Constraint Patterns for Tractable Ontology-Mediated Queries with Datatypes

André Hernich, Julio Lemos, and Frank Wolter

University of Liverpool, UK
{hernich,jlemos,wolter}@liverpool.ac.uk

Abstract. Adding datatypes to ontology-mediated conjunctive queries (OMQs) often makes query answering hard. This applies, in particular, to datatypes with non-unary predicates. In this paper we propose a new, non-uniform way, of analysing the data-complexity of OMQ answering with datatypes containing higher-arity predicates. We aim at a classification of the patterns of datatype atoms in OMQs into those that can occur in non-tractable OMQs and those that only occur in tractable OMQs. Our main result is a P/coNP-dichotomy for OMQs over DL-Lite TBoxes and rooted CQs using the datatype $(\mathbb{Q}, \leq)$. The proof employs a recent dichotomy result by Bodirsky and Kara for temporal constraint satisfaction problems.

1 Introduction

In recent years, querying data using ontologies has become one of the main applications of description logics (DLs). The general idea is that an ontology is used to enrich incomplete and heterogeous data with a semantics and with background knowledge, thus serving as an interface for querying data and allowing to derive additional facts. In this area called ontology-based data management (OBDM) one of the main research problems is to identify ontology languages and queries for which query answering scales to large amounts of data [11, 6]. In DL, ontologies take the form of a TBox, data is stored in an ABox, and the most important class of queries are conjunctive queries (CQs). A basic observation regarding this setup is that even for DLs from the DL-Lite family that have been designed for tractable OBDM the addition of datatypes to the TBoxes or the CQs typically leads to non-tractable query answering problems [2, 20]. As a consequence of this, the use of datatypes in ontology and query languages for OBDM has been severely restricted. For example, the OWL2 QL standard admits datatypes with unary predicates only. Nevertheless, in applications there is clearly a need for more expressive datatypes and, in particular, datatypes with predicates of higher arity.

The aim of this paper is to revisit OBDM with expressive datatypes from a new, non-uniform, perspective. Instead of the standard approach that aims at the definition of DLs $\mathcal{L}$ and query languages $\mathcal{Q}$ such that for any TBox $\mathcal{T}$ in $\mathcal{L}$ and any query q in $\mathcal{Q}$, answering q under $\mathcal{T}$ is tractable in data-complexity we now aim at describing the complexity of query answering with datatypes at a more fine-grained level by taking into account the way in which datatype atoms can occur in queries. In more detail, an ontology-mediated query (OMQ) Q is a triple $Q = (\mathcal{T}, q, \mathcal{D})$ consisting of a TBox $\mathcal{T}$[1], a CQ q, and a datatype $\mathcal{D}$ (that we identify with a relational structure $(D, R_1, \ldots)$). The TBox $\mathcal{T}$ in Q can refer to the datatype $\mathcal{D}$ using existential restrictions $\exists U$ where U is an attribute. The CQ q can contain atoms using attributes and relations R_i from $\mathcal{D}$. We aim at a classification of the data complexity of query answering with OMQs $(\mathcal{T}, q, \mathcal{D})$ based on the *datatype pattern* $\mathsf{dtype}(q)$ of q that consists of all atoms using the relations in $\mathcal{D}$. To illustrate, the CQ q uses the datatype $\mathcal{D} = (\mathbb{Q}, \leq)$ and asks for all rectangles whose width is larger than its height:

$$q(x) \quad \leftarrow \quad \mathsf{rectangle}(x) \wedge \mathsf{height}(x, u) \wedge \mathsf{width}(x, v) \wedge u \leq v.$$

Thus, height and width are attributes and the datatype pattern $\mathsf{dtype}(q)$ of q is $u \leq v$. The following CQ q' uses $(\mathbb{Q}, \leq)$ as well and asks for all countries having a neigbour to the west that is larger and a neighbour to the east that is smaller:

$$\begin{aligned} q'(x) \leftarrow\ & \mathsf{country}(x) \wedge \mathsf{westneighbour}(x, y_1) \wedge \mathsf{eastneighbour}(x, y_2) \wedge \\ & \mathsf{area}(x, u) \wedge \mathsf{area}(y_1, u_1) \wedge \mathsf{area}(y_2, u_2) \wedge \\ & (u \leq u_1) \wedge (u_2 \leq u). \end{aligned}$$

Its datatype pattern $\mathsf{dtype}(q')$ is $(u \leq u_1) \wedge (u_2 \leq u)$.

Our main results assume that either the CQ q is a rooted CQ (a CQ in which each variable is connected to an answer variable not in $\mathsf{dtype}(q)$) or that the chase of the TBox under consideration is finite for all ABoxes. Under this assumption, we first show a close link between the complexity of query evaluation for OMQs with datatype $\mathcal{D}$ and the evaluation problem for positive existential sentences in the structure $\bar{\mathcal{D}}$ in which every relation R has been replaced by its complement (thus, $\bar{\mathcal{D}} = (\mathbb{Q}, >)$ if $\mathcal{D} = (\mathbb{Q}, \leq)$). In more detail, we show that evaluating an OMQ with datatype $\mathcal{D}$ and a datatype pattern with at most k atoms is polynomially reducible to the complement of the problem $\mathsf{PE}_k(\bar{\mathcal{D}})$ of evaluating positive existential formulas in k-CNF in $\bar{\mathcal{D}}$. Conversely, $\mathsf{PE}_k(\bar{\mathcal{D}})$ is polynomially reducible to the complement of evaluating a single fixed OMQ (depending only on k) with datatype $\mathcal{D}$ and a datatype pattern with nk atoms, where n is the number of relations in $\mathcal{D}$.

Basic complexity results can be obtained easily from this observation. For example, from the fact that $\mathsf{PE}_1(\mathcal{D})$ is tractable for $\mathcal{D} \in \{(\mathbb{Z}, <), (\mathbb{Z}, \leq), (\mathbb{Q}, <), (\mathbb{Q}, \leq)\}$, we obtain that evaluating OMQs $(\mathcal{T}, q, \mathcal{D})$ in which q is a rooted CQ whose datatype pattern is a *singleton* is tractable. Conversely, from the fact that $\mathsf{PE}_2(\mathbb{Q}, >)$ is non-tractable we obtain an intractable OMQ $(\mathcal{T}, q, \mathcal{D})$ with datatype $\mathcal{D} = (\mathbb{Q}, \leq)$ and a rooted CQ q whose datatype pattern consists of two atoms.

Our main result is a P/coNP-dichotomy for OMQs using the datatype $\mathcal{D} = (\mathbb{Q}, \leq)$. Namely, we show that for any datatype pattern q_0 over $(\mathbb{Q}, \leq)$ exactly one of the following two conditions holds (unless P=coNP):

[1] The results presented in this paper do not depend on the particular tractable DL we extend with datatypes. To prove the lower bounds we require that the TBox is given in a DL containing DL-Lite$_{\text{core}}$; the upper bounds can be proved for all standard Horn-DLs for which CQ evaluation is in PTime.

– Evaluating rooted OMQs $(\mathcal{T}, q, \mathcal{D})$ with $\mathsf{dtype}(q) = q_0$ is in PTime.
– There exists a rooted OMQ $(\mathcal{T}, q, \mathcal{D})$ with $\mathsf{dtype}(q) = q_0$ whose evaluation problem is coNP-hard.

The proof uses a recent dichotomy result by Bodirsky and Kára for temporal constraint satisfaction problems [7] and provides a sufficient and necessary condition for the evaluation problem to be in PTime that can be checked in linear time.

Related Work Expressive DLs with datatypes (or concrete domains) have been introduced in [4] and studied extensively [15]. In the context of tractable DLs, reasoning with datatypes has been studied in [3, 17] for $\mathcal{EL}$ and in [19, 20, 2] for DL-Lite. These works focus on finding ontology languages for which typical reasoning tasks are tractable. In contrast, here we start with ontology languages for which query answering is intractable in general, and aim at a complexity classification of query answering guided by the datatype pattern. Our methodology is closely related to recent work relating OBDM to constraint satisfaction problems [16, 5, 13]. However, here we classify datatype patterns according to the data-complexity of evaluating the OMQs containing them, whereas in [16] TBoxes are classified according to the data-complexity of OMQs containing them, and in [5] OMQs themselves are classified according to their data-complexity. Consequently, here we establish a link to temporal constraint satisfiaction [7] whereas the work mentioned above establishes a link to standard constraint satisfaction and the Feder-Vardi conjecture [12, 10].

2 Preliminaries

A *datatype* is a tuple $\mathcal{D} = (D, R_1, R_2, \dots)$, where D is a non-empty set and $R_1, R_2, \dots$ are relations on D. We call $\mathrm{dom}(\mathcal{D}) := D$ the *domain* of $\mathcal{D}$. To simplify the presentation, we will not distinguish between a relation R_i and its name (i.e., we use R_i both as a relation and as the name of the relation R_i). The *complement* of $\mathcal{D}$, denoted by $\bar{\mathcal{D}}$, is obtained by replacing each k-ary relation R_i in $\mathcal{D}$ by its complement $\bar{R}_i := D^k \setminus R_i$. For example, we have $\overline{(\mathbb{Q}, \leq)} = (\mathbb{Q}, >)$.

