

# On the Data Complexity of Ontology-Mediated Queries with a Covering Axiom

O. Gerasimova<sup>1</sup>, S. Kikot<sup>2</sup>, V. Podolskii<sup>3,1</sup>, and M. Zakharyashev<sup>2</sup>

<sup>1</sup> National Research University Higher School of Economics, Moscow, Russia

<sup>2</sup> Birkbeck, University of London, U.K.

<sup>3</sup> Steklov Mathematical Institute, Moscow, Russia

**Abstract.** This paper reports on our ongoing work that aims at a classification of conjunctive queries  $q$  according to the data complexity of answering ontology-mediated queries  $(\{A \sqsubseteq T \sqcup F\}, q)$ . We give examples of queries from the complexity classes  $\mathcal{C} \in \{\text{AC}^0, \text{L}, \text{NL}, \text{P}, \text{coNP}\}$ , and obtain a few syntactical conditions for  $\mathcal{C}$ -membership and  $\mathcal{C}$ -hardness.

## 1 Introduction

The *OWL 2 QL* profile of *OWL 2*—as well as the underlining description logics from the *DL-Lite* family [4, 2]—were designed to ensure that every ontology-mediated query (OMQ, for short)  $(\mathcal{T}, q)$  with an *OWL 2 QL* ontology  $\mathcal{T}$  and a conjunctive query (CQ)  $q$  is first-order (FO) rewritable. However, when developing ontologies for ontology-based data access (OBDA) [10] applications, domain experts are often tempted to use axioms with constructs that are not available in *OWL 2 QL*. For example, the NPD FactPages ontology,<sup>4</sup> which was created to facilitate querying the datasets of the Norwegian Petroleum Directorate,<sup>5</sup> contains cardinality restrictions and covering axioms of the form  $A \sqsubseteq B_1 \sqcup \dots \sqcup B_n$ . Typical answers to the question whether such axioms could have a negative impact on OMQ rewriting are as follows: (i) the data satisfies the axioms anyway (because of the database schema), (ii) our ‘real-world queries’ are never affected by them, and (iii) OBDA systems such as Ontop drop everything outside *OWL 2 QL*. Ideally, of course, we would rather want our system to detect automatically whether the given OMQ is FO-rewritable and alert the user if this is not so. Furthermore, in case of non-FO-rewritability, we might want the system to check whether a datalog rewriting is possible, and so on. From the complexity-theoretic point of view, we are thus interested in the data complexity of answering a given OMQ with an expressive ontology.

A systematic investigation of this problem was started in [3], which showed among other results that answering OMQs of the form  $(Dis_n, u)$ , where  $u$  is a *union* of CQs (UCQ) and  $Dis_n = \{A \sqsubseteq B_1 \sqcup \dots \sqcup B_n\}$ , is polynomially equivalent to the constraint satisfaction problems  $\text{CSP}(\mathfrak{A})$ . In particular, a P/coNP dichotomy for such OMQs would give a dichotomy for CSPs, thereby confirming the Feder-Vardi conjecture. As

<sup>4</sup> <http://sws.ifi.uio.no/project/npd-v2/>

<sup>5</sup> <http://factpages.npd.no/factpages/>

shown in [7], answering CQs with basic schema.org ontologies (in particular,  $Dis_n$ ) and CQs of  $qvar$ -size  $\leq 2$  is in P for combined complexity, where  $q$  is of  $qvar$ -size  $n$  if the restriction of  $q$  to its quantified variables is a disjoint union of CQs with at most  $n$  variables each. Moreover, FO- and datalog-rewritability of OMQs of the form  $(\mathcal{T}, \mathbf{u})$ , where  $\mathcal{T}$  is a schema.org ontology and  $\mathbf{u}$  is a UCQ, are decidable in NEXPTIME. It has also been recently established in [5] that checking FO-rewritability of OMQs with ontologies formulated in any description logic between  $\mathcal{ALCI}$  and  $\mathcal{SHI}$  is 2NEXPTIME-complete. Datalog rewritability of OMQs with ontologies given in disjunctive datalog has been investigated in [8].

In this paper, we consider one fixed non-Horn ontology  $Dis = \{A \sqsubseteq T \sqcup F\}$ . Ultimately aiming at a complete classification of CQs  $q$  according to the data complexity of answering OMQs  $Q = (Dis, q)$ , here we present our initial observations about this problem. Ideally, we would like to obtain transparent necessary and sufficient conditions relating the structure of  $q$ —say, the way how  $T$  and  $F$  occur in it—with the complexity of answering  $Q$ . For example, one such condition guaranteeing datalog rewritability, and so tractability of answering  $Q$  follows from [8, Theorem 27]: it suffices that  $q$  contains at most one occurrence of  $F$  or at most one occurrence of  $T$ . We obtain a few conditions in the same spirit for the complexity classes  $AC^0$ , L, NL and P. We also give quite a few simple and instructive CQs distinguishing between NL and P, and develop techniques for establishing P and CONP lower bounds.

## 2 Preliminaries

In our context, a *conjunctive query* (CQ) is a first-order (FO) formula of the form  $q(\mathbf{x}) = \exists \mathbf{y} \varphi(\mathbf{x}, \mathbf{y})$ , where  $\varphi$  is a conjunction of unary or binary atoms  $P(z)$  with  $z \subseteq \mathbf{x} \cup \mathbf{y}$ . *Unions of conjunctive queries* (UCQ) is a disjunction of conjunctive queries. Given an ABox (or data instance)  $\mathcal{A}$ , we denote by  $\text{ind}(\mathcal{A})$  the set of individual names that occur in  $\mathcal{A}$ . A tuple  $\mathbf{a} \subseteq \text{ind}(\mathcal{A})$  is a *certain answer* to the OMQ  $Q = (Dis, q(\mathbf{x}))$  over  $\mathcal{A}$  if  $\mathfrak{M} \models q(\mathbf{a})$ , for every model  $\mathfrak{M}$  of  $Dis \cup \mathcal{A}$ ; in this case we write  $Dis, \mathcal{A} \models q(\mathbf{a})$ . If the set  $\mathbf{x}$  of *answer variables* is empty, a *certain answer* to  $Q$  over  $\mathcal{D}$  is ‘yes’ if  $\mathfrak{M} \models q$ , for every model  $\mathfrak{M}$  of  $Dis \cup \mathcal{A}$ , and ‘no’ otherwise. OMQs and CQs without answer variables  $\mathbf{x}$  are called *Boolean*. We often regard CQs as *sets* of their atoms. In this paper, we assume that the all CQs are *connected*.

