Proceedings of the 17th International Conference on Principles of Knowledge Representation and Reasoning (KR 2020)
Main Track

A Data Complexity and Rewritability Tetrachotomy of Ontology-Mediated Queries with a Covering Axiom

Olga Gerasimova[1] , **Stanislav Kikot**[2] , **Agi Kurucz**[3] , **Vladimir Podolskii**[4,1] ,
Michael Zakharyaschev[5]
[1]National Research University Higher School of Economics, Moscow, Russia
[2]School of Computing and Digital Media, London Metropolitan University, U.K.
[3]Department of Informatics, King's College London, U.K.
[4]Steklov Mathematical Institute, Moscow, Russia
[5]Department of Computer Science, Birkbeck, University of London, U.K.

Abstract

Aiming to understand the data complexity of answering conjunctive queries mediated by an axiom stating that a class is covered by the union of two other classes, we show that deciding their first-order rewritability is PSPACE-hard and obtain a number of sufficient conditions for membership in AC^0, L, NL, and P. Our main result is a complete syntactic AC^0/NL/P/CONP tetrachotomy of path queries under the assumption that the covering classes are disjoint.

1 Introduction

The general problem we are interested in is to determine the data complexity of answering a given ontology-mediated query (OMQ) and reduce it, if possible, to evaluating a conventional query with optimal complexity. In the context of datalog, this problem (called optimisation) has been investigated since the late 1980s. For example, it was shown that boundedness (FO-rewritability) is undecidable for linear datalog programs with binary IDB predicates (Vardi 1988) and single rule programs (Marcinkowski 1996), 2EXPTIME-complete for monadic programs (Cosmadakis et al. 1988; Benedikt et al. 2015), and PSPACE-complete for linear monadic programs (Cosmadakis et al. 1988). Considerable efforts have been made to understand linearisability of datalog programs ensuring evaluation in NL (Ullman and Gelder 1988; Ramakrishnan et al. 1989; Saraiya 1989; Zhang, Yu, and Troy 1990; Afrati, Gergatsoulis, and Toni 2003), and datalog rewritability of disjunctive datalog programs (Kaminski, Nenov, and Grau 2016).

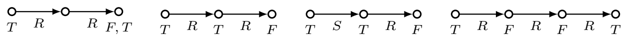
The rise of description logics (DLs), Web ontology languages, and especially the paradigm of ontology-based data access—OBDA, for short—(Poggi et al. 2008) have led to the development of formalisms that uniformly guarantee answering OMQs with the desired data complexity. Thus, OMQs with *DL-Lite* or *OWL 2 QL* ontologies and conjunctive queries (CQs) are FO-rewritable and can be answered in AC^0 (Calvanese et al. 2007), while those with *OWL 2 EL* and *horn*$\mathcal{SHIQ}$ ontologies can be answered in P (Hustadt, Motik, and Sattler 2005; Rosati 2007) and are datalog-rewritable (Eiter et al. 2012).

On the other hand, in OBDA practice, ontologies are usually designed by domain experts who are more concerned about capturing relevant knowledge than following the restrictions of this or that language. For instance, the NPD FactPages ontology, used for testing OBDA in industry (Hovland et al. 2017; Kharlamov et al. 2017), contains *covering axioms* of the form $A \sqsubseteq B_1 \sqcup \cdots \sqcup B_n$, which are not allowed in *DL-Lite* as there exist CONP-hard OMQs with covering. One can show, however, that the concrete queries provided by the end-users in (Hovland et al. 2017) do not 'feel' those dangerous axioms and are FO-rewritable.

The problem of determining the *non-uniform* data complexity and rewritability of OMQs was attacked by Lutz and Wolter (2012) for individual DL ontologies with varying CQs and Bienvenu et al. (2014) for individual OMQs. In particular, the latter found a connection of OMQs to non-uniform CSPs and used it to show that deciding FO- and datalog-rewritability of OMQs with a $\mathcal{SHIU}$ ontology and an atomic query is NEXPTIME-complete. The Feder-Vardi dichotomy of CSPs (Bulatov 2017; Zhuk 2017) implies a P/CONP dichotomy of such OMQs, which is decidable in NEXPTIME. An AC^0/NL/P trichotomy of $\mathcal{EL}$ OMQs, deciding which is EXPTIME-complete, was established by Lutz and Sabellek (2017).