We assume countably infinite and mutually disjoint sets of *concept names*, *role names*, *attribute names*, and *individual names*. Concept names will typically be denoted by A, role names by P, attribute names by U, and individual names by a, b, c. The logic DL-Lite$_{\text{core}}$ and Horn DLs such as Horn-$\mathcal{ALCHIQ}$ are defined as usual [11, 1, 14, 9]. We consider the extension L^{attrib} of such DLs L in which the existential restrictions $\exists U$ for attribute names U can occur in exactly the same places in concepts and concept inclusions as concept names can occur in L. Unless stated otherwise, TBoxes range over L^{attrib} TBoxes, where L is DL-Lite$_{\text{core}}$ or any standard Horn-DL with data complexity for CQs in PTime.

Let $\mathcal{D}$ be a datatype. A *$\mathcal{D}$-ABox* consists of *assertions* of the form $A(a)$, $P(a, b)$, and $U(a, u)$, where A is a concept name, P is a role name, U is an attribute name, a, b are individual names, and $u \in \mathrm{dom}(\mathcal{D})$. A *$\mathcal{D}$-knowledge base ($\mathcal{D}$-KB)* is a pair $(\mathcal{T}, \mathcal{A})$ consisting of a TBox $\mathcal{T}$ and a $\mathcal{D}$-ABox $\mathcal{A}$.

An *interpretation* $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$ over a datatype $\mathcal{D}$ consists of a non-empty domain $\Delta^{\mathcal{I}} = \Delta^{\mathcal{I}}_{\mathsf{ind}} \cup \mathrm{dom}(\mathcal{D})$ and an interpretation function $\cdot^{\mathcal{I}}$ that assigns to each concept name A a set $A^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}}_{\mathsf{ind}}$, to each role name P a relation $P^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}}_{\mathsf{ind}} \times \Delta^{\mathcal{I}}_{\mathsf{ind}}$, and to each attribute name U a relation $U^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}}_{\mathsf{ind}} \times \mathrm{dom}(\mathcal{D})$. The elements in $\Delta^{\mathcal{I}}_{\mathsf{ind}}$ are called *individuals*, whereas the elements in $\mathrm{dom}(\mathcal{D})$ are called *data values*. We assume that $\Delta^{\mathcal{I}}_{\mathsf{ind}}$ and $\mathrm{dom}(\mathcal{D})$ are disjoint. Throughout this paper, we make the *standard name assumption*: if $\mathcal{I}$ is an interpretation, then we set $a^{\mathcal{I}} := a$ for all individual names a. We also set $u^{\mathcal{I}} := u$ for each $u \in \mathrm{dom}(\mathcal{D})$, and $R^{\mathcal{I}} := R$ for each relation R of $\mathcal{D}$. The interpretation $\mathcal{I}$ induces the interpretations $C^{\mathcal{I}}$ and $S^{\mathcal{I}}$ for each complex concept C and complex role S in the standard way.

We say that $\mathcal{I}$ is a *model* of a KB $(\mathcal{T}, \mathcal{A})$ if $a^{\mathcal{I}} \in A^{\mathcal{I}}$, $(a^{\mathcal{I}}, b^{\mathcal{I}}) \in P^{\mathcal{I}}$, and $(a^{\mathcal{I}}, u^{\mathcal{I}}) \in U^{\mathcal{I}}$ for all assertions $A(a)$, $P(a, b)$, and $U(a, u)$ in $\mathcal{A}$, and if for every inclusion $X \sqsubseteq Y$ in $\mathcal{T}$ we have $X^{\mathcal{I}} \subseteq Y^{\mathcal{I}}$. A KB $(\mathcal{T}, \mathcal{A})$ is *satisfiable* if it has a model; in this case we say that $\mathcal{A}$ is *satisfiable relative to* $\mathcal{T}$.

We consider *conjunctive queries (CQs)* q (over $\mathcal{D}$) of the form $q(\bar{x}) \leftarrow \varphi$, where $\bar{x}$ is the tuple of *answer variables* of q, and φ is a conjunction of atomic formulas of the form $A(y)$, $P(y, z)$, $U(y, u)$, or $R(u_1, \ldots, u_k)$, where A, P, U, and R range over concept names, role names, attribute names, and relation names in $\mathcal{D}$, respectively; each y, z is a variable; and each $u, u_1, \ldots, u_k$ is a variable. As usual, all variables of $\bar{x}$ must occur in some atom of φ. The *size* $|q|$ of q is the number of atoms in q. The *datatype pattern* of q, denoted by $\mathsf{dtype}(q)$, is the conjunction of all atoms in φ that use a relation in $\mathcal{D}$. The variables that occur in $\mathsf{dtype}(q)$ are called *data variables*. We assume that all data variables occur in some atom $U(\cdot, \cdot)$ outside of $\mathsf{dtype}(q)$. A *match* of q in an interpretation $\mathcal{I}$ is a mapping μ from the variables of φ to $\Delta^{\mathcal{I}}$ such that for each atom $X(t_1, \ldots, t_k)$ of φ we have $(\mu(t_1), \ldots, \mu(t_k)) \in X^{\mathcal{I}}$. A tuple $\bar{c}$ of individual names and data values is an *answer* to q in an interpretation $\mathcal{I}$ if there is a match μ of q in $\mathcal{I}$ such that $\mu(\bar{x}) = \bar{c}$. We denote this by $\mathcal{I} \models q(\bar{c})$. Given a KB $(\mathcal{T}, \mathcal{A})$, we write $\mathcal{T}, \mathcal{A} \models q(\bar{c})$ if $\bar{c}$ is an answer to q in every model of $(\mathcal{T}, \mathcal{A})$.

The *connection graph* of a CQ q is the undirected graph with the variables of q as its vertices and an edge between any two distinct variables if they occur together in an atom of q that does not belong to $\mathsf{dtype}(q)$. We say that q is *rooted* if for every variable y of q the connection graph contains a path from y to an answer variable of q.

We consider *ontology-mediated queries (OMQs)* of the form $Q = (\mathcal{T}, q, \mathcal{D})$, where $\mathcal{D}$ is a datatype, $\mathcal{T}$ is a TBox, and q is a CQ over $\mathcal{D}$. Given a $\mathcal{D}$-ABox $\mathcal{A}$ and a tuple $\bar{c}$, we write $\mathcal{A} \models Q(\bar{c})$ if $(\mathcal{T}, \mathcal{A}) \models q(\bar{c})$. An OMQ $Q = (\mathcal{T}, q, \mathcal{D})$ is *rooted* if q is rooted. The *query evaluation problem* for Q is the problem to decide for given $\mathcal{D}$-ABox $\mathcal{A}$ and $\bar{c}$ whether $\mathcal{A} \models Q(\bar{c})$.

3 Query Evaluation and Positive Existential Sentences

We establish a tight link between the OMQ evaluation problem with datatype $\mathcal{D}$ and the satisfaction problem for positive existential sentences over $\bar{\mathcal{D}}$. To this end we first introduce a variant of the universal (or canonical) model for standard Horn-DL KBs. In contrast to KBs with Horn-DL TBoxes without datatypes, in general there does not exist a universal model for KBs with datatypes.

Example 1. Consider the KB $(\mathcal{T},\mathcal{A})$ with $\mathcal{T}=\{A\sqsubseteq\exists U_1, A\sqsubseteq\exists U_2\}$ and $\mathcal{A}=\{A(a)\}$ and with datatype $(\mathbb{Q},\leq)$. Consider the OMQs $Q_i=(\mathcal{T},q_i,(\mathbb{Q},\leq))$, $i=1,2$, where

$$q_1(x)\leftarrow U_1(x,u_1)\wedge U_2(x,u_2)\wedge u_1\leq u_2 \quad q_2(x)\leftarrow U_1(x,u_1)\wedge U_2(x,u_2)\wedge u_2\leq u_1$$

Clearly $\mathcal{A}\not\models Q_1(a)$ since for the interpretation $\mathcal{I}$ with $U_1^{\mathcal{I}}=\{(a,2)\}$ and $U_2^{\mathcal{I}}=\{(a,1)\}$ we have $\mathcal{I}\not\models q_1(a)$. Also, $\mathcal{A}\not\models Q_2(a)$ since for the interpretation $\mathcal{J}$ with $U_1^{\mathcal{J}}=\{(a,1)\}$ and $U_2^{\mathcal{J}}=\{(a,2)\}$ we have $\mathcal{J}\not\models q_2(a)$. However, there does not exist a model $\mathcal{I}$ of $\mathcal{T}$ and $\mathcal{A}$ such that $\mathcal{I}\not\models q_i(a)$ for both $i=1$ and $i=2$.