Let  $Q = (Dis, q(\mathbf{x}))$  be a fixed OMQ. By *answering*  $Q$ , we understand the problem of checking, given an ABox  $\mathcal{A}$  and a tuple  $\mathbf{a} \subseteq \text{ind}(\mathcal{A})$ , whether  $Dis, \mathcal{A} \models q(\mathbf{a})$ . It is readily seen that this problem is always in CONP. It is in the complexity class  $AC^0$  if there is an FO-formula  $q'(\mathbf{x})$ , called an *FO-rewriting* of  $Q$ , such that  $Dis, \mathcal{A} \models q(\mathbf{a})$  iff  $q'(\mathbf{a})$  holds in the model given by  $\mathcal{A}$ , for any ABox  $\mathcal{A}$  and any tuple  $\mathbf{a} \subseteq \text{ind}(\mathcal{A})$ .

A *datalog program*,  $\Pi$ , is a finite set of *rules* of the form  $\forall z (\gamma_0 \leftarrow \gamma_1 \wedge \dots \wedge \gamma_m)$ , where each  $\gamma_i$  is an atom  $Q(\mathbf{y})$  with  $\mathbf{y} \subseteq z$  or an equality  $(z = z')$  with  $z, z' \in z$ . (As usual, we omit  $\forall z$ .) The atom  $\gamma_0$  is the *head* of the rule, and  $\gamma_1, \dots, \gamma_m$  its *body*. All variables in the head must occur in the body, and  $=$  can only occur in the body. The predicates in the heads of rules in  $\Pi$  are *IDB predicates*, the rest (including  $=$ ) *EDB predicates*. A program  $\Pi$  is called *linear* if the body of every rule in  $\Pi$  contains at most one IDB predicate.

A *datalog query* is a pair  $(\Pi, G(\mathbf{x}))$ , where  $\Pi$  is a datalog program and  $G(\mathbf{x})$  an atom. A tuple  $\mathbf{a} \subseteq \text{ind}(\mathcal{A})$  is an *answer to*  $(\Pi, G(\mathbf{x}))$  over an ABox  $\mathcal{A}$  if  $G(\mathbf{a})$  holds in the first-order structure with domain  $\text{ind}(\mathcal{A})$  obtained by closing  $\mathcal{A}$  under the rules in  $\Pi$ ; in this case we write  $\Pi, \mathcal{A} \models G(\mathbf{a})$ . A datalog query  $(\Pi, G(\mathbf{x}))$  is a *datalog rewriting* of an OMQ  $\mathbf{Q} = (\mathcal{D}is, \mathbf{q}(\mathbf{x}))$  in case  $\mathcal{D}is, \mathcal{A} \models \mathbf{q}(\mathbf{a})$  iff  $\Pi, \mathcal{A} \models G(\mathbf{a})$ , for any ABox  $\mathcal{A}$  and any  $\mathbf{a} \subseteq \text{ind}(\mathcal{A})$ . The *evaluation problem* for  $(\Pi, G(\mathbf{x}))$ —that is, checking, given an ABox  $\mathcal{A}$  and a tuple  $\mathbf{a} \subseteq \text{ind}(\mathcal{A})$ , whether  $\Pi, \mathcal{A} \models G(\mathbf{a})$ —is known to be in P; for linear  $\Pi$ , this problem is in NL; see [6] and references therein.

### 3 AC<sup>0</sup>

By a *solitary occurrence* of  $F$  in a CQ  $\mathbf{q}$  we mean any  $F(x) \in \mathbf{q}$  such that  $T(x) \notin \mathbf{q}$ ; likewise, a *solitary occurrence* of  $T$  in  $\mathbf{q}$  is any  $T(x) \in \mathbf{q}$  such that  $F(x) \notin \mathbf{q}$ .

**Theorem 1.** *For any CQ  $\mathbf{q}$  without solitary occurrences of  $F$  (or  $T$ ), answering the OMQ  $\mathbf{Q} = (\mathcal{D}is, \mathbf{q})$  is in AC<sup>0</sup>.*

*Proof.* We show that  $\mathcal{D}is, \mathcal{A} \models \mathbf{q}(\mathbf{a})$  iff  $\mathcal{A} \models \mathbf{q}(\mathbf{a})$ . Suppose that  $\mathcal{A} \not\models \mathbf{q}(\mathbf{a})$  and  $F(x) \in \mathbf{q} \Rightarrow T(x) \in \mathbf{q}$ . Take  $\mathcal{A}' = \mathcal{A} \cup \{F(a) \mid a \in \text{ind}(\mathcal{A}) \wedge T(a) \notin \mathcal{A}\}$ . Clearly,  $\mathcal{A}' \models \mathcal{D}is$  and  $\mathcal{A}' \not\models \mathbf{q}(\mathbf{a})$ . The converse direction is trivial.

In particular, answering any OMQ  $\mathbf{Q} = (\mathcal{D}is, \mathbf{q})$ , where  $\mathbf{q}$  does not contain one of  $F$  or  $T$ , is in AC<sup>0</sup>. This observation can be easily generalised to OMQs with ontologies  $\mathcal{D}is_n = \{A \sqsubseteq B_1 \sqcup \dots \sqcup B_n\}$ , for  $n \geq 2$ :

**Theorem 2.** *Suppose  $\mathbf{q}$  is any CQ that does not contain an occurrence of  $B_i$ , for some  $i$  ( $1 \leq i \leq n$ ). Then answering the OMQ  $\mathbf{Q} = (\mathcal{D}is_n, \mathbf{q})$  is in AC<sup>0</sup>.*

Thus, only those CQs can ‘feel’  $\mathcal{D}is_n$  as far as FO-rewritability is concerned that contain all the  $B_n$  (which makes them quite complex in practice). Theorem 1 also shows that  $\mathbf{Q} = (\mathcal{D}is, \mathbf{q})$  satisfying the respective condition has a trivial FO-rewriting, viz.  $\mathbf{q}$  itself. This is not accidental as shown by the following observation:

**Proposition 1.** *If  $\mathbf{Q} = (\mathcal{D}is, \mathbf{q})$  is in AC<sup>0</sup>, then  $\mathbf{q}$  is a rewriting of  $\mathbf{Q}$ .*

*Proof.* By [3, Proposition 5.9], if  $\mathbf{Q}$  is FO-rewritable, it has a UCQ rewriting. Then there is a homomorphism from  $\mathbf{q}$  to any CQ  $\mathbf{q}'$  in this rewriting.