All in all, the general problem of optimising datalog programs or rewriting DL OMQs turns out to be computationally very hard. Moreover, in spite of multiple attempts, very few practically useful partial algorithms or easily checkable syntactic conditions have been discovered so far.

One possible way forward, at least in the DL setting, is to understand the impact of typical ontology axioms, which are used in conceptual data modelling, on the complexity of answering OMQs. Motivated by the use case with the NPD FactPages ontology and the attempt of Hernich et al. (2015) to classify OMQs with Schema.org ontologies, which feature disjunctive concept inclusions, in this paper we start investigating OMQs whose ontology consists of a single 'covering' axiom $A \sqsubseteq F \sqcup T$ with concept names (unary predicates) A, F, T and Boolean CQs with unary and binary predicates. To illustrate, the axiom could mean that everyone in the U.K. (A) is either a Brexiteer (F) or a Remainer (T), though people prefer not to publicise their views, and so it would be interesting to investigate, say, Facebook by running queries like 'is there a Brexiteer with a Remainer spouse who follows a Brexiteer?' Seemingly trivial, such OMQs exhibit quite complex computational behaviour: for example, the OMQs with this covering axiom and the very primitive path CQs shown in the picture below are, respectively, in AC^0 and NL-, P- and CONP-complete.



Curiously, the Polyanna program (Gault and Jeavons 2004), designed to check tractability of CSPs, failed to recognise the intractability of the fourth OMQ as the reduction to CSP is unavoidably exponential in general. Thus, classifying OMQs with the covering axiom and CQs having two unary predicates F and T appears to play a fundamental role in the non-uniform approach to OBDA with expressive ontologies.

Our contribution. In this paper, we obtain a series of results on the complexity of answering Boolean OMQs of the form

$$\boldsymbol{Q} = (\mathsf{cov}_A, \boldsymbol{q}) \qquad \text{with} \qquad \mathsf{cov}_A = \{A \sqsubseteq F \sqcup T\} \qquad (1)$$

and a CQ $\boldsymbol{q}$ and their rewritability. On the 'negative' side, we show that, despite the language of our OMQs is reduced to the bare bones, in the presence of covering, CQs can encode $\forall\exists$3SAT and capture some aspects of the acyclicity problem for succinct graphs. More precisely, **(a)** we show that, in general, answering OMQs (1) is Π_2^p-complete for combined complexity (in the size of $\boldsymbol{q}$ and the data), that is, harder than answering *DL-Lite* and $\mathcal{EL}$ OMQs (unless $\mathrm{NP} = \Pi_2^p$, and so $\mathrm{NP} = \mathrm{PSPACE}$); **(b)** we prove that the problem of determining FO-rewritability of these OMQs is even harder, namely, PSPACE-hard in the size of $\boldsymbol{q}$, which indicates that a general syntactic classification of CQs $\boldsymbol{q}$ according to the data complexity of answering $\boldsymbol{Q}$ and the type of its rewritability will be extremely difficult to find. This result is quite surprising in comparison with the PSPACE-hardness proofs for boundedness of linear monadic datalog programs (Cosmadakis et al. 1988) and FO-rewritability of OMQs with a Schema.org ontology and a union of CQs (Hernich et al. 2015), where different rules in a datalog program or different CQs in a UCQ were used to ensure correctness of a Turing machine computation. Here, we encode the acyclicity of a graph, succinctly represented by a Boolean formula, using just a single dag-shaped CQ.

