy for the truth pebble game for $\overline{\Phi}$, this property also holds for all functions f contained in winning strategies for the truth pebble game on Φ, and so the sentence $\Phi[u \leftarrow v]$ is true if and only if the sentence Φ is true.

We have thus shown that the replacement step preserves the truth of the sentence; it follows by Proposition 3.13 that if the algorithm returns "FALSE", then the input sentence was indeed false. It remains to show that if the algorithm returns "TRUE", then the sentence is true. To demonstrate this, it suffices to provide a proof of the following theorem.

Theorem 3.14: Let $\mathbf{B}$ be a GOH structure, let k be the arity of the QNUF from Corollary 3.7, and let Φ be a quantified constraint sentence that is an instance of QCSP($\mathbf{B}$). If the sentence $\overline{\Phi}$ has a winning strategy for the k-pebble game with backmoves that does not imply any equalities, then the sentence Φ has a winning strategy for the truth pebble game.

We establish Theorem 3.14 via a sequence of lemmas. In what follows, we assume that the hypotheses of Theorem 3.14 are in effect. We will make use of the following notion. Relative to a sentence Φ, let us say that a projective homomorphism f is *exinjective* ("existentially injective") when for all pairs of variables $u, v \in \mathsf{dom}(f)$, if $u <_\Phi v$ and $Q_v = \exists$, then $f(u) \neq f(v)$. We say that a set of projective homomorphisms is *exinjective* if all of the projective homomorphisms it contains are exinjective.

Lemma 3.15: If the sentence $\overline{\Phi}$ has a winning strategy S for the k-pebble game with backmoves that does not imply any equalities, then this sentence has an exinjective winning strategy for the k-pebble game.

In the proof of this lemma, we will use the notation $\overline{c}$, where $c \in \mathbb{Q}$, to denote the k-tuple $(c, \ldots, c) \in \mathbb{Q}^k$ having all entries equal to c. We will also make use of the following type of operation. For a constant $c \in \mathbb{Q}$, define $p_c : \mathbb{Q}^2 \to \mathbb{Q}$ to be a surjective operation such that $p_c(x, y) < p_c(x', y')$ if and only if either (1) $x < x'$, or (2) $x = x' = c$ and $y < y'$; it is easy to see that such operations exist.

See Fig. 1 for an illustration of such an operation. In this illustration, if we link $(x, y), (u, v) \in \mathbb{Q}^2$ by an arc from (x, y) to (u, v) then this means that $f(x, y) < f(u, v)$; if (x, y) is linked to (u, v) by an undirected line, then $f(x, y) = f(u, v)$.

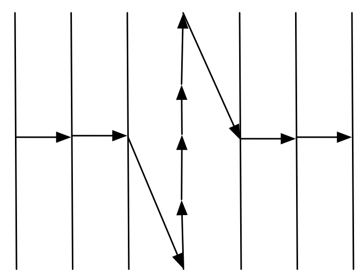

Fig. 1. An illustration of an operation p_c.

Observe that all such operations p_c preserve all basic OH formulas. We present the following sub-lemma.

Lemma 3.16: Let R be a relation defined by a basic OH formula and $c \in \mathbb{Q}$ be a constant. Then p_c preserves R.

Proof. In what follows, we write ϕ_R for a GOH formula defining R. Moreover, t_x denotes a coordinate of $t \in R$ that corresponds to a variable x occurring in ϕ_R.

A basic OH formula is in one of four forms. If ϕ_R is an equality, then R is preserved by all functions and hence by p_c. We now consider the case where ϕ_R is of the form $x \leq y$. Let $t, u \in R$. Then, by the definition of p_c, it is not hard to see that $p_c(t_x, u_x) \leq p_c(t_y, u_y)$. Thus $p_c(t, u)$ is in R and p_c preserves R.

If ϕ_R is of the form $(x_1 \neq y_1 \vee \ldots \vee x_p \neq y_p)$ and $t \in R$, then there is $i \in [p]$ such that $t_{x_i} \neq t_{y_i}$. Assume without loss of generality that $t_{x_i} < t_{y_i}$. Let $u \in R$ be any tuple. We now show that $p_c(t,u)$ is in R and thereby that R is preserved by p_c. To see this observe that $p_c(t_{x_i}, u_{x_i}) < p_c(t_{y_i}, u_{y_i})$.