We do not know yet whether the sufficient condition for FO-rewritability given by Theorem 1 is also a necessary one for *minimal* CQs  $\mathbf{q}$  (that are not equivalent to any of their proper subqueries). For non-minimal CQs, this is not the case as shown by  $F \xleftarrow{R} \circ \xrightarrow{R} FT \xleftarrow{R} \circ \xrightarrow{R} T$  which is in AC<sup>0</sup> because it is equivalent to the CQ  $\circ \xrightarrow{R} FT \xleftarrow{R} \circ$ . Below we obtain some partial results showing how a single  $F$ -atom and a single  $T$ -atom in  $\mathbf{q}$  can cause L- and NL-hardness.

## 4 L and NL

We say that a Boolean CQ  $q$  is an *F-T-CQ* if it has exactly one atom of the form  $F(x)$ , exactly one atom of the form  $T(y)$ , and the variables  $x$  and  $y$  are distinct.

**Theorem 3.** *Answering any OMQ  $Q = (\mathcal{D}is, q)$  with an F-T-CQ  $q$  is L-hard.*

*Proof.* The proof is by reduction to the reachability problem for undirected graphs, which is known to be L-complete; see, e.g., [1]. Let  $q'$  be the CQ obtained from  $q$  by removing the atoms  $F(x)$  and  $T(y)$ . Suppose we are given an undirected graph  $G = (V, E)$  and two vertices  $s, t \in V$ . It will be convenient to regard  $G$  as a directed graph such that  $(u, v) \in E$  iff  $(v, u) \in E$ , for any  $u, v \in V$ . We encode  $G$  by means of an ABox  $\mathcal{A}_G$  that is obtained from  $G$  as follows. For every edge  $e = (u, v) \in E$ , let  $q'_e$  be the set of atoms in  $q'$  with  $x$  renamed to  $u$ ,  $y$  to  $v$  and all other variables  $z$  to  $z_e$ . Then  $\mathcal{A}_G$  comprises all such  $q'_e$ , for  $e \in E$ , as well as  $F(s)$ ,  $T(t)$  and  $A(v)$ , for  $v \in V \setminus \{s, t\}$ . Our aim is to show that  $s \rightarrow_G t$  iff  $\mathcal{D}is, \mathcal{A}_G \models q$ .

Suppose  $s \rightarrow_G t$ , that is, there exists a path  $s = v_0, v_1, \dots, v_n = t$  in  $G$  with  $e_i = (v_i, v_{i+1}) \in E$ , for  $i < n$ . Consider an arbitrary model  $\mathcal{I}$  of  $\mathcal{D}is$  and  $\mathcal{A}_G$ . Since  $\mathcal{I} \models \mathcal{D}is$  and  $F(s), T(t), A(v_i)$ , for  $1 \leq i < n$ , are all in  $\mathcal{A}_G$ , we can find some  $i < n$  such that  $\mathcal{I} \models F(v_i)$  and  $\mathcal{I} \models T(v_{i+1})$ . As  $q'_{e_i}$  is an isomorphic copy of  $q'$ , we obtain  $\mathcal{I} \models q$ . Conversely, suppose  $s \not\rightarrow_G t$ . Define an interpretation  $\mathcal{I}$  by extending the ABox  $\mathcal{A}_G$  with  $F^{\mathcal{I}} = \{v \in V \mid s \rightarrow_G v\}$  and  $T^{\mathcal{I}} = \{v \in V \mid s \not\rightarrow_G v\}$ . Clearly,  $\mathcal{I}$  is a model of  $\mathcal{D}is$ . By the construction, the elements of the connected component of  $\mathcal{I}$  containing  $s$  cannot be instances of  $T$ , while the remaining elements of  $\mathcal{I}$  cannot be instances of  $F$ . Since  $q$  is connected, it follows that  $\mathcal{I} \not\models q$ .

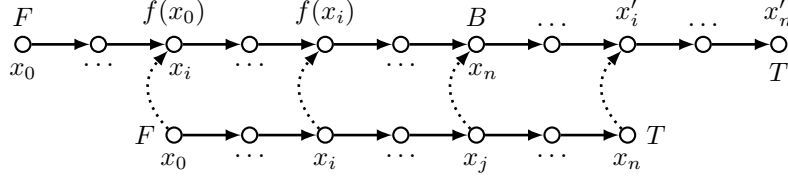
We call a Boolean CQ  $q$  *linear-directed* if all of its variables can be arranged in a sequence  $v_0, \dots, v_m$  such that all binary predicates in  $q$  are of the form  $R(v_i, v_{i+1})$ , for some  $i$ ,  $0 \leq i < m$ .

**Theorem 4.** *Answering any OMQ  $Q = (\mathcal{D}is, q)$  with a linear-directed CQ  $q$  containing both a solitary  $F$  and a solitary  $T$  is NL-hard.*

*Proof.* Suppose  $F(v_k) \in q$ ,  $T(v_k) \notin q$  and  $F(v_l) \notin q$ ,  $T(v_l) \in q$ , for some  $k, l$  with  $0 \leq k < l \leq m$ . We rename the sequence  $v_k, \dots, v_l$  to  $x_0, \dots, x_n$ . The proof proceeds by reduction to the reachability problem in directed graphs, which is known to be NL-complete; see, e.g., [1]. Given a *directed* graph  $G = (V, E)$  and vertices  $s, t \in V$ , we construct the ABox  $\mathcal{A}_G$  in the same way as in the proof of Theorem 3 treating  $x_0$  as  $x$  and  $x_n$  as  $y$ . Again, we show that  $s \rightarrow_G t$  iff  $\mathcal{D}is, \mathcal{A}_G \models q$ . The implication ( $\Rightarrow$ ) is established exactly as above.