The reason that universal models do not exist is that distinct interpretations of attributes can be required to refute the entailment of CQs. The notion of pre-interpretation formalizes this intuition: it fixes the interpretation of concept and roles names but leaves the interpretation of attributes open by adding placeholders for data values (called data nulls) to the set of possible values of attributes. A *pre-interpretation* $\mathcal{J}$ over $\mathcal{D}$ is the same as an interpretation over $\mathcal{D}$ with the exception that attribute names U are now interpreted as sets $U^{\mathcal{J}}\subseteq\Delta_{\mathsf{ind}}^{\mathcal{J}}\times(\mathrm{dom}(\mathcal{D})\cup\Delta_{\mathsf{null}}^{\mathcal{J}})$, where $\Delta_{\mathsf{null}}^{\mathcal{J}}$ is a set of *data nulls* disjoint from $\Delta_{\mathsf{ind}}^{\mathcal{J}}\cup\mathrm{dom}(\mathcal{D})$. The definitions of the interpretations $C^{\mathcal{J}}$ of a concept C and $S^{\mathcal{J}}$ of a role S are extended from interpretations to pre-interpretations in the obvious way. A *pre-model* of a KB is a pre-interpretation that satisfies all assertions and inclusions in the KB.

Pre-interpretations $\mathcal{J}$ can be completed to interpretations by assigning data values to data nulls. A *completion function* f for $\mathcal{J}$ is a mapping $f\colon\Delta_{\mathsf{null}}^{\mathcal{J}}\to\mathrm{dom}(\mathcal{D})$. The *completion* $f(\mathcal{J})$ of $\mathcal{J}$ by f is the interpretation $\mathcal{I}$ obtained from $\mathcal{J}$ by setting $A^{\mathcal{I}}=A^{\mathcal{J}}$ for all concept names A, $P^{\mathcal{I}}=P^{\mathcal{J}}$ for all role names P, and $U^{\mathcal{I}}=(U^{\mathcal{J}}\cap(\Delta_{\mathsf{ind}}^{\mathcal{J}}\times\mathrm{dom}(\mathcal{D})))\cup\{(d,f(v))\mid(d,v)\in U^{\mathcal{J}},v\in\Delta_{\mathsf{null}}^{\mathcal{J}}\}$ for all attribute names U. An interpretation $\mathcal{I}$ is called a *completion* of $\mathcal{J}$ if there exists a completion function f for $\mathcal{J}$ such that $f(\mathcal{J})=\mathcal{I}$.

Using a straightforward modification of the standard chase procedure for Horn-DLs (see, e.g., [9]) one can construct a pre-model $\mathrm{can}(\mathcal{T},\mathcal{A})$ of any satisfiable $\mathcal{D}$-KB $(\mathcal{T},\mathcal{A})$ such that for any CQ q over $\mathcal{D}$ and any $\bar{c}$:

$$(\mathcal{T},\mathcal{A})\models q(\bar{c}) \quad\Leftrightarrow\quad f(\mathrm{can}(\mathcal{T},\mathcal{A}))\models q(\bar{c}) \text{ for all completion functions } f.$$

We call $\mathrm{can}(\mathcal{T},\mathcal{A})$ with this property a *universal pre-model* of $\mathcal{T}$ and $\mathcal{A}$.

Lemma 1. *For every satisfable $\mathcal{D}$-KB $(\mathcal{T},\mathcal{A})$ there exists a universal pre-model $can(\mathcal{T},\mathcal{A})$.*

Example 2. A universal pre-model $\mathrm{can}(\mathcal{T},\mathcal{A})$ for the KB $(\mathcal{T},\mathcal{A})$ given in Example 1 is given by setting $\Delta^{\mathrm{can}(\mathcal{T},\mathcal{A})}=\{a,u_1,u_2\}$; $A^{\mathrm{can}(\mathcal{T},\mathcal{A})}=\{a\}$; $U_1^{\mathrm{can}(\mathcal{T},\mathcal{A})}=\{(a,u_1)\}$; and $U_2^{\mathrm{can}(\mathcal{T},\mathcal{A})}=\{(a,u_2)\}$.

A completion function f for $\mathrm{can}(\mathcal{T},\mathcal{A})$ maps u_1 and u_2 to rational numbers and defines a completion $f(\mathrm{can}(\mathcal{T},\mathcal{A}))$ in which U_1 is interpreted as $(a,f(u_1))$ and U_2 is interpreted as $(a,f(u_2))$.

The universal pre-model $\mathrm{can}(\mathcal{T},\mathcal{A})$ can be infinite. If we are given a rooted OMQ $Q = (\mathcal{T}, q, \mathcal{D})$, then it is sufficient to consider the subinterpretation $\mathrm{can}^n(\mathcal{T},\mathcal{A})$ of $\mathrm{can}(\mathcal{T},\mathcal{A})$ induced by the set of domain elements that are reachable from ABox elements in at most $n = |q|$ steps. We call $\mathrm{can}^n(\mathcal{T},\mathcal{A})$ a *n-universal pre-model* of $\mathcal{T}$ and $\mathcal{A}$. As Q is rooted, it has the following property for any $\bar{c}$:

$$(\mathcal{T},\mathcal{A}) \models q(\bar{c}) \quad \Leftrightarrow \quad f(\mathrm{can}^n(\mathcal{T},\mathcal{A})) \models q(\bar{c}) \text{ for all completion functions } f.$$

A straightforward modification of the standard chase shows that an n-universal pre-model $\mathrm{can}^n(\mathcal{T},\mathcal{A})$ can be computed in polynomial time in the size of $\mathcal{A}$ [9].

Lemma 2. *Assume OMQ $Q = (\mathcal{T}, q, \mathcal{D})$ is given. Then one can compute for any ABox $\mathcal{A}$ that is satisfiable relative to $\mathcal{T}$ a $|q|$-universal pre-model of $\mathcal{T}$ and $\mathcal{A}$ in polynomial time in the size of $\mathcal{A}$.*

A *positive existential sentence Φ* over a datatype $\mathcal{D}$ is a first-order sentence built from atomic formulas over the relations in $\mathcal{D}$ by using solely conjunction, disjunction and existential quantifiers. Atomic formulas can use both individual variables and constants from D. Φ is in Conjunctive Normal Form (CNF) if it has the form

$$\Phi = \exists \bar{x} \bigwedge_{i=1}^{m} c_i, \quad \text{where } c_i = \bigvee_{j=1}^{n_i} c_{i,j} \text{ for } i = 1, \ldots m$$

where $c_{i,j}$ are atomic formulas. If $n_i = k$, for each i, then we say that Φ is in k-CNF. The problem of deciding whether a positive existential sentence in k-CNF is satisfied in $\mathcal{D}$ is denoted $\mathsf{PE}_k(\mathcal{D})$. We now show that we have a polynomial time reduction from evaluating OMQs over $\mathcal{D}$ to the complement of the problem $\mathsf{PE}_k(\bar{\mathcal{D}})$ and vice versa.

Theorem 1. *Let $k > 0$ and let $\mathcal{D} = (D, R_1, \ldots, R_n)$ be a datatype.*

Let $Q = (\mathcal{T}, q, \mathcal{D})$ be a rooted OMQ and assume that $\mathsf{dtype}(q)$ has k atoms. Then evaluating Q is polynomially reducible to the complement of $\mathsf{PE}_k(\bar{\mathcal{D}})$.

Conversely, there is a rooted OMQ $Q = (\mathcal{T}, q, \mathcal{D})$ such that $\mathsf{dtype}(q)$ has nk atoms and $\mathsf{PE}_k(\bar{\mathcal{D}})$ is polynomially reducible to the complement of evaluating Q.

Proof. Assume q is given as $q(\bar{x}) \leftarrow \varphi$. Let $\mathcal{A}$ be an ABox satisfiable relative to $\mathcal{T}$ and let $\bar{c}$ be a tuple of individual names and data values in $\mathcal{A}$ of the same length as $\bar{x}$. Remove from φ the datatype pattern of q and denote by ψ the remaining atoms in φ. A *match* of ψ in a pre-interpretation $\mathcal{I}$ is a mapping μ from the variables in ψ to $\Delta^{\mathcal{I}}$ such that for each atom $X(t_1, \ldots, t_k)$ in ψ we have $(\mu(t_1), \ldots, \mu(t_k)) \in X^{\mathcal{I}}$. Consider the set X of all matches μ of ψ with $\mu(\bar{x}) = \bar{c}$ in $\mathrm{can}^n(\mathcal{T},\mathcal{A})$, where $n = |q|$. Now assume that $\mathsf{dtype}(q) = \bigwedge_{i=1}^{k} S_i(\bar{z}_i)$ and let

$$\Phi := \exists \bar{u} \bigwedge_{\mu \in X} \bigvee_{i=1}^{k} \bar{S}_i(\mu(\bar{z}_i)),$$

where $\bar{u}$ is a repetition-free enumeration of all data nulls in the set $\{\mu(\bar{z}_i) \mid \mu \in X, 1 \leq i \leq k\}$ (here we identify data nulls with individual variables in the k-CNF

Φ). It is readily checked that $\mathcal{T}, \mathcal{A} \models q(\bar{c})$ if, and only if, $\bar{\mathcal{D}} \not\models \Phi$. This establishes the first part.