These negative results might appear to suggest that even our primitive OMQs are too 'sophisticated' for a fine complexity analysis. However, we also obtain substantial and encouraging positive results: **(c)** First, we show a number of general syntactic and semantic partial conditions for various types of rewritability and data complexity that are applicable to arbitrary CQs. We begin by observing that a CQ without FT-twins (that is, without both $F(x)$ and $T(x)$, for any x, unlike in the first CQ depicted above) gives rise to an FO-rewritable OMQ (i.e., in AC^0) if it does not contain occurrences of one of F or T; otherwise the OMQ is L-hard and even NL-hard for a path CQ. This simple criterion fails for CQs with FT-twins, where the problem of finding a syntactic characterisation turns out to be extremely difficult. The OMQs with a CQ containing a single solitary F (or T) are shown to be datalog-rewritable (and so in P). As far as we are aware, there is no known semantic or syntactic criterion distinguishing between datalog programs in NL and P, though Lutz and Sabellek (Lutz and Sabellek 2017) gave a nice criterion for OMQs with an $\mathcal{EL}$ ontology. We combine their ideas with the automata-theoretic technique of Cosmadakis et al. (Cosmadakis et al. 1988), and prove a useful sufficient semantic condition for our OMQs to be linear-datalog-rewritable (and so in NL). **(d)** We use some of these conditions to obtain the main result of this paper: a complete and transparent syntactic AC^0/NL/P/CONP and rewritability tetrachotomy of the OMQs with 'FT-twinless' path CQs. The 'FT-twinless' restriction is redundant if the ontology is extended with the disjointness axiom $F \sqcap T \sqsubseteq \bot$. We show that (i) such CQs $\boldsymbol{q}$ without occurrences of F (or T) and only them give rise to FO-rewritable OMQs $\boldsymbol{Q}$ (in AC^0), that (ii) $\boldsymbol{Q}$ is linear-datalog-rewritable and NL-complete just in case $\boldsymbol{q}$ has a certain periodic structure, and prove that (iii) otherwise $\boldsymbol{Q}$ can simulate monotone circuit evaluation, and so is P-hard. Finally, (iv) the most surprising and technically difficult part of our tetrachotomy is the construction showing that path CQs with at least two Fs and at least two Ts, and only them give rise to CONP-hard OMQs.

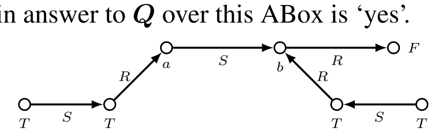
The omitted proofs can be found in the full version of this paper (Gerasimova et al. 2020).

2 Preliminaries

Using the standard description logic syntax and semantics, we consider *ontology-mediated queries* (OMQs) of the form $\boldsymbol{Q} = (\mathcal{T}, \boldsymbol{q})$, where $\mathcal{T}$ is one of the two ontologies

$$\mathsf{cov}_A = \{A \sqsubseteq F \sqcup T\}, \ \ \mathsf{cov}_A^{\bot} = \{A \sqsubseteq F \sqcup T,\ F \sqcap T \sqsubseteq \bot\}$$

(sometimes we set $A = \top$) and $\boldsymbol{q}$ a *Boolean conjunctive query* (CQ), i.e., an FO-sentence $\boldsymbol{q} = \exists \boldsymbol{x}\, \varphi(\boldsymbol{x})$, in which φ is a conjunction of atoms with variables from $\boldsymbol{x}$. We often think of $\boldsymbol{q}$ as a set of its atoms. In the context of this paper, CQs may only contain two unary predicates F, T and arbitrary binary predicates. Atoms $F(x), T(x) \in \boldsymbol{q}$ are referred to as *FT-twins* in $\boldsymbol{q}$. An *ABox*, $\mathcal{A}$, is a finite set of ground atoms; $\mathsf{ind}(\mathcal{A})$ is the set of constants (individuals) in $\mathcal{A}$. The *certain answer* to $\boldsymbol{Q}$ over $\mathcal{A}$ is 'yes' if $\mathcal{I} \models \boldsymbol{q}$ for all models $\mathcal{I}$ of $\mathcal{T}$ and $\mathcal{A}$, in which case we write $\mathcal{T}, \mathcal{A} \models \boldsymbol{q}$, and 'no' otherwise. To illustrate, consider the OMQ $\boldsymbol{Q} = (\mathsf{cov}_\top, \boldsymbol{q})$ with the third path CQ $\boldsymbol{q}$ in the picture above. By analysing the four possible cases for $a, b \in F^{\mathcal{I}}, T^{\mathcal{I}}$ in an arbitrary model $\mathcal{I}$ of $\mathsf{cov}_\top$ and the ABox below, it is readily seen that the certain answer to $\boldsymbol{Q}$ over this ABox is 'yes'.