To prove ( $\Leftarrow$ ), we assume that  $s \not\rightarrow_G t$  and consider the same model  $\mathcal{I}$  as defined in the proof of Theorem 3. Taking account of linear-directedness of  $q$ , we immediately conclude that there is no homomorphism  $h: q \rightarrow \mathcal{I}$  with  $h(x_0) \in V$ . It remains to show that there is no homomorphism  $h: q \rightarrow \mathcal{I}$  with  $h(x_0) \notin V$  either. Suppose to the contrary that such a homomorphism exists. Then there exist  $B \in \{F, T\}$  and a homomorphism  $f: q \rightarrow (\mathcal{A}_{G_2} \cup \{B(r)\})$ , where  $G_2 = (\{s, r, t\}, \{(s, r), (r, t)\})$ . We denote the points of  $\mathcal{A}_{G_2}$  between  $s$  and  $r$  by  $x_0, x_1, \dots, x_n$  and those between  $r$  and

$t$  by  $x_n, x'_1, \dots, x'_n$ . By comparing the lengths of appropriate segments of  $\mathbf{q}$ , we obtain  $f(x_0) = x_i$ , for some  $i$  ( $0 < i < n$ ). As  $F(x_0) \in \mathbf{q}$ , we must have  $F(x_i) \in \mathbf{q}$ ; see the picture below. As  $f(x_i) = x_{2i}$  if  $2i \leq n$ , and  $f(x_i) = x'_{2i \bmod n}$  otherwise, we also have  $F(x_{2i \bmod n}) \in \mathbf{q}$ ; more generally,  $F(x_{ki \bmod n}) \in \mathbf{q}$  for all natural  $k$ . Now, since the equation of the form ' $iX = n \bmod n$ ' always has a solution,  $F(x_n) \in \mathbf{q}$ , which is impossible if  $B = T$ . If  $B = F$ , we use a similar argument starting from  $T(x_i) \in \mathbf{q}$  and show that  $T(x_n) \in \mathbf{q}$ , which is again a contradiction.



Theorems 1 and 4 give the following *dichotomy* for OMQs  $\mathbf{Q} = (\mathcal{D}is, \mathbf{q})$  with linear-directed CQs  $\mathbf{q}$ :

- either  $\mathbf{q}$  does not contain a solitary  $F$  or a solitary  $T$ , and answering  $\mathbf{Q}$  is in  $\text{AC}^0$ ,
- or  $\mathbf{q}$  contains both solitary  $F$  and  $T$ , and answering  $\mathbf{Q}$  is NL-hard.

We now complement the sufficient conditions of L- and NL-hardness obtained above with sufficient conditions of OMQ answering in L- and NL.

A CQ  $\mathbf{q}'(x, y)$  is *symmetric* if the CQs  $\mathbf{q}'(x, y)$  and  $\mathbf{q}'(y, x)$  are equivalent in the sense that  $\mathbf{q}'(a, b)$  holds in  $\mathcal{A}$  iff  $\mathbf{q}'(b, a)$  holds in  $\mathcal{A}$ , for any ABox  $\mathcal{A}$  and  $a, b \in \text{ind}(\mathcal{A})$ .

**Theorem 5.** *Let  $\mathbf{Q} = (\mathcal{D}is, \mathbf{q})$  be any OMQ such that*

$$\mathbf{q} = \exists x, y (F(x) \wedge \mathbf{q}'_1(x) \wedge \mathbf{q}'(x, y) \wedge \mathbf{q}'_2(y) \wedge T(y)),$$

*for some connected CQs  $\mathbf{q}'(x, y)$ ,  $\mathbf{q}'_1(x)$  and  $\mathbf{q}'_2(y)$  that do not contain solitary  $T$  and  $F$ , and  $\mathbf{q}'(x, y)$  is symmetric. Then answering  $\mathbf{Q}$  can be done in L.*

*Proof.* It is not hard to show that, for any ABox  $\mathcal{A}$ , we have  $\mathcal{D}is, \mathcal{A} \models \mathbf{q}$  iff there exist  $v_0, v_1, \dots, v_n \in \text{ind}(\mathcal{A})$ , for some  $n \geq 1$ , such that the following conditions hold:

- $F(v_0), A(v_1), \dots, A(v_{n-1}), T(v_n) \in \mathcal{A}$ ;
- $\mathcal{A} \models \mathbf{q}'(v_i, v_{i+1})$  for  $0 \leq i < n$ ;
- $\mathcal{A} \models \mathbf{q}'_1(v_i)$  for  $0 \leq i < n$ ;
- $\mathcal{A} \models \mathbf{q}'_2(v_i)$  for  $1 \leq i \leq n$ .

It remains to observe that checking these conditions reduces to checking  $V_T$ - $V_F$  reachability in the undirected graph  $G_{\mathcal{A}} = (V_{\mathcal{A}}, E_{\mathcal{A}})$  defined below. The vertices in  $G_{\mathcal{A}}$  comprise the set  $V_{\mathcal{A}} = V_T \cup V_A \cup V_F$ , where

- $V_T = \{v \in \text{ind}(\mathcal{A}) \mid \mathcal{A} \models T(v) \wedge \mathbf{q}'_2(v)\}$ ;
- $V_A = \{v \in \text{ind}(\mathcal{A}) \mid \mathcal{A} \models A(v) \wedge \mathbf{q}'_1(v) \wedge \mathbf{q}'_2(v)\}$ ;
- $V_F = \{v \in \text{ind}(\mathcal{A}) \mid \mathcal{A} \models F(v) \wedge \mathbf{q}'_1(v)\}$ .

The edges in  $G_{\mathcal{A}}$  comprise the set  $E_{\mathcal{A}} = E_{TA} \cup E_{AA} \cup E_{FA}$ , where

- $E_{all} = \{(x, y) \mid \mathcal{A} \models \mathbf{q}'(x, y)\}$ ;
- $E_{TA} = \{(x, y) \in E_{all} \mid (x \in V_T \wedge y \in V_A) \vee (y \in V_T \wedge x \in V_A)\}$ ;
- $E_{AA} = \{(x, y) \in E_{all} \mid x \in V_A \wedge y \in V_A\}$ ;
- $E_{FA} = \{(x, y) \in E_{all} \mid (x \in V_F \wedge y \in V_A) \vee (y \in V_F \wedge x \in V_A)\}$ .

It is readily seen that  $G_{\mathcal{A}} = (V_{\mathcal{A}}, E_{\mathcal{A}})$  is undirected in the sense that, for all of its vertices  $u$  and  $v$ ,  $(u, v) \in E_{\mathcal{A}}$  iff  $(v, u) \in E_{\mathcal{A}}$ .