Conversely, assume $\Phi = \exists \bar{x} \bigwedge_{i=1}^{m} c_i$, where $c_i = \bigvee_{j=1}^{k} c_{i,j}$ for $1 \leq i \leq m$. We first assume that Φ is *uniform*, that is, for each $1 \leq j \leq k$ there exists a relation S_j (independent from i) such that $c_{i,j}$ is of the form $\bar{S}_j(\bar{t}_{i,j})$. In this case we can construct the required OMQ $(\mathcal{T}, q, \mathcal{D})$ with $|\mathsf{dtype}(q)| = k$. The TBox $\mathcal{T}$ is independent from k and defined as $\mathcal{T} = \{A \sqsubseteq \exists U\}$. Assume that the relations S_j are of arity l_j and $c_{i,j} = \bar{S}_j(\bar{t}_{i,j})$ for all $1 \leq i \leq m$. Before defining the CQ q we define an ABox $\mathcal{A}_\Phi$ as follows:

- $\mathcal{A}_\Phi$ uses individuals c_Φ and $c_1, \ldots, c_m$ that are connected by a role name P using the assertions $P(c_\Phi, c_i)$ for $1 \leq i \leq m$;
- in addition $\mathcal{A}_\Phi$ uses individuals $c_{i,j}$ which are connected to the individuals c_i by role names $P_1, \ldots, P_k$ using the assertions $P_j(c_i, c_{i,j})$ for $1 \leq i \leq m$ and $1 \leq j \leq k$;
- in addition $\mathcal{A}_\Phi$ uses individuals d_t for each variable and constant t in Φ that are connected to the $c_{i,j}$ using role names $N_1^j, \ldots, N_{l_j}^j$ for $1 \leq j \leq k$ and the assertions $N_r^j(c_{i,j}, d_t)$ if the r-th component of $\bar{t}_{i,j}$ equals t;
- finally $\mathcal{A}_\Phi$ contains $A(d_t)$ if t is a variable in Φ and $U(d_t, t)$ if t is a constant in Φ.

Define the query $q(x) \leftarrow \psi$, by setting

$$\psi = P(x,y) \wedge \bigwedge_{j=1}^{k} \Big(P_j(y, y_j) \wedge \bigwedge_{r=1}^{l_j} \big(N_r^j(y_j, z_{j,r}) \wedge U(z_{j,r}, u_{j,r}) \big) \wedge S_j(u_{j,1}, \ldots, u_{j,l_j}) \Big)$$

It is not difficult to show that $\bar{\mathcal{D}} \models \Phi$ if, and only if, $\mathcal{T}, \mathcal{A}_\Phi \not\models q(c_\Phi)$. For

$$\Phi_0 = \exists x_1 \exists x_2 \, \big((R_1(1, x_1) \vee R_2(x_2, x_1)) \wedge (R_1(x_1, x_2) \vee R_2(x_2, 2)) \big)$$

the ABox $\mathcal{A}_\Phi$ and query q are shown in Figure 1.

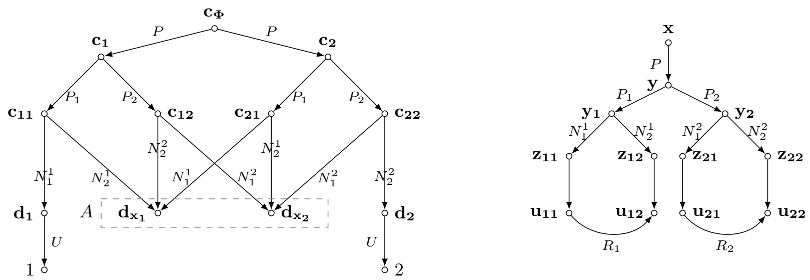

Fig. 1. ABox $\mathcal{A}_{\Phi_0}$ and CQ q

It remains to consider the case in which Φ is not uniform. We may assume that $R_i \neq \emptyset$ for all $1 \leq i \leq n$. We equivalently transform Φ into a uniform sentence Ψ

in nk-CNF over $\bar{\mathcal{D}}$. To this end, each conjunct c_i of Φ is equivalently transformed into a conjunct c_i' of the form

$$\bigvee_{j=1}^{k} \bar{R}_1(\bar{t}^1_{i,j}) \vee \cdots \vee \bigvee_{j=1}^{k} \bar{R}_n(\bar{t}^n_{i,j})$$

We construct $\bar{R}_1(\bar{t}^1_{i,j})$ for a fixed i and $1 \leq j \leq k$. The remaining atoms are constructed in the same way. c_i contains between 0 and k disjuncts of the form $\bar{R}_1(\bar{t})$. Thus, if c_i contains at least one disjunct of this form we take sufficiently many copies to obtain $\bar{R}_1(\bar{t}^1_{i,1}) \vee \ldots, \vee R_1(\bar{t}^1_{i,k})$ that is equivalent to the disjunction over all atoms of the form $\bar{R}_1(\bar{t})$ in c_i. If c_i does not contain any $\bar{R}_1(\bar{t})$, then take $\bar{c}$ with $\bar{c} \in R_1$ and let $\bar{t}^1_{i,1} = \cdots = \bar{t}^1_{i,k} = \bar{c}$. Then $\bar{R}_1(\bar{t}^1_{i,1}) \vee \ldots \vee \bar{R}_1(\bar{t}^1_{i,k})$ is unsatisfiable, as required. □

We illustrate Theorem 1 by transfering some complexity results from temporal CSP over $(\mathbb{Q}, <)$ to OMQs. Recall that we are after a complexity classification. The following result shows that answering OMQs with datatype $(\mathbb{Q}, \leq)$ can be intractable already for datatype patterns of size two.

Corollary 1. *There is a rooted OMQ* $Q = (\mathcal{T}, q, (\mathbb{Q}, \leq))$ *with* $\mathcal{T} = \{A \sqsubseteq \exists U\}$ *and* $|\mathsf{dtype}(q)| = 2$ *such that evaluating* Q *is coNP-hard.*

Proof. The BETWEENNESS problem in temporal constraints satisfaction is the problem to decide whether $\beta := \exists \bar{u} \bigwedge_{(x,y,z) \in C} (x < y < z \vee z < y < x)$ is satisfiable in $(\mathbb{Q}, <)$, where C is a set of triples of the form (x, y, z). This problem is NP-complete [18]. Clearly, β is equivalent to the 2-CNF

$$\exists \bar{u} \bigwedge_{(x,y,z) \in C} (x < y \vee z < y) \wedge (x < y \vee y < x) \wedge (y < z \vee z < y) \wedge (y < z \vee y < x).$$

The claim now follows from Theorem 1. □

This is optimal as shown by the following result.

Corollary 2. *Let* $\mathcal{D} \in \{(\mathbb{Z}, <), (\mathbb{Z}, \leq), (\mathbb{Q}, <), (\mathbb{Q}, \leq)\}$. *Then evaluating rooted OMQs* $Q = (\mathcal{T}, q, \mathcal{D})$ *with* $|\mathsf{dtype}(q)| = 1$ *is in PTime.*

Proof. Follows from Theorem 1 and the observation that satisfiability of sentences in 1-CNF in $\mathcal{D}$ is decidable in PTime. □

4 A Dichotomy for $(\mathbb{Q}, \leq)$

In this section, we focus on rooted OMQs that use the datatype $(\mathbb{Q}, \leq)$. We prove a P/coNP-dichotomy of evaluating such OMQs based on their datatype pattern, and provide a simple syntactic characterization of the datatype patterns of rooted OMQs with datatype $(\mathbb{Q}, \leq)$ for which the evaluation problem can be solved in polynomial time (Theorem 4). These results are based on a recent dichtomy result by Bodirsky and Kára [7] for temporal constraint satisfaction problems.

We start by reviewing the temporal constraint satisfaction framework of [7]. A *temporal constraint language* is a logical structure $\Gamma = (\mathbb{Q}, R_1, R_2, \dots)$, where each R_i of arity k is definable by a first-order formula $\Phi(x_1, \dots, x_k)$ over $(\mathbb{Q}, <)$, i.e., $R_i = \{(a_1, \dots, a_k) \in \mathbb{Q}^k \mid (\mathbb{Q}, <) \models \Phi(a_1, \dots, a_k)\}$. A *primitive positive sentence* over Γ is a first-order sentence over Γ built from atomic formulas using conjunction and existential quantification. It is crucial for the results in [7] that both first-order definitions of relations in Γ and primitive positive sentences over Γ do not contain constants. The problem of deciding whether a primitive positive sentence over Γ is satisfied in Γ is denoted by $\mathsf{CSP}(\Gamma)$.