Finally, let ϕ_R be of the form $(x_1 \neq x_2 \vee \ldots \vee x_1 \neq x_q) \vee (x_1 < y_1) \vee (y_1 \neq y_2 \vee \ldots \vee y_1 \neq y_{q'})$. If $t \in R$, then we have one of the following situations:

- there is $i \in [q]$ such that $t_{x_1} \neq t_{x_i}$;
- there is $i \in [q']$ such that $t_{y_1} \neq t_{y_i}$;
- $t_{x_1} < t_{y_1}$.

Let $u \in R$. To show that $p_c(t,u) \in R$ regardless of the reason why $t \in R$ we use the same argument as in the case where ϕ_R is of the form $(x_1 \neq y_1 \vee \ldots \vee x_p \neq y_p)$.

For each type of a basic OH formula ϕ_R we showed that R is preserved by p_c. Thus we proved the lemma. $\square$

Proof. (Lemma 3.15) Let S' be the set containing all projective homomorphisms in S that are exinjective. We will show that S' gives the desired exinjective winning strategy. Let us consider the properties of Definition 3.8. It is clear that S' inherits properties (1) and (2) from S. Also, S' inherits the property (3) in the case that $Q_v = \forall$. It thus suffices to prove the following claim.

Claim: if $f \in S$ is exinjective, $|dom(f)| < k$, $v \in V_\Phi$, $Q_v = \exists$, and $\mathsf{dom}(f) <_\Phi v$, then there is an exinjective extension $g \in S$ of f with $\mathsf{dom}(g) = \mathsf{dom}(f) \cup \{v\}$.

We prove the claim by induction on $|\mathsf{dom}(f)|$. For $|dom(f)| = 0$, the claim is obvious. For $|dom(f)| = 1$, the claim follows from the hypothesis that S does not imply any equalities. So, we suppose that $|dom(f)| \geq 2$. As S is a winning strategy for the k-pebble game, there exists an extension $h \in S$ of f with $\mathsf{dom}(h) = \mathsf{dom}(f) \cup \{v\}$. Let $I = \{w \in \mathsf{dom}(f) \mid h(v) = f(w)\}$. We now consider two cases; it can be remarked that if $I = \emptyset$, then one can simply take $g = h$.

Case $I \neq \mathsf{dom}(f)$: let $u \in \mathsf{dom}(f) \setminus I$, that is, let $u \in \mathsf{dom}(f)$ be a variable such that $h(v) \neq f(u)$. Let h_1 be the restriction of h to $\mathsf{dom}(f) \setminus \{u\}$. By the induction hypothesis, there exists an exinjective extension h_2 of h_1 whose domain $\mathsf{dom}(h_2)$ is equal to $\mathsf{dom}(h) \setminus \{u\} = (\mathsf{dom}(f) \cup \{v\}) \setminus \{u\}$. By applying a backmove to h_2, we obtain an extension h_3 of h_2 with $\mathsf{dom}(h_3) = \mathsf{dom}(h)$.

Consider the function $p_{h(v)}(h, h_3)$. Recall that the definition of the polymorphism p_c, where $c \in \mathbb{Q}$, is above Lemma 3.16. The function h is equal to $h(v)$ at $I \cup \{v\}$; on $I \cup \{v\}$, the function h_3 is equal to $h(v)$ on I, but equal to a value different from $h(v)$ at v. Hence, the function $p_{h(v)}(h, h_3)$ is exinjective, but there exists an automorphism β such that $\beta(p_{h(v)}(h, h_3)) = f$ on $\mathsf{dom}(f)$, and so we can take $g = \beta(p_{h(v)}(h, h_3))$.