If we do not require  $\mathbf{q}'(x, y)$  to be symmetric, the complexity upper bound increases to NL:

**Theorem 6.** *Let  $\mathbf{Q} = (\mathcal{D}is, \mathbf{q})$  be any OMQ such that*

$$\mathbf{q} = \exists x, y (F(x) \wedge T(y) \wedge \mathbf{q}'(x, y)),$$

*for some connected CQ  $\mathbf{q}'(x, y)$  without solitary occurrences of  $F$  and  $T$ . Then answering  $\mathbf{Q}$  can be done in NL.*

*Proof.* We claim that the datalog query  $(\Pi, G)$  with the following linear datalog program  $\Pi$ , where  $\tilde{\mathbf{q}}'$  is the result of omitting all the  $\exists$  from  $\mathbf{q}'$ :

$$\begin{aligned} G &\leftarrow F(x) \wedge \tilde{\mathbf{q}}'(x, y) \wedge P(y) \\ P(x) &\leftarrow T(x) \\ P(x) &\leftarrow A(x) \wedge \tilde{\mathbf{q}}'(x, y) \wedge P(y) \end{aligned}$$

is a datalog rewriting of  $\mathbf{Q}$ . Indeed, if  $\Pi, \mathcal{A} \models G$  then there are  $v_0, v_1, \dots, v_n \in \text{ind}(\mathcal{A})$  such that  $F(v_0), A(v_1), \dots, A(v_{n-1}), T(v_n) \in \mathcal{A}$  and  $\mathbf{q}'(v_i, v_{i+1})$  holds in  $\mathcal{A}$ , for  $0 \leq i < n$ . Clearly, in any model  $\mathcal{I}$  of  $\mathcal{D}is$  and  $\mathcal{A}$  there is  $i$  with  $\mathcal{I} \models F(v_i) \wedge T(v_{i+1})$ . It follows that  $\mathcal{D}is, \mathcal{A} \models \mathbf{q}$ .

Conversely, suppose  $\Pi, \mathcal{A} \not\models G$ . Let  $V_P = \{v \in \text{ind}(\mathcal{A}) \mid \Pi, \mathcal{A} \models P(v)\}$ . Define a model  $\mathcal{I}$  of  $\mathcal{D}is$  with domain  $\text{ind}(\mathcal{A})$  by setting

$$T^{\mathcal{I}} = \{v \mid T(v) \in \mathcal{A}\} \cup \{v \in V_P \mid A(v) \in \mathcal{A}\}, \quad F^{\mathcal{I}} = F^{\mathcal{A}} \cup \{v \notin V_P \mid A(v) \in \mathcal{A}\}.$$

We claim that  $\mathcal{I} \not\models \mathbf{q}$ . Indeed, otherwise there is a homomorphism  $h: \mathbf{q} \rightarrow \mathcal{I}$ . As  $h(y) \in T^{\mathcal{I}}$ , we have  $\Pi, \mathcal{A} \models P(h(y))$ . As  $h(x) \in F^{\mathcal{I}}$ , we have either  $F(h(x)) \in \mathcal{A}$  or  $A(h(x)) \in \mathcal{A}$ , contrary to  $\Pi, \mathcal{A} \not\models G$ .

The sufficient conditions of Theorems 5 and 6 only apply to CQs with exactly one solitary occurrence of  $F$  and exactly one solitary occurrence of  $T$ . What happens if we allow more than one solitary occurrences of  $F$  or  $T$ ?

## 5 P

The following result is a consequence of [8, Theorem 27]:

**Theorem 7.** *Let  $\mathbf{Q} = (\mathcal{D}is, \mathbf{q})$  be any OMQ such that*

$$\mathbf{q} = \exists x, y_1, \dots, y_n (F(x) \wedge T(y_1) \wedge \dots \wedge T(y_n) \wedge \mathbf{q}'(x, y_1, \dots, y_n)),$$

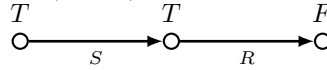
*for some connected CQ  $\mathbf{q}'(x, y_1, \dots, y_n)$  without solitary occurrences of  $T$  and  $F$ . Then answering  $\mathbf{Q}$  can be done in P.*

Indeed, for any ABox  $\mathcal{A}$ , we have  $\mathcal{D}is, \mathcal{A} \models \mathbf{q}$  iff  $\mathcal{I}, \mathcal{A} \models G$ , where  $\mathcal{I}$  is the following datalog program and  $\tilde{\mathbf{q}}'$  is the result of omitting all the  $\exists$  from  $\mathbf{q}'$ :

$$\begin{aligned} G &\leftarrow F(x) \wedge \tilde{\mathbf{q}}'(x, y_1, \dots, y_n) \wedge P(y_1) \wedge \dots \wedge P(y_n) \\ P(x) &\leftarrow T(x) \\ P(x) &\leftarrow A(x) \wedge \tilde{\mathbf{q}}'(x, y_1, \dots, y_n) \wedge P(y_1) \wedge \dots \wedge P(y_n). \end{aligned}$$

Is the P-upper bound of Theorem 7 optimal? The following example gives a typical OMQ in the scope of that theorem answering which is P-hard.

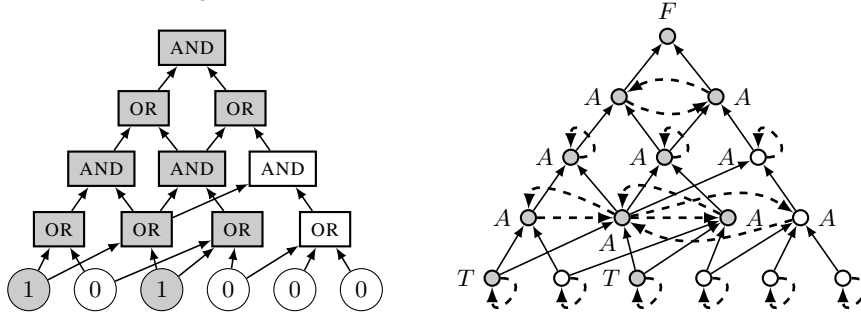
*Example 1.* We show that  $\mathbf{Q} = (\mathcal{D}is, \mathbf{q})$  is P-hard for  $\mathbf{q}$  shown in the picture below.



The proof is by reduction of the alternating monotone circuit evaluation problem, which is known to be P-complete [9]. An example of an alternating monotone circuit is shown in the picture below. Given such a circuit  $\mathcal{C}$  and an input  $\alpha$ , we define an ABox  $\mathcal{A}_{\mathcal{C}}^{\alpha}$  as the set of the following atoms:

- $R(g, h)$ , if a gate  $g$  is an input of a gate  $h$ ;
- $S(g, h)$ , if  $g$  and  $h$  are distinct inputs of some AND-gate;
- $S(g, g)$ , if  $g$  is an input gate or a non-output AND-gate;
- $T(g)$ , if  $g$  is an input gate with 1 under  $\alpha$ ;
- $F(g)$ , for the only output gate  $g$ ;
- $A(g)$ , for those  $g$  that are neither inputs nor the output.