Our concern is the *combined* and *data complexity* of deciding whether $\mathcal{T}, \mathcal{A} \models \boldsymbol{q}$. In the former case, $\mathcal{T}$, $\boldsymbol{q}$ and $\mathcal{A}$ are regarded as input; in the latter, $\mathcal{T}$ and $\boldsymbol{q}$ are fixed. Clearly, Π_2^p is an obvious upper bound for the combined and CONP for the data complexity of our problem.

An OMQ $\boldsymbol{Q} = (\mathcal{T}, \boldsymbol{q})$ is *FO-rewritable* if there is an FO-sentence Φ such that $\mathcal{T}, \mathcal{A} \models \boldsymbol{q}$ iff Φ is true in the structure $\mathcal{A}$; in terms of circuit complexity, FO-rewritability is equivalent to answering $\boldsymbol{Q}$ in logtime-uniform AC^0 (Immerman 1999). Note that if $\boldsymbol{q}$ contains FT-twins, then $\exists x\,\big(F(x) \land T(x)\big)$ is an FO-rerwriting of $(\mathsf{cov}_A^{\perp}, \boldsymbol{q})$.

We often regard CQs and ABoxes as digraphs with labelled edges and partially labelled nodes (by F, T in CQs and F, T, A in ABoxes). Without loss of generality, we assume that these graphs are *connected* as undirected graphs. A *path CQ* is a (simple) directed path each of whose edges is labelled by a single binary predicate. A *minimal model* of $\mathcal{T}$ and $\mathcal{A}$ is obtained from $\mathcal{A}$ by adding to each 'undecided' A-node (which is labelled by neither F nor T) exactly one of F or T as label. Clearly, $\mathcal{T}, \mathcal{A} \models \boldsymbol{q}$ iff $\mathcal{I} \models \boldsymbol{q}$ for every minimal model $\mathcal{I}$ of $\mathcal{T}$ and $\mathcal{A}$. So, from now on 'model' means 'minimal model'. Finally, note that $\mathcal{I} \models \boldsymbol{q}$ iff there is a digraph homomorphism $h\colon \boldsymbol{q} \to \mathcal{I}$ preserving the labels of nodes and edges.

A *datalog program*, Π, is a finite set of *rules* of the form $\forall \boldsymbol{x}\,(\gamma_0 \leftarrow \gamma_1 \land \dots \land \gamma_m)$, where each γ_i is a (constant- and function-free) atom $Q(\boldsymbol{y})$ with $\boldsymbol{y} \subseteq \boldsymbol{x}$. (As usual, we omit $\forall \boldsymbol{x}$.) The atom γ_0 is the *head* of the rule, and $\gamma_1, \dots, \gamma_m$ its *body*. All of the variables in the head must occur in the body. The predicates in the head of rules are *IDB predicates*, the rest *EDB predicates*. A *datalog query* in this paper takes the form $(\Pi, \boldsymbol{G})$ with a 0-ary goal atom $\boldsymbol{G}$. The *answer* to $(\Pi, \boldsymbol{G})$ over an ABox $\mathcal{A}$ is 'yes' if $\boldsymbol{G}$ is true in the structure $\Pi(\mathcal{A})$ obtained by closing $\mathcal{A}$ under Π, in which case we write $\Pi, \mathcal{A} \models \boldsymbol{G}$. We call $(\Pi, \boldsymbol{G})$ a *datalog-rewriting* of an OMQ $\boldsymbol{Q} = (\mathcal{T}, \boldsymbol{q})$ in case $\mathcal{T}, \mathcal{A} \models \boldsymbol{q}$ iff $\Pi, \mathcal{A} \models \boldsymbol{G}$ for any ABox $\mathcal{A}$. If $\boldsymbol{Q}$ is datalog-rewritable, then it can be answered in P for data complexity; if there is a rewriting to a $(\Pi, \boldsymbol{G})$ with a *linear* program Π, having at most one IDB predicate in the body of each of its rules, then $\boldsymbol{Q}$ can be answered in NL (nondeterministic logarithmic space).

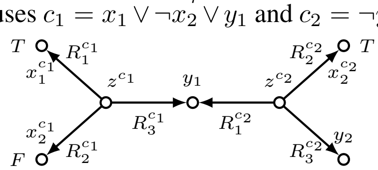
3 General Results

In this section, we obtain a number of complexity and rewritability results that are applicable to arbitrary CQs $\boldsymbol{q}$. By writing $\boldsymbol{Q} = (\mathcal{T}, \boldsymbol{q})$ we mean 'any $\mathcal{T} \in \{\mathsf{cov}_A, \mathsf{cov}_A^{\perp}\}$'.