Case $I = \mathsf{dom}(f)$: let us assume that h is equal to 0 everywhere on its domain; for other values, the same argument will apply under translation by an automorphism. Observe that, since f is exinjective, all variables of $\mathsf{dom}(f)$, except for possibly the earliest one, are universally quantified. Let y be the latest occurring variable in $\mathsf{dom}(f)$, which must be universally quantified. Fix $\alpha > 0$ to be a positive constant. By making use of automorphisms and the fact that S is a winning strategy, we can obtain a homomorphism $a_1 \in S$ with $\mathsf{dom}(a_1) = \mathsf{dom}(f)$ where $a_1(y) = -\alpha$ and $a_1 = \alpha$ elsewhere. Observe that a_1 is exinjective, and hence any restriction thereof is also exinjective. We claim that there is an extension c_1 of a_1 with $\mathsf{dom}(c_1) = \mathsf{dom}(a_1) \cup \{v\}$ where $c_1(v) \notin \{-\alpha, \alpha\}$. Let $b \in S$ be an extension of a_1 with $\mathsf{dom}(b) = \mathsf{dom}(a_1) \cup \{v\}$. If b does not satisfy the desired property, then $b(v) \in \{-\alpha, \alpha\}$. Let $b_1 \in S$ be the restriction of a_1 to the set $J = \{w \in \mathsf{dom}(a_1) \mid a_1(w) = b(v)\}$. By the induction hypothesis, b_1 can be extended to an exinjective $b_1' \in S$ with $\mathsf{dom}(b_1') = \mathsf{dom}(b_1) \cup \{v\}$. By backpebbling, b_1' can be extended to an operation $b_2' \in S$ with $\mathsf{dom}(b_2') = \mathsf{dom}(f) \cup \{v\}$. Consider the function $p_{b(v)}(b, b_2')$. The function b is equal to $b(v)$ at $J \cup \{v\}$; on that set, the function b_2' is equal to $b(v)$ on J, and to a different value at v. We can thus take $c_1 = p_{b(v)}(b, b_2')$.

After application of an automorphism, we have obtained an operation c_1 having $\mathsf{dom}(c_1) = \mathsf{dom}(f) \cup \{v\}$ where $c_1(y) = -\alpha$, $c_1(v) \notin \{-\alpha, \alpha\}$, and $c_1 = \alpha$ elsewhere. By a similar argument, we can obtain an operation c_2 having $\mathsf{dom}(c_2) = \mathsf{dom}(f) \cup \{v\}$ where $c_2(y) = \alpha$, $c_2(v) \notin \{-\alpha, \alpha\}$, and $c_2 = -\alpha$ elsewhere.

Suppose that one of the values $c_1(v), c_2(v)$ is strictly greater than α, and that the other is strictly less than $-\alpha$. Let $d_1, d_2 \in \{c_1, c_2\}$ be such that $d_1(v) > \alpha$ and $d_2(v) < -\alpha$. We can find automorphisms γ_1, γ_2 such that $\gamma_1(d_1)$ is positive on v, and negative elsewhere; and, $\gamma_2(d_2)$ is positive everywhere. Consider the function $g = q(\gamma_1(d_1), \gamma_2(d_2), h, \ldots, h)$, where q is the QNUF from the formulation of Theorem 3.14. On $\mathsf{dom}(f)$, this g is equal to $q(\overline{0})$, since on these variables the values that q is applied to yield the main value 0. On the other hand, $g(v)$ is not equal to $q(\overline{0})$, since on v both $\gamma_1(d_1), \gamma_2(d_2)$ are positive, and hence there is no main value.

In other cases, we will demonstrate that there are automorphisms γ_1, γ_2 each of which fixes both $-\alpha$ and α such that $g = q(\gamma_1(d_1), \gamma_2(d_2), h, \ldots, h)$ is not equal to $q(\overline{0})$ on v; this suffices, since for such automorphisms g is equal to $q(\overline{0})$ elsewhere. If both $d_1(v), d_2(v)$ are strictly greater than α, then one can take both γ_1, γ_2 to be the identity automorphisms, as then the tuple q is applied to will have no main value. Similarly, if both $d_1(v), d_2(v)$ are strictly less than $-\alpha$, then one can take both γ_1, γ_2 to be the identity automorphisms. The remaining case is where one or both of $d_1(v), d_2(v)$ is in the interval $(-\alpha, \alpha)$. Suppose that $d_1(v) \in (-\alpha, \alpha)$. The automorphism γ_2 can be defined so that $\gamma_2(d_2(v)) \neq 0$. Then, γ_1 can be defined so that $\gamma_1(d_1(v))$ is positive if $\gamma_2(d_2(v))$ is positive, and negative if $\gamma_2(d_2(v))$ is negative. The argumentation for the subcase where $d_2(v) \in (-\alpha, \alpha)$ is similar. $\square$

Lemma 3.17: If the sentence $\overline{\Phi}$ has an exinjective winning strategy for the k-pebble game then this sentence has an exinjective winning strategy for the truth pebble game.