To illustrate, the picture below shows an alternating monotone circuit  $\mathcal{C}$ , an input  $\alpha$  for it, and the ABox  $\mathcal{A}_{\mathcal{C}}^{\alpha}$ , where the solid arrows represent  $R$  and the dashed ones  $S$ :



One can show that  $\mathcal{C}(\alpha) = 1$  iff  $\mathcal{D}is, \mathcal{A}_{\mathcal{C}}^{\alpha} \models \mathbf{q}$ .

Curiously, by changing  $S$  to  $R$  in the CQ from Example 1, we obtain an OMQ that is NL-complete as follows from Theorem 8 below.

## 6 NL vs. P

**Theorem 8.** Answering any OMQ  $\mathbf{Q} = (\mathcal{D}is, \mathbf{q}_n)$  with

$$\mathbf{q}_n = \exists x_1, \dots, x_n, y \bigwedge_{i=1}^{n-1} (T(x_i) \wedge R(x_i, x_{i+1})) \wedge T(x_n) \wedge R(x_n, y) \wedge F(y),$$

for  $n \geq 1$ , is NL-complete.

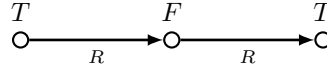
*Proof.* The lower bound follows from Theorem 4. The proof of the upper one is by reduction to directed reachability. We split  $q_n$  into two CQs:

$$\begin{aligned} q'_n &= \exists x_1, \dots, x_n \bigwedge_{i=1}^{n-1} (T(x_i) \wedge R(x_i, x_{i+1})) \wedge T(x_n), \\ q &= \exists x, y (T(x) \wedge R(x, y) \wedge F(y)). \end{aligned}$$

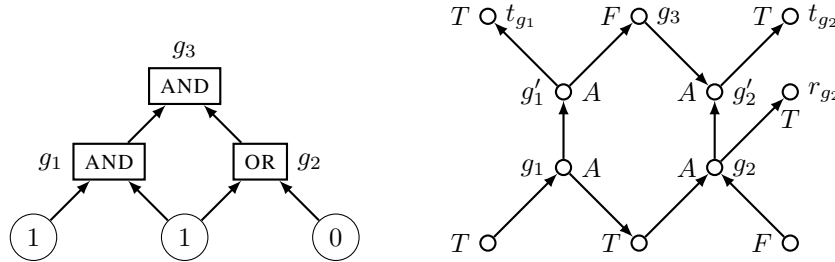
One can show that, for any ABox  $\mathcal{A}$ , we have  $\text{Dis}, \mathcal{A} \models q_n$  iff there exist a homomorphism  $f: q'_n \rightarrow \mathcal{A}$  and a directed  $R$ -path  $f(x_n), v_0, v_1, \dots, v_m \in \text{ind}(\mathcal{A})$  such that  $A(v_i) \in \mathcal{A}$ , for  $i = 1, \dots, m-1$ , and  $F(v_m) \in \mathcal{A}$ . Clearly, this criterion reduces to directed reachability.

To further illustrate how minor modifications to the structure of CQs can send them to different complexity classes, we collect in Table 1 a number of CQs in the scope of Theorem 7, some of which turn out to be NL-complete, while others are P-complete. (All the omitted labels on the arrows in Table 1 are assumed to be  $R$ ,  $-/A$  means either blank or  $A$ , and  $FT/A$  means either  $FT$  or  $A$ ).

Here, we only sketch the proof of P-hardness for the OMQ  $(\text{Dis}, q)$ , where  $q$  is



The proof is by reduction of the monotone circuit evaluation problem. Given a monotone circuit  $C$  and an input  $\alpha$ , we define an ABox  $\mathcal{A}_C^\alpha$  as the following labelled directed graph, all of whose edges are labelled with  $R$ . For each gate  $g$  of  $C$  except the inputs and output, the graph contains two vertices  $g$  and  $g'$  labelled with  $A$ ; the output gate  $g$  gives rise to only one vertex  $g$  labelled with  $F$ , while each input gate  $g$  to only one vertex  $g'$  labelled according to  $\alpha$ . For an OR-gate  $g = h_1 \vee h_2$ , we have the directed edges  $(h'_1, g), (h'_2, g), (g, r_g)$ , where  $r_g$  is a new vertex labelled with  $T$ . For an AND-gate  $g = h_1 \wedge h_2$ , we have the edges  $(h'_1, g), (g, h'_2)$ . Also, for each gate  $g$ , we have the edges  $(g, g'), (g', t_g)$ , where  $t_g$  is a new vertex labelled with  $T$ . An example illustrating the construction is given below. One can show that  $C(\alpha) = 1$  iff  $\text{Dis}, \mathcal{A}_C^\alpha \models q$ .



The membership in NL for the CQs in the left column of Table 1 can be shown by constructing appropriate linear datalog programs. For example, answering the OMQ



| NL-complete | P-complete |
|-------------|------------|
|             |            |
|             |            |
|             |            |
|             |            |
|             |            |
|             |            |
|             |            |
|             |            |

**Table 1.** NL- and P-complete OMQs in the scope of Theorem 7.

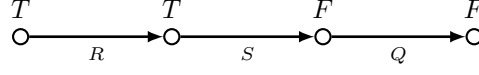
with the last CQ of the left column can be done by the following linear program:

$$\begin{aligned}
P(x) &\leftarrow R(x, y), T(y), R(x, z), R(z, v), T(v) \\
P(x) &\leftarrow R(x, y), T(y), R(x, z), R(z, v), P(v), A(v) \\
P(x) &\leftarrow R(x, y), P(y), A(y) \\
G &\leftarrow P(x), F(x)
\end{aligned}$$

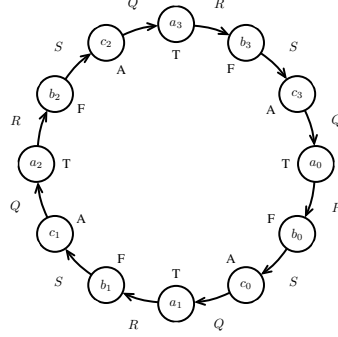
Note that the classification problem we deal with in this section can be regarded as an instance of a more general problem of classifying *datalog programs* in terms of their data complexity, in particular, finding an NL/P dichotomy.