Bodirsky and Kára [7] proved that for every temporal constraint language Γ, $\mathsf{CSP}(\Gamma)$ is either in PTime or NP-complete. They characterize the temporal languages Γ for which $\mathsf{CSP}(\Gamma)$ is tractable in terms of preservation properties of the relations in Γ. A function $f \colon \mathbb{Q}^k \to \mathbb{Q}$ *preserves* a relation $R \subseteq \mathbb{Q}^n$ if for all $t_1, \dots, t_k \in R$ we have $f(t_1, \dots, t_k) \in R$. Here, $f(t_1, \dots, t_k)$ is obtained as follows. Given a tuple t of length n and an integer $i \in \{1, \dots, n\}$, let $t[i]$ denote the ith component of t. Then, $f(t_1, \dots, t_k) = \big(f(t_1[1], \dots, t_k[1]), \dots, f(t_1[n], \dots, t_k[n])\big)$. We say that f preserves a temporal constraint language Γ if f preserves all relations in Γ. The following functions are considered in [7]:

- $\mathit{min} \colon \mathbb{Q}^2 \to \mathbb{Q}$ which maps its two arguments to the minimal one;
- $\mathit{mi} \colon \mathbb{Q}^2 \to \mathbb{Q}$ which maps $(x, y) \in \mathbb{Q}^2$ to $\alpha(x)$ if $x = y$, to $\beta(y)$ if $x > y$, and to $\gamma(x)$ if $x < y$, where α, β, γ are any functions with $\alpha(x) < \beta(x) < \gamma(x) < \alpha(y)$ for all $x < y$;
- $\mathit{mx} \colon \mathbb{Q}^2 \to \mathbb{Q}$ which maps $(x, y) \in \mathbb{Q}^2$ to $\beta(x)$ if $x = y$, and to $\alpha(\min\{x, y\})$ if $x \neq y$, where α, β are any functions with $\alpha(x) < \beta(x) < \alpha(y)$ for all $x < y$;
- $\mathit{ll} \colon \mathbb{Q}^2 \to \mathbb{Q}$ which is any function that satisfies $\mathit{ll}(x, y) < \mathit{ll}(x', y')$ iff $x \leq 0$ and (x, y) is lexicographically smaller than (x', y'), or $x, x' > 0$ and (y, x) is lexicographically smaller than (y', x');
- The dual of $f \in \{\mathit{min}, \mathit{mi}, \mathit{mx}, \mathit{ll}\}$, which maps $(x, y) \in \mathbb{Q}^2$ to $-f(-x, -y)$.

Theorem 2 ([7]). *Let Γ be a temporal constraint language. If Γ is preserved under min, mi, mx, ll, one of their duals, or a constant function, then* $\mathsf{CSP}(\Gamma)$ *is in PTime. Otherwise,* $\mathsf{CSP}(\Gamma)$ *is NP-complete.*

We now translate the evaluation problem for rooted OMQs over $(\mathbb{Q}, \leq)$ into the temporal constraint satisfaction framework. With every datatype pattern $q_0(z_1, \dots, z_n) = \bigwedge_{i=1}^{k} z_{s_i} \leq z_{t_i}$ we associate the temporal constraint language

$$\Gamma_{q_0} := (\mathbb{Q}, <, R_{q_0}) \text{ where } R_{q_0} := \{(c_1, \dots, c_n) \in \mathbb{Q}^n \mid (\mathbb{Q}, <) \models \textstyle\bigvee_{i=1}^{k} c_{t_i} < c_{s_i}\}.$$

Theorem 3. *Let $q_0(z_1, \dots, z_n) = \bigwedge_{i=1}^{k} z_{s_i} \leq z_{t_i}$ be a datatype pattern.*

If $Q = (\mathcal{T}, q, (\mathbb{Q}, \leq))$ is a rooted OMQ with $\mathsf{dtype}(q) = q_0$*, then evaluating Q is polynomially reducible to the complement of* $\mathsf{CSP}(\Gamma_{q_0})$.

Conversely, there is a rooted OMQ $Q = (\mathcal{T}, q, (\mathbb{Q}, \leq))$ with $\mathsf{dtype}(q) = q_0$ *such that* $\mathsf{CSP}(\Gamma_{q_0})$ *is polynomially reducible to the complement of evaluating Q.*

Proof (Sketch). We refine the proof of Theorem 1. For the first part, we construct the positive existential sentence as in the proof of Theorem 1, and then express each disjunctive clause by a single R_{q_0}-atom. The result is a primitive positive sentence Ψ' over Γ_{q_0}. Note that Γ_{q_0} may still contain constants. To obtain a primitive positive sentence Ψ without constants, we turn each constant in Ψ' into an existentially quantified variable and add constraints that ensure that the order of the elements assigned to these variables corresponds to the order of the constants. More precisely, let $c_1 < \cdots < c_l$ be the constants in Ψ'. Then, $\Psi = \exists c_1 \cdots \exists c_l \left(\Psi' \wedge \bigwedge_{i=1}^{l-1} c_i < c_{i+1}\right)$. It can be shown that $\Gamma_{q_0} \models \Psi$ iff $\Gamma_{q_0} \models \Psi'$.

For the second part, we first translate a given primitive positive sentence over Γ_{q_0} into a positive existential sentence over $(\mathbb{Q}, >)$. Atoms using the relation R_{q_0} are replaced by disjunctive formulas expressing the predicate defining the relation R_{q_0}. Atoms using $<$ are expanded into disjunctions with k atoms. The resulting sentence is in k-CNF. The second part of the theorem now follows from the construction in the proof of Theorem 1. Details can be found in Appendix A.2. □

Since $<$ is preserved under *min*, *mi*, *mx*, *ll* and their duals, but not under any constant function (see Proposition 1 in Appendix A.1), Theorem 2 implies that $\mathsf{CSP}(\Gamma_{q_0})$ is in PTime iff R_{q_0} is preserved under *min*, *mi*, *mx*, *ll* or one of their duals. Together with Theorem 3, we obtain the following corollary.

Corollary 3. *Let q_0 be a datatype pattern. If R_{q_0} is preserved under min, mi, mx, ll, or one of their duals, then evaluating rooted OMQs $(\mathcal{T}, q, (\mathbb{Q}, \leq))$ with $\mathsf{dtype}(q) = q_0$ is in PTime. Otherwise, there is a rooted OMQ $Q = (\mathcal{T}, q, (\mathbb{Q}, \leq))$ with $\mathsf{dtype}(q) = q_0$ such that evaluating Q is coNP-complete.*

To illustrate, consider the datatype pattern $q_0(x, y, z) = x \leq y \wedge y \leq z$. It is straightforward to verify that $R_{q_0} = \{(a, b, c) \in \mathbb{Q}^3 \mid a > b \vee b > c\}$ is not preserved under *min*, *mi*, *mx*, *ll* or their duals, so there are OMQs $(\mathcal{T}, q, (\mathbb{Q}, \leq))$ with $\mathsf{dtype}(q) = q_0$ for which the evaluation problem is coNP-complete. On the other hand, if q_0 has the form $\bigwedge_{i=1}^{n} x_0 \leq x_i$ or $\bigwedge_{i=1}^{n} x_i \leq x_0$, then it follows from [8, Proposition 3.5] that R_{q_0} is preserved under *ll* or its dual. In particular, evaluation of OMQs $(\mathcal{T}, q, (\mathbb{Q}, \leq))$ with $\mathsf{dtype}(q) = q_0$ is in PTime. In fact, we will now show that these two forms of datatype patterns, which we call *min-pattern* and *max-pattern*, respectively, essentially characterize all the tractable cases.

The following lemma is the core of the characterization result. It implies that preservation under *min*, *mi*, *mx* or their duals collapses to preservation under *ll* or its dual for relations that are definable by normal disjunctive formulas. By a *disjunctive* formula we mean a disjunction of atoms of the form $x < y$. A disjunctive formula $\Phi(x_1, \ldots, x_n)$ is *normal* if the directed graph with vertex set $\{x_1, \ldots, x_n\}$ and edge set $\{(x_j, x_i) \mid x_i < x_j \in \Phi\}$ is acyclic.

Lemma 3. *Let $R \subseteq \mathbb{Q}^n$ be defined by a normal disjunctive formula $\Phi(x_1, \ldots, x_n)$ over $(\mathbb{Q}, <)$. If R is preserved under min, mi, mx, ll, or one of their duals, then Φ has the form $\bigvee_{i=1}^{n} x_0 > x_i$ or $\bigvee_{i=1}^{n} x_i > x_0$.*

In particular, [8, Proposition 3.5] implies that a relation defined by a formula of the form $\bigvee_{i=1}^{n} x_0 > x_i$ or $\bigwedge_{i=1}^{n} x_i > x_0$ is preserved under *ll* or its dual. The proof of Lemma 3 can be found in Appendix A.3.