Proof. Let S be an exinjective winning strategy for the k-pebble game. Let T be the set containing all exinjective functions f from a subset of V_Φ to $\mathbb{Q}$ such that every restriction of f to a set of size $\leq k$ is contained in S. Every function in T is a projective homomorphism by Theorem 2.2. Clearly, T is non-empty, as $S \subseteq T$, and is closed under restriction. We will prove the following claim.

Claim: for every function $f \in T$ and variable $v \in V_\Phi$ with $\mathsf{dom}(f) <_\Phi v$, the function f can be (v, Q_v)-extended in T.

This claim suffices to give the lemma, as then the set of functions in T having domain of the form $\{v_1, v_2, \ldots, v_i\}$ forms the desired exinjective winning strategy for the truth pebble game.

The claim is clear when $Q_v = \forall$ by the definition of T, so let $Q_v = \exists$. We prove the claim by induction on $|\mathsf{dom}(f)|$. The base case $|\mathsf{dom}(f)| < k$ is clear by definition of T, so suppose that $|\mathsf{dom}(f)| \geq k$. Let $u_1, \ldots, u_k$ be distinct elements from $\mathsf{dom}(f)$, and for each $i \in [k]$ define f_i to be the restriction of f to $\mathsf{dom}(f) \setminus \{u_i\}$. Each f_i has, by induction, an exinjective extension $g_i \in T$ defined on $(\mathsf{dom}(f) \setminus \{u_i\}) \cup \{v\}$. Let g be the function on $\mathsf{dom}(f) \cup (v)$ defined by $g(u) = q(\overline{f(u)})$ for all $u \in \mathsf{dom}(f)$, and $g(v) = q(g_1(v), \ldots, g_k(v))$. By the QNUF identities and the fact that each g_i is in T, we obtain that g is in T. Since each of the g_i is exinjective, for all $i, j \in [k]$ with $j \neq i$, it holds that $g_j(v) \neq g_j(u_i) = f(u_i)$. It follows that the tuple $(g_1(v), \ldots, g_k(v))$ cannot have as main value any of the values $f(u_1), \ldots, f(u_k)$. Hence, the function g is exinjective. □

Lemma 3.18: If the sentence $\overline{\Phi}$ has an exinjective winning strategy for the truth pebble game then that strategy is a winning strategy for the truth pebble game on Φ.

Proof. Let $R(t)$ be an atomic formula from the quantifier-free part of Φ, where t is a tuple of variables, and let $\phi(t)$ be the normal GOH formula for $R(t)$. Let $\phi'(t)$ be the guards formula of $\phi(t)$; the formula $\phi'(t)$ is the formula for $\overline{R(t)}$. The formula $\phi(t)$ can be viewed as the conjunction of GOH formulas of types 1 and 3. We show that each such formula is satisfied by any $f \in S$ whose domain contains the variables of t, which suffices.

A basic OH formula is satisfied by f because f satisfies the corresponding identical subformula in $\phi'(t)$.

Now consider a formula

$$(x_1 \leq y_1) \wedge \ldots \wedge (x_m \leq y_m) \wedge (x_1 \neq y_1 \vee \ldots \vee x_m \neq y_m \vee \psi)$$

of type 3. It suffices to show that each inequality $x_i \leq y_i$ is satisfied in a way that x and y are set to different values. Let $u, v \in V_\Phi$ be such that $\{u, v\} = \{x_i, y_i\}$ and $u <_\Phi v$. Note that x_i, y_i are distinct variables, since $\phi(t)$ is a normal GOH formula. We claim that $Q_v = \exists$, which suffices, as the strategy S is exinjective.

Suppose for a contradiction that $Q_v = \forall$. Then the restriction of f to $\{u\}$ can be extended to a function $g \in S$ with $\mathsf{dom}(g) = \{u, v\}$ such that g does not satisfy $x_i \leq y_i$, contradicting that S is a winning strategy for $\overline{\Phi}$. □

We can now conclude our main polynomial-time decidability result.

Theorem 3.19: Let $\mathbf{B}$ be a GOH structure. The problem QCSP($\mathbf{B}$) is polynomial-time decidable.