## 7 CONP

On the other hand, a minor extension of the CQ from Example 1 can lead to CONP-completeness. First we show that answering the OMQ  $Q = (Dis, q)$  with the Boolean CQ  $q$  given in the picture below is CONP-complete.

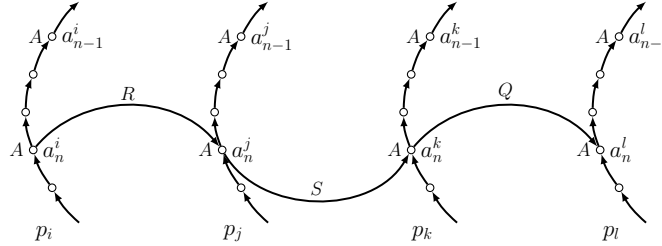


Consider the ABoxes  $\mathcal{A}_N$  constructed according to the pattern shown below for  $N = 3$ :



Let  $V = \{a_0, \dots, a_N\}$ . It is not hard to see that (i) for any interpretation  $\mathcal{I}$  based on  $\mathcal{A}_N$ , if  $\mathcal{I} \not\models \mathbf{q}$  then either  $V \subseteq T^{\mathcal{I}}$  or  $V \subseteq F^{\mathcal{I}}$ ; (ii) the interpretations  $\mathcal{I}$  and  $\mathcal{I}'$  obtained by extending  $\mathcal{A}_N$  with  $T^{\mathcal{I}'} = T^{\mathcal{A}} \cup V$  and  $F^{\mathcal{I}'} = F^{\mathcal{A}} \cup V$ , respectively, are both models of  $\mathcal{D}is$  that do not satisfy  $\mathbf{q}$ .

Given a 2+2-CNF  $\phi$  with clauses  $D_1, \dots, D_N$  and variables  $p_1, \dots, p_M$ , we take  $M$  disjoint copies of  $\mathcal{A}_N$ , distinguishing between them by the superscripts  $1, \dots, M$ . For example,  $a_3^2$  is the  $a_3$ -point of the second copy of  $\mathcal{A}_N$  and  $V^2 = \{a_0^2, \dots, a_N^2\}$ . For each  $D_n$  of the form  $\neg p_i \vee \neg p_j \vee p_k \vee p_l$ , we add to those copies the atoms  $R(a_n^i, a_n^j)$ ,  $S(a_n^j, a_n^k)$  and  $Q(a_n^k, a_n^l)$ , and denote the resulting ABox by  $\mathcal{A}_\phi$ .



We show that  $\phi$  is satisfiable iff  $\mathcal{D}is, \mathcal{A}_\phi \not\models \mathbf{q}$ . Let  $\mathbf{q}' = R(x, y) \wedge S(y, z) \wedge Q(z, w)$ . Observe that any possible match of  $\mathbf{q}'$  in  $\mathcal{A}_\phi$  falls into one of the two groups:

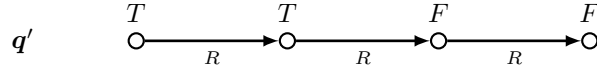
- (A)  $(a_n^i, b_n^i, c_n^i, a_{n+1}^i)$ , for  $0 \leq n \leq N$ ,  $1 \leq i \leq M$  and addition modulo  $N + 1$ ;
- (B)  $(a_n^i, a_n^j, a_n^k, a_n^l)$ , for some clause  $D_n = (\neg p_i \vee \neg p_j \vee p_k \vee p_l)$  in  $\phi$ .

Suppose  $\phi$  is satisfiable under an assignment  $\mathbf{a}$ . We define a model  $\mathcal{I}$  of  $\mathcal{D}is$  by extending  $\mathcal{A}_\phi$  with  $T^{\mathcal{I}} = T^{\mathcal{A}_\phi} \cup \bigcup \{V^i \mid \mathbf{a}(p_i) = 1\}$ ,  $F^{\mathcal{I}} = F^{\mathcal{A}_\phi} \cup \bigcup \{V^i \mid \mathbf{a}(p_i) = 0\}$ . We claim that  $\mathcal{I} \not\models \mathbf{q}$ . Indeed, the tuples in (A) cannot yield a match by (ii) above, while the tuples in (B) do not give a match since  $\mathbf{a}(D_n) = 1$ , for all  $n \leq N$ . To see this, suppose a tuple  $(a_n^i, a_n^j, a_n^k, a_n^l)$  from (B) is a match for  $\mathbf{q}$  in  $\mathcal{I}$ . Then  $\{a_n^i, a_n^j\} \subseteq T^{\mathcal{I}}$  and  $\{a_n^k, a_n^l\} \subseteq F^{\mathcal{I}}$ , from which  $\mathbf{a}(p_i) = 1$ ,  $\mathbf{a}(p_j) = 1$ ,  $\mathbf{a}(p_k) = 0$  and  $\mathbf{a}(p_l) = 0$ , and so the clause  $D_n = \neg p_i \vee \neg p_j \vee p_k \vee p_l$  is false under  $\mathbf{a}$ .

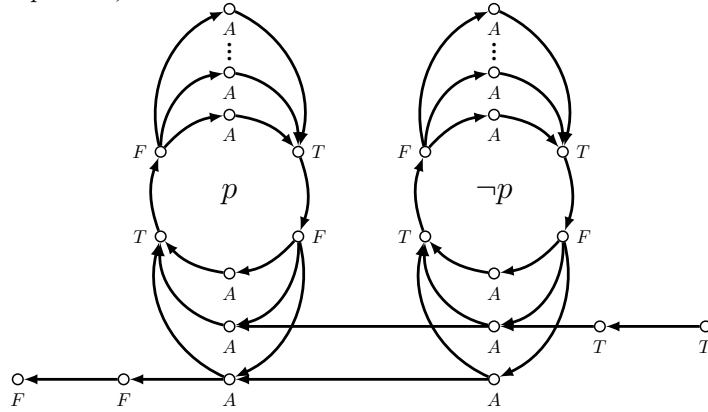
Conversely, suppose  $\mathcal{D}is, \mathcal{A}_\phi \not\models \mathbf{q}$ . Then there is a model  $\mathcal{I}$  of  $\mathcal{D}is$  based on  $\mathcal{A}_\phi$  such that  $\mathcal{I} \not\models \mathbf{q}$ . By (i) above applied to the copies of  $\mathcal{A}_N$ , for every  $i \leq M$ , we have

either  $V_i \subseteq T^{\mathcal{I}}$  or  $V_i \subseteq F^{\mathcal{I}}$ . In the former case, we set  $\mathbf{a}(p_i) = 1$ ; in the latter one, we set  $\mathbf{a}(p_i) = 0$ . We claim that  $\phi$  is satisfiable under  $\mathbf{a}$ . Indeed, if  $D_n = \neg p_i \vee \neg p_j \vee p_k \vee p_l$  is false under  $\mathbf{a}$ , then  $\mathbf{a}(p_i) = 1$ ,  $\mathbf{a}(p_j) = 1$ ,  $\mathbf{a}(p_k) = 0$  and  $\mathbf{a}(p_l) = 0$ , and so the tuple  $(a_n^i, a_n^j, a_n^k, a_n^l)$  would be a match for  $\mathbf{q}$  in  $\mathcal{I}$ .