We apply the lemma to relations R_{q_0} for acyclic datatype patterns q_0. A datatype pattern q_0 is *acyclic* if the directed graph with the variables of q_0 as vertices and an edge (x, y) for each atom $x \leq y$ of q_0 is acyclic. Since a cycle $x_0 \leq x_1 \wedge x_1 \leq x_2 \wedge \cdots \wedge x_n \leq x_0$ tells us that $x_0, x_1, \ldots, x_n$ have to be mapped to the same data value, we can eliminate any cycle by removing all of its atoms, and replacing $x_1, \ldots, x_n$ by x_0. Let q_0^{acyclic} be the acyclic datatype pattern obtained from a datatype pattern q_0 by eliminating all of its cycles.

Theorem 4. *Let q_0 be a datatype pattern over $(\mathbb{Q}, \leq)$. If q_0^{acyclic} is a min-pattern or a max-pattern, then evaluating rooted OMQs $(\mathcal{T}, q, (\mathbb{Q}, \leq))$ with $\mathsf{dtype}(q) = q_0$ is in PTime. Otherwise, there is a rooted OMQ $Q = (\mathcal{T}, q, (\mathbb{Q}, \leq))$ with $\mathsf{dtype}(q) = q_0$ such that evaluating Q is coNP-complete.*

Proof. To simplify the presentation, we assume without loss of generality that q_0 is acyclic. By Corollary 3, it suffices to show that q_0 is a min-pattern or a max-pattern iff R_{q_0} is preserved under *min*, *mi*, *mx*, *ll*, or one of their duals.

If q_0 is a min-pattern, then R_{q_0} is defined by a formula of the form $\bigvee_{i=1}^{n} x_0 > x_i$. Similarly, if q_0 is a max-pattern, then R_{q_0} is defined by a formula of the form $\bigvee_{i=1}^{n} x_i > x_0$. It is known [8, Proposition 3.5] that relations defined by such formulas are preserved under *ll* and *dual-ll*, respectively.

Now suppose that R_{q_0} is preserved under *min*, *mi*, *mx*, *ll*, or one of their duals. Let $q_0(z_1, \ldots, z_n) = \bigwedge_{i=1}^{k} z_{s_i} \leq z_{t_i}$. Then, R_{q_0} is defined by $\Phi(z_1, \ldots, z_n) = \bigvee_{i=1}^{k} z_{t_i} < z_{s_i}$. Clearly, Φ is disjunctive, and since q_0 is acyclic it is also normal. Thus, Lemma 3 implies that Φ has the form $\bigvee_{i=1}^{n} x_0 > x_i$ or $\bigvee_{i=1}^{n} x_i > x_0$. This implies that q_0 is a min-pattern or a max-pattern. ☐

Note that the fact that q_0^{acyclic} is neither a min-pattern nor a max-pattern does not imply that evaluation is hard for *all* OMQs $Q = (\mathcal{T}, q, (\mathbb{Q}, \leq))$ with $\mathsf{dtype}(q) = q_0$. For example, q_0^{acyclic} may have several connected components, each a min-pattern or a max-pattern. If no component is connected to another one via a path of existential variables in q one can show that evaluating Q is in PTime.

5 Conclusion

We have presented first promising results for a non-uniform complexity analysis of ontology-mediated queries with expressive datatypes. Many research questions arise within this framework, including the following: (1) It remains an interesting open problem whether our results generalize to non-rooted OMQs. Such OMQs can "look" arbitrarily deep into a universal pre-model. We suspect that it is enough to inspect only a finite portion of it, but this is far from obvious. (2) The TBoxes we consider have very limited expressive power regarding datatypes and it would be of interest to generalize our method to TBoxes that admit more constructors using datatypes. (3) In this paper, we have used datatype patterns within CQs to classify the complexity of OMQs. It would be of interest to complement and extend this approach with classifications based on TBoxes, OMQs, or extended patterns in CQs that take into account additional atoms not using datatype relations. (4) In addition to the PTime/coNP dichotomy considered above, it would be of interest to investigate FO-rewritability and Datalog-rewritability of OMQs as well [5].

References

1. Artale, A., Calvanese, D., Kontchakov, R., Zakharyaschev, M.: The *DL-Lite* family and relations. J. Artif. Intell. Res. (JAIR) 36, 1–69 (2009)
2. Artale, A., Ryzhikov, V., Kontchakov, R.: *DL-Lite* with attributes and datatypes. In: ECAI. pp. 61–66 (2012)
3. Baader, F., Brandt, S., Lutz, C.: Pushing the $\mathcal{EL}$ envelope. In: IJCAI-05. pp. 364–369 (2005)
4. Baader, F., Hanschke, P.: A scheme for integrating concrete domains into concept languages. In: IJCAI 1991. pp. 452–457
5. Bienvenu, M., ten Cate, B., Lutz, C., Wolter, F.: Ontology-based data access: A study through disjunctive datalog, csp, and MMSNP. ACM Trans. Database Syst. 39(4), 33:1–33:44 (2014)
6. Bienvenu, M., Ortiz, M.: Ontology-mediated query answering with data-tractable description logics. In: Reasoning Web 2015. pp. 218–307
7. Bodirsky, M., Kára, J.: The complexity of temporal constraint satisfaction problems. J. ACM 57(2) (2010)
8. Bodirsky, M., Kára, J.: A fast algorithm and datalog inexpressibility for temporal reasoning. ACM Trans. Comput. Log. 11(3) (2010), `http://doi.acm.org/10.1145/1740582.1740583`
9. Botoeva, E., Kontchakov, R., Ryzhikov, V., Wolter, F., Zakharyaschev, M.: Query inseparability for description logic knowledge bases. In: KR 2014
10. Bulatov, A.A., Jeavons, P., Krokhin, A.A.: Classifying the complexity of constraints using finite algebras. SIAM J. Comput. 34(3), 720–742 (2005)
11. Calvanese, D., De Giacomo, G., Lembo, D., Lenzerini, M., Rosati, R.: Tractable reasoning and efficient query answering in description logics: The *DL-Lite* family. J. Autom. Reasoning 39(3), 385–429 (2007)
12. Feder, T., Vardi, M.Y.: Monotone monadic SNP and constraint satisfaction. In: Proc. of the ACM Symposium on Theory of Computing. pp. 612–622 (1993)
13. Hernich, A., Lutz, C., Ozaki, A., Wolter, F.: Schema.org as a description logic. In: IJCAI 2015. pp. 3048–3054
14. Hustadt, U., Motik, B., Sattler, U.: Data complexity of reasoning in very expressive description logics. In: IJCAI 2005. pp. 466–471
15. Lutz, C.: Description logics with concrete domains-a survey. In: Advances in Modal Logic 4. pp. 265–296 (2002)
16. Lutz, C., Wolter, F.: Non-uniform data complexity of query answering in description logics. In: KR 2012
17. Magka, D., Kazakov, Y., Horrocks, I.: Tractable extensions of the description logic $\mathcal{EL}$ with numerical datatypes. J. Autom. Reasoning 47(4), 427–450 (2011)
18. Opatrny, J.: Total ordering problem. SIAM J. Comput. 8(1), 111–114 (1979)
19. Poggi, A., Lembo, D., Calvanese, D., De Giacomo, G., Lenzerini, M., Rosati, R.: Linking data to ontologies. J. Data Semantics 10, 133–173 (2008)
20. Savkovic, O., Calvanese, D.: Introducing datatypes in *DL-Lite*. In: ECAI 2012. pp. 720–725

A Proofs omitted in Section 4

A.1 Preservation Properties of $<$

Proposition 1.

1. *$<$ is preserved under min, mi, mx, ll and their duals.*
2. *$<$ is not preserved under any constant function.*

Proof. We only consider preservation under *min*, *mi*, *mx*, *ll* and constant functions. The proofs for the duals of *min*, *mi*, *mx*, *ll* are similar.

Preservation under *min*: Let $a_1 < a_2$ and $b_1 < b_2$. We have to show that $c_1 < c_2$, where $c_i := min(a_i, b_i)$. If $c_1 = a_1$, then $c_1 = a_1 < a_2$ and $c_1 = a_1 \leq b_1 < b_2$, thus $c_1 < c_2$. Similarly, if $c_1 = b_1$, then $c_1 = b_1 \leq a_1 < a_2$ and $c_1 = b_1 < b_2$, thus $c_1 < c_2$. This shows that $<$ is preserved under *min*.