Proof. If the algorithm returns "FALSE", falsity of the input sentence follows directly from Proposition 3.13. If the algorithm returns "TRUE", truth of the input sentence follows directly from Theorem 3.14 and Proposition 3.12. □

Acknowledgment

Hubie Chen is supported by the Spanish program "Ramon y Cajal" and MICINN grant TIN2010-20967-C04-02.

References

[1] D. Kozen, *Theory of Computation*. Springer, 2006.
[2] S. Basu, R. Pollack, and M.-F. Roy, *Algorithms in Real Algebraic Geometry, 2nd edition*. Springer-Verlag, 2009.
[3] R. H. Möhring, M. Skutella, and F. Stork, "Scheduling with and/or precedence constraints," *SIAM J. Comput.*, vol. 33, no. 2, pp. 393–415, 2004.
[4] M. Broxvall and P. Jonsson, "Point algebras for temporal reasoning: Algorithms and complexity," *Artif. Intell.*, vol. 149, no. 2, pp. 179–220, 2003.
[5] B. Aspvall, M. F. Plass, and R. E. Tarjan, "A linear-time algorithm for testing the truth of certain quantified boolean formulas," *Inf. Process. Lett.*, vol. 8, no. 3, pp. 121–123, 1979.
[6] M. Karpinski, H. K. Büning, and P. H. Schmitt, "On the computational complexity of quantified horn clauses," in *CSL*, 1987, pp. 129–137.
[7] D. Kozen, "Positive first-order logic is NP-complete," *IBM Journal of Research and Development*, vol. 25, no. 4, pp. 327–332, 1981.
[8] P. van Beek, "Reasoning about qualitative temporal information," *Artificial Intelligence*, vol. 58, pp. 297–326, 1992.
[9] M. Fisher, D. Gabbay, and L. Vila, Eds., *Handbook of Temporal Reasoning in Artificial Intelligence*. Elsevier, 2005.
[10] M. Bodirsky and H. Chen, "Qualitative temporal and spatial reasoning revisited," in *CSL'07*, 2007, pp. 194–207.
[11] B. Nebel and H.-J. Bürckert, "Reasoning about temporal relations: A maximal tractable subclass of Allen's interval algebra," *JACM*, vol. 42, no. 1, pp. 43–66, 1995.
[12] P. Balbiani, J.-F. Condotta, and L. F. del Cerro, "Tractability results in the block algebra," *J. Log. Comput.*, vol. 12, no. 5, pp. 885–909, 2002.
[13] M. Bodirsky and J. Kára, "The complexity of temporal constraint satisfaction problems," *Accepted for publication in the Journal of the ACM*, 2009, an extended abstract appeared in the proceedings of STOC'08.
[14] M. Bodirsky and H. Chen, "Quantified equality constraints," *SIAM Journal on Computing*, vol. 39, no. 8, pp. 3682–3699, 2010.
[15] M. Wrona, "Syntactically characterizing local-to-global consistency in ord-horn," in the proceedings of 18th International Conference on Principles and Practice of Constraint Programming.
[16] L. Barto and M. Kozik, "Constraint satisfaction problems of bounded width," in *Proceedings of FOCS'09*, 2009.
[17] K. Subramani, "On a decision procedure for quantified linear programs," *Ann. Math. Artif. Intell.*, vol. 51, no. 1, pp. 55–77, 2007.
[18] W. Charatonik and M. Wrona, "Tractable quantified constraint satisfaction problems over positive temporal templates," in *LPAR*, 2008, pp. 543–557.
[19] ——, "Quantified positive temporal constraints," in *CSL*, 2008, pp. 94–108.
[20] W. Hodges, *A shorter model theory*. Cambridge: Cambridge University Press, 1997.
[21] A. Bulatov, A. Krokhin, and P. G. Jeavons, "Classifying the complexity of constraints using finite algebras," *SIAM Journal on Computing*, vol. 34, pp. 720–742, 2005.
[22] M. Bodirsky and H. Chen, "Oligomorphic clones," *Algebra Universalis*, vol. 57, no. 1, pp. 109–125, 2007.
[23] V. Dalmau, P. G. Kolaitis, and M. Y. Vardi, "Constraint satisfaction, bounded treewidth, and finite-variable logics," in *Proceedings of CP'02*, 2002, pp. 310–326.
[24] H. Chen and V. Dalmau, "From pebble games to tractability: An ambidextrous consistency algorithm for quantified constraint satisfaction," in *CSL*, 2005, pp. 232–247.