The proposed method is generic in the sense that we can try to apply it to any ‘sufficiently asymmetric’ CQ  $\mathbf{q}$  with two  $T$ -atoms and two  $F$ -atoms: we use a  $T$ - $F$  fragment of  $\mathbf{q}$  for copying the values of the Boolean variables, and the whole  $\mathbf{q}$  for encoding the clauses of a  $2 + 2$ -CNF. However, this method does not work for the CQ

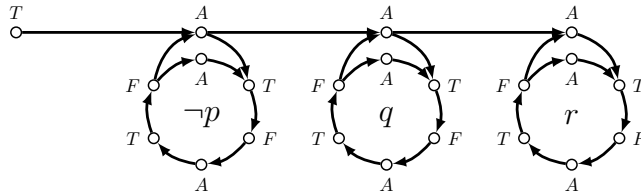


which requires a somewhat different technique. We show CONP-hardness of  $(Dis, \mathbf{q}')$  by reduction of 3SAT. Given a 3CNF  $\psi$ , we define an ABox  $\mathcal{A}_\psi$  as follows. First, for every variable  $p$  in  $\psi$ , we construct a ‘gadget’ shown in the picture below, where the number of  $A$ -nodes above each of the circles matches the number of clauses in  $\psi$ ; we refer to these nodes as  $p$ -nodes and, respectively,  $\neg p$ -nodes (below the circles, there are  $2 p$ - and  $2 \neg p$ -nodes):



Observe that, for any model  $\mathcal{I}$  of  $Dis$  and the constructed gadget for  $p$ , if  $\mathcal{I} \not\models \mathbf{q}$  then either (i) the  $p$ -nodes are all in  $F^{\mathcal{I}}$  and the  $\neg p$ -nodes are all in  $T^{\mathcal{I}}$ , or (ii) the  $p$ -nodes are all in  $T^{\mathcal{I}}$  and the  $\neg p$ -nodes are all in  $F^{\mathcal{I}}$ .

Now, for every clause  $c = (l_1 \vee l_2 \vee l_3)$  in  $\psi$ , we add to the constructed gadgets the atoms  $T(c)$ ,  $R(c, a_{-l_1}^c)$ ,  $R(a_{-l_1}^c, a_{l_2}^c)$ ,  $R(a_{l_2}^c, a_{l_3}^c)$ , where  $c$  is a new individual,  $a_{-l_1}^c$  a fresh  $\neg l_1$ -node,  $a_{l_2}^c$  a fresh  $l_2$ -node, and  $a_{l_3}^c$  a fresh  $l_3$ -node. For example, for the clause  $c = (p \vee q \vee r)$ , we obtain the fragment below. The resulting ABox is denoted by  $\mathcal{A}_\psi$ .



One can show that  $\psi$  is satisfiable iff  $Dis, \mathcal{A}_\psi \not\models \mathbf{q}'$ .

**Acknowledgements.** The work of O. Gerasimova and M. Zakharyashev was carried out at the National Research University Higher School of Economics and supported by the Russian Science Foundation under grant 17-11-01294; the work of V. Podolskii was supported by the Russian Academic Excellence Project ‘5-100’ and by grant MK-7312.2016.1. Thanks are due to Frank Wolter and Carsten Lutz for comments, suggestions and discussions.

## References

1. Arora, S., Barak, B.: Computational Complexity: A Modern Approach. Cambridge University Press, New York, NY, USA, 1st edn. (2009)
2. Artale, A., Calvanese, D., Kontchakov, R., Zakharyashev, M.: The DL-Lite family and relations. *Journal of Artificial Intelligence Research (JAIR)* 36, 1–69 (2009)
3. Bienvenu, M., ten Cate, B., Lutz, C., Wolter, F.: Ontology-based data access: A study through disjunctive datalog, csp, and MMSNP. *ACM Transactions on Database Systems* 39(4), 33:1–44 (2014)
4. Calvanese, D., De Giacomo, G., Lembo, D., Lenzerini, M., Rosati, R.: Tractable reasoning and efficient query answering in description logics: the *DL-Lite* family. *Journal of Automated Reasoning* 39(3), 385–429 (2007)
5. Feier, C., Kuusisto, A., Lutz, C.: Rewritability in monadic disjunctive datalog, MMSNP, and expressive description logics. CoRR abs/1701.02231 (2017), <http://arxiv.org/abs/1701.02231>
6. Gottlob, G., Papadimitriou, C.H.: On the complexity of single-rule datalog queries. *Inf. Comput.* 183(1), 104–122 (2003), [http://dx.doi.org/10.1016/S0890-5401\(03\)00012-9](http://dx.doi.org/10.1016/S0890-5401(03)00012-9)
7. Hernich, A., Lutz, C., Ozaki, A., Wolter, F.: Schema.org as a description logic. In: Calvanese, D., Konev, B. (eds.) *Proceedings of the 28th International Workshop on Description Logics, Athens, Greece, June 7-10, 2015. CEUR Workshop Proceedings*, vol. 1350. CEUR-WS.org (2015), <http://ceur-ws.org/Vol-1350/paper-24.pdf>
8. Kaminski, M., Nenov, Y., Grau, B.C.: Datalog rewritability of disjunctive datalog programs and non-Horn ontologies. *Artif. Intell.* 236, 90–118 (2016), <http://dx.doi.org/10.1016/j.artint.2016.03.006>
9. Papadimitriou, C.: Computational Complexity. Addison-Wesley (1994)
10. Poggi, A., Lembo, D., Calvanese, D., De Giacomo, G., Lenzerini, M., Rosati, R.: Linking data to ontologies. *Journal on Data Semantics X*, 133–173 (2008)