Preservation under *mi*: Let $a_1 < a_2$ and $b_1 < b_2$. We have to show that $c_1 < c_2$, where $c_i := mi(a_i, b_i)$. Since $<$ is preserved by *min*, we have $\min(a_1, b_1) < \min(a_2, b_2)$. This implies $c_1 = mi(a_1, b_1) < mi(a_2, b_2) = c_2$. Altogether, we have shown that $<$ is preserved under *mi*.

Preservation under *mx*: Let $a_1 < a_2$ and $b_1 < b_2$. We have to show that $c_1 < c_2$, where $c_i := mx(a_i, b_i)$. Since $<$ is preserved by *min*, we have $\min(a_1, b_1) < \min(a_2, b_2)$. This implies $c_1 = mx(a_1, b_1) < mx(a_2, b_2) = c_2$. Altogether, we have shown that $<$ is preserved under *mx*.

Preservation under *ll*: Let $a_1 < a_2$ and $b_1 < b_2$. We have to show that $ll(a_1, b_1) < ll(a_2, b_2)$. If $a_1 \leq 0$, then $a_1 < a_2$ immediately yields $ll(a_1, b_1) < ll(a_2, b_2)$. Now suppose that $a_1 > 0$. Since $a_1 < a_2$, we also have $a_2 > 0$. But then, $b_1 < b_2$ immediately yields $ll(a_1, b_1) < ll(a_2, b_2)$.

Non-preservation under constant functions: For a contradiction, suppose that $<$ is preserved under a constant function $f\colon \mathbb{Q}^k \to \{c\}$. Take any $a_1 < b_1, \ldots, a_k < b_k$. Since f preserves $<$, we have $c = f(a_1, \ldots, a_k) < f(b_1, \ldots, b_k) = c$, which is clearly false. ☐

A.2 Proof of Theorem 3

Theorem 3. Let $q_0(z_1, \ldots, z_n) = \bigwedge_{i=1}^k z_{s_i} \leq z_{t_i}$ be a datatype pattern.

If $Q = (\mathcal{T}, q, (\mathbb{Q}, \leq))$ is a rooted OMQ with $\mathsf{dtype}(q) = q_0$, then evaluating Q is polynomially reducible to the complement of $\mathsf{CSP}(\Gamma_{q_0})$.

Conversely, there is a rooted OMQ $Q = (\mathcal{T}, q, (\mathbb{Q}, \leq))$ with $\mathsf{dtype}(q) = q_0$ such that $\mathsf{CSP}(\Gamma_{q_0})$ is polynomially reducible to the complement of evaluating Q.

Proof. Let $\mathcal{A}$ be an ABox satisfiable relative to $\mathcal{T}$, let $\bar{x}$ be the tuple of answer variables of q, and let $\bar{c}$ be a tuple of individual names and data values in $\mathcal{A}$ of the same length as $\bar{x}$. As shown in the proof of Theorem 1, we have $\mathcal{T}, \mathcal{A} \models q(\bar{c})$ if, and only if, $(\mathbb{Q}, >) \not\models \Phi$, where $\Phi = \exists \bar{u} \bigwedge_{\mu \in X} \bigvee_{i=1}^k \mu(z_{s_i}) > \mu(z_{t_i})$, and X and $\bar{u}$ are defined as in the proof of Theorem 1. We have constructed Γ_{q_0} in such a way that $(\mathbb{Q}, >) \models \Phi$ is equivalent to $\Gamma_{q_0} \models \Psi'$, where $\Psi' := \exists \bar{u} \bigwedge_{\mu \in X} R_{q_0}(\mu(\bar{z}))$. Note that Ψ' may contain constants. To eliminate these constants, let $c_1, \dots, c_l$ be the list of all constants that occur in Ψ', sorted in ascending order. We view these constants as variables, and define

$$\Psi := \exists c_1 \cdots \exists c_l \exists \bar{u} \left(\bigwedge_{\mu \in X} R_{q_0}(\mu(\bar{z})) \wedge \bigwedge_{i=1}^{l-1} c_i < c_{i+1} \right).$$

Then, $\Gamma_{q_0} \models \Psi'$ if, and only if, $\Gamma_{q_0} \models \Psi$. The "only if" direction is trivial. To see the "if" direction, suppose that $\Gamma_{q_0} \models \Psi$. Let f be an assignment of data values to the existential variables in Ψ that satisfies the quantifier-free part of Ψ in $(\mathbb{Q}, <)$. Pick any automorphism α of $(\mathbb{Q}, <)$ such that $\alpha(f(c_i)) = c_i$ for all $i \in \{1, \dots, l\}$. Then, $\alpha \circ f$ satisfies the quantifier-free part of Ψ' in $(\mathbb{Q}, <)$, and consequently $\Gamma_{q_0} \models \Psi'$. Altogether, this shows that $\mathcal{T}, \mathcal{A} \models q(\bar{c})$ if, and only if, $\Gamma_{q_0} \not\models \Psi$, and establishes the first part.

For the second part, let $\Psi = \exists \bar{x} \bigwedge_{j=1}^{m} c_j$ be a primitive positive sentence over Γ_{q_0}. We translate Ψ into a positive existential sentence $\Psi' = \exists \bar{x} \bigwedge_{j=1}^{m} c'_j$ over $(\mathbb{Q}, >)$, where the c'_j are defined as follows. If c_j has the form $y_1 < y_2$, then $c'_j := \bigvee_{i=1}^{k} y_2 > y_1$. If c_j has the form $R_{q_0}(y_1, \dots, y_n)$, then $c'_i := \bigvee_{i=1}^{k} y_{s_i} > y_{t_i}$. By the definition of R_{q_0}, we have $\Gamma_{q_0} \models \Psi$ iff $(\mathbb{Q}, >) \models \Psi'$. The second part of the lemma now follows from the construction in the proof of Theorem 1. □

A.3 Proof of Lemma 3

Before we prove Lemma 3, we establish two auxilliary lemmas. Lemma 3 then follows as a corollary of the second lemma.

Recall that $t[i]$ denotes the ith component of a tuple t.

Lemma 4. *Let $f \colon \mathbb{Q}^2 \to \mathbb{Q}$ and let $a_1, \dots, a_4, b_1, \dots, b_4 \in \mathbb{Q}$ be such that*

$$f(a_1, b_1) \geq \cdots \geq f(a_4, b_4).$$

Let $1 \leq i_1 \leq i_2 \leq i_3 \leq i_4 \leq n$, and suppose that $(a_j, b_j) = (a_{j'}, b_{j'})$ if $i_j = i_{j'}$. Then, there are tuples $t_1, t_2 \in \mathbb{Q}^n$ such that $t_1[i_j] = a_j$ and $t_2[i_j] = b_j$ for all $j \in \{1, 2, 3, 4\}$, and

$$f(t_1[1], t_2[1]) \geq \cdots \geq f(t_1[n], t_2[n]).$$

Proof. Define $t_1, t_2 \in \mathbb{Q}^n$ such that for all $p \in \{1, \dots, n\}$, we have that

$$t_1[p] := \begin{cases} a_1, & \text{if } p \leq i_1 \\ a_2, & \text{if } i_1 < p \leq i_2 \\ a_3, & \text{if } i_2 < p \leq i_3 \\ a_4, & \text{if } i_3 < p. \end{cases} \qquad t_2[p] := \begin{cases} b_1, & \text{if } p \leq i_1 \\ b_2, & \text{if } i_1 < p \leq i_2 \\ b_3, & \text{if } i_2 < p \leq i_3 \\ b_4, & \text{if } i_3 < p. \end{cases}$$

Clearly, $t_1[i_j] = a_j$ and $t_2[i_j] = b_j$ for all $j \in \{1, 2, 3, 4\}$. From the construction of t_1 and t_2 and the properties of $a_1, \dots, a_4, b_1, \dots, b_4$, it immediately follows that $f(t_1[1], t_2[1]) \geq \cdots \geq f(t_1[n], t_2[n])$. □

Lemma 5. *Let $R \subseteq \mathbb{Q}^n$ be defined by a normal disjunctive formula $\Phi(x_1, \ldots, x_n)$ over $(\mathbb{Q}, <)$. If R is preserved under min, mi, mx, or one of their duals, then for every two disjuncts $x_i < x_j$ and $x_{i'} < x_{j'}$ of Φ, either $i = i'$ or $j = j'$.*

Proof. We will only consider the case that R is preserved under min, mi, or mx. The duals of min, mi, or mx can be dealt with analogously (just replace the numbers in the constructions below by their negative).

So, let R be preserved under $f \in \{min, mi, mx\}$, and let $x_i < x_j$ and $x_{i'} < x_{j'}$ be disjuncts of Φ. For the sake of contradiction, assume $i \neq i'$ and $j \neq j'$. Without loss of generality, we assume that $i < i'$. We are going to construct tuples $t_1, t_2 \in R$ such that $t_3 = f(t_1, t_2) \notin R$.

Since Φ is normal, we can assume that the variables $x_1, \ldots, x_n$ are topologically sorted, i.e., if $x_p < x_q$ is an atom of Φ, then $p < q$. In particular, $i < j$ and $i' < j'$. By the topological ordering, any tuple $t \in \mathbb{Q}^n$ with $t[i] < t[j]$ or $t[i'] < t[j']$ belongs to R, whereas no tuple $t \in \mathbb{Q}^n$ with $t[1] \geq \cdots \geq t[n]$ can belong to R. We will use these properties to obtain the desired tuples t_1 and t_2.

We distinguish the following three cases:

Case 1 ($i < j \leq i' < j'$): Let $a_i, a_j, a_{i'}, a_{j'} \in \mathbb{Q}$ and $b_i, b_j, b_{i'}, b_{j'} \in \mathbb{Q}$ be defined by $a_i = 2$, $b_j = b_{i'} = 1$, $a_{j'} = 0$, and $b_i = a_j = a_{i'} = b_{j'} = 3$; see Figure 2 for an illustration. Then, $a_i < a_j$ and $b_{i'} < b_{j'}$. It is also straightforward to

a_i	a_j	$a_{i'}$	$a_{j'}$
$\boxed{2}$	$\boxed{3}$	$\boxed{3}$	$\boxed{0}$
$\boxed{3}$	$\boxed{1}$	$\boxed{1}$	$\boxed{3}$
b_i	b_j	$b_{i'}$	$b_{j'}$

Fig. 2. Choice of $a_i, a_j, a_{i'}, a_{j'} \in \mathbb{Q}$ and $b_i, b_j, b_{i'}, b_{j'} \in \mathbb{Q}$ in Case 1.

verify that $f(a_i, b_i) > f(a_j, b_j) = f(a_{i'}, b_{i'}) > f(a_{j'}, b_{j'})$. Indeed, $min(a_i, b_i) = 2$, $min(a_j, b_j) = min(a_{i'}, b_{i'}) = 1$, and $min(a_{j'}, b_{j'}) = 0$, so the claim is true for $f = min$. For mi and mx, the claim is true, since $min(x, y) > min(x', y')$ implies $mi(x, y) > mi(x', y')$ and $mx(x, y) > mx(x', y')$. Now, Lemma 4 implies that there are tuples $t_1, t_2 \in \mathbb{Q}^n$ such that $t_1[i] < t_1[j]$, $t_2[i'] < t_2[j']$, and $f(t_1[1], t_2[1]) \geq \cdots \geq f(t_1[n], t_2[n])$. Hence, $t_1, t_2 \in R$ and $t_3 = f(t_1, t_2) \notin R$.

Case 2 ($i < i' < j' < j$): Let $a_i, a_{i'}, a_{j'}, a_j \in \mathbb{Q}$ and $b_i, b_{i'}, b_{j'}, b_j \in \mathbb{Q}$ be defined by $a_i = 3$, $b_{i'} = 2$, $a_{j'} = 1$, $b_j = 0$, and $a_{i'} = a_j = b_i = b_{j'} = 4$; see Figure 3 for an illustration. Then, $a_i < a_j$ and $b_{i'} < b_{j'}$. It is also straightforward to verify that $f(a_i, b_i) > f(a_{i'}, b_{i'}) > f(a_{j'}, b_{j'}) > f(a_j, b_j)$. Indeed, $min(a_i, b_i) = 3$, $min(a_{i'}, b_{i'}) = 2$, $min(a_{j'}, b_{j'}) = 1$, and $min(a_j, b_j) = 0$, so the claim is true for $f = min$. For mi and mx, the claim is true, since $min(x, y) > min(x', y')$ implies $mi(x, y) > mi(x', y')$ and $mx(x, y) > mx(x', y')$. Now, Lemma 4 implies that there are tuples $t_1, t_2 \in \mathbb{Q}^n$ such that $t_1[i] < t_1[j]$, $t_2[i'] < t_2[j']$, and $f(t_1[1], t_2[1]) \geq \cdots \geq f(t_1[n], t_2[n])$. Hence, $t_1, t_2 \in R$ and $t_3 = f(t_1, t_2) \notin R$.

$$\begin{array}{cccc} a_i & a_{i'} & a_{j'} & a_j \\ \boxed{3} & \boxed{4} & \boxed{1} & \boxed{4} \\ & & & \\ \boxed{4} & \boxed{2} & \boxed{4} & \boxed{0} \\ b_i & b_{i'} & b_{j'} & b_j \end{array}$$

Fig. 3. Choice of $a_i, a_{i'}, a_{j'}, a_j \in \mathbb{Q}$ and $b_i, b_{i'}, b_{j'}, b_j \in \mathbb{Q}$ in Case 2.

Case 3 ($i < i' < j < j'$): Let $a_i, a_{i'}, a_j, a_{j'} \in \mathbb{Q}$ and $b_i, b_{i'}, b_j, b_{j'} \in \mathbb{Q}$ be defined by $b_i = 3$, $a_{i'} = 2$, $b_j = 1$, $a_{j'} = 0$, $a_i = b_{i'} = 4$, and $a_j = b_{j'} = 5$; see Figure 4 for an illustration. Then, $a_i < a_j$ and $b_{i'} < b_{j'}$. It is also straightforward to

$$\begin{array}{cccc} a_i & a_{i'} & a_j & a_{j'} \\ \boxed{4} & \boxed{2} & \boxed{5} & \boxed{0} \\ & & & \\ \boxed{3} & \boxed{4} & \boxed{1} & \boxed{5} \\ b_i & b_{i'} & b_j & b_{j'} \end{array}$$

Fig. 4. Choice of $a_i, a_{i'}, a_j, a_{j'} \in \mathbb{Q}$ and $b_i, b_{i'}, b_j, b_{j'} \in \mathbb{Q}$ in Case 3.

verify that $f(a_i, b_i) > f(a_{i'}, b_{i'}) > f(a_j, b_j) > f(a_{j'}, b_{j'})$. Indeed, $min(a_i, b_i) = 3$, $min(a_{i'}, b_{i'}) = 2$, $min(a_j, b_j) = 1$, and $min(a_{j'}, b_{j'}) = 0$, so the claim is true for $f = min$. For mi and mx, the claim is true, since $min(x, y) > min(x', y')$ implies $mi(x, y) > mi(x', y')$ and $mx(x, y) > mx(x', y')$. Now, Lemma 4 implies that there are tuples $t_1, t_2 \in \mathbb{Q}^n$ such that $t_1[i] < t_1[j]$, $t_2[i'] < t_2[j']$, and $f(t_1[1], t_2[1]) \geq \cdots \geq f(t_1[n], t_2[n])$. Hence, $t_1, t_2 \in R$ and $t_3 = f(t_1, t_2) \notin R$.

Altogether, this concludes the proof. $\square$

An *ll-Horn formula* is a formula of the form $\bigvee_{i=1}^{n} x_0 > x_i$. A *dual-ll-Horn formula* is a formula of the form $\bigvee_{i=1}^{n} x_i > x_0$.

Lemma 3. Let $R \subseteq \mathbb{Q}^n$ be defined by a normal disjunctive formula $\Phi(x_1, \ldots, x_n)$ over $(\mathbb{Q}, <)$. If R is preserved under min, mi, mx, ll, or one of their duals, then Φ is a ll-Horn or a *dual-ll*-Horn formula.

Proof. If R is preserved under ll or its dual, then the lemma follows from [8]. Otherwise, let $\Phi(x_1, \ldots, x_n) = \bigvee_{i=1}^{k} x_{s_i} < x_{t_i}$. Without loss of generality, we can assume that any two pairs (s_p, t_p), (s_q, t_q) with $p \neq q$ are distinct. If $k = 1$, then Φ is ll-Horn. Otherwise, by Lemma 5, for each $j \in \{2, \ldots, k\}$ we have $s_1 = s_j$ or $t_1 = t_j$. To show that Φ is a ll-Horn or a *dual-ll*-Horn formula, it suffices to show that there are no two $j, j' \in \{2, \ldots, k\}$ such that $s_1 = s_j$ and $t_1 = t_{j'}$.

For a contradiction, suppose that there are $j, j' \in \{2, \ldots, k\}$ with $s_1 = s_j$ and $t_1 = t_{j'}$. By Lemma 5, we either have $s_j = s_{j'}$ or $t_j = t_{j'}$. In the first case, we have $(s_1, t_1) = (s_j, t_{j'}) = (s_{j'}, t_{j'})$, and in the second case we have $(s_1, t_1) = (s_j, t_{j'}) = (s_j, t_j)$. Both cases violate our assumption that any two pairs (s_p, t_p), (s_q, t_q) with $p \neq q$ are distinct. Consequently, there are no two $j, j' \in \{2, \ldots, k\}$ such that $s_1 = s_j$ and $t_1 = t_{j'}$, which implies that Φ is a *ll*-Horn or a *dual-ll*-Horn formula. $\square$