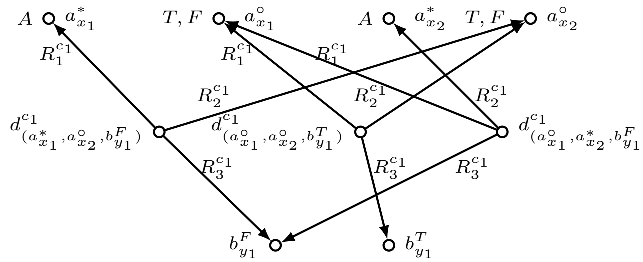
Our first result pushes to the limit (Hernich et al. 2015, Theorem 5) according to which answering OMQs with Schema.org ontologies is Π_2^p-complete for combined complexity (their proof of Π_2^p-hardness used an ontology with an enumeration definition $E = \{0, 1\}$ and additional concept names, none of which is available in our case).

Theorem 1. *Answering OMQs $(\mathcal{T}, \boldsymbol{q})$ is Π_2^p-complete for combined complexity.*

Proof. The proof is by reduction of Π_2^p-complete $\forall\exists$3SAT (Stockmeyer 1976). Given $\varphi = \forall \boldsymbol{x} \exists \boldsymbol{y} \psi(\boldsymbol{x}, \boldsymbol{y})$ with a 3CNF $\psi(\boldsymbol{x}, \boldsymbol{y})$, we construct a CQ $\boldsymbol{q}_\varphi$ shown below for ψ comprising two clauses $c_1 = x_1 \lor \neg x_2 \lor y_1$ and $c_2 = \neg y_1 \lor x_2 \lor y_2$:



For cov_A, the ABox $\mathcal{A}_\varphi$ is defined as follows. For $x \in \boldsymbol{x}$, we take individuals a_x^*, a_x° and, for $y \in \boldsymbol{y}$, individuals b_y^F, b_y^T. $\mathcal{A}_\varphi$ comprises the atoms $A(a_x^*)$, $F(a_x^\circ)$, $T(a_x^\circ)$, for $x \in \boldsymbol{x}$. For each $c = \ell_1 \lor \ell_2 \lor \ell_3$, we define a set E^c of triples of the above individuals: $(e_1, e_2, e_3) \in E^c$ iff (i) $e_i = a_x^\mu$ for some $\mu \in \{*, \circ\}$ whenever $x \in \boldsymbol{x}$ is in ℓ_i, (ii) $e_i = b_y^\nu$ for some $\nu \in \{F, T\}$ whenever $y \in \boldsymbol{y}$ is in ℓ_i, and (iii) either $e_i = a_x^*$ for some i, or $e_i = b_y^\nu$ for some i and the assignment $y = \nu$ makes ℓ_i true. For any c and $t = (e_1, e_2, e_3)$ in E^c, we take a fresh individual d_t^c and add $R_i^c(d_t^c, e_i)$, $i = 1, 2, 3$, to $\mathcal{A}_\varphi$.



One can show that φ is satisfiable iff $\mathsf{cov}_A, \mathcal{A}_\varphi \models \boldsymbol{q}_\varphi$. For $\mathsf{cov}_A^{\perp}$, the construction is slightly more involved. ❑

From now on, we focus on the *data* complexity of (answering) OMQs $(\mathcal{T}, \boldsymbol{q})$. If $\boldsymbol{q}$ does not contain FT-twins, we call it *twinless*. By a *solitary* F (or T) we mean a non-twin F-node (respectively, T-node). Finally, we call $\boldsymbol{q}$ a 0-*CQ* if it does not have a solitary F or a solitary T (but it might contain twins). Note that, for any twinless $\boldsymbol{q}$, $(\mathsf{cov}_A, \boldsymbol{q})$ and $(\mathsf{cov}_A^{\perp}, \boldsymbol{q})$ have the same data complexity.

Theorem 2. (i) *If $\boldsymbol{q}$ is a 0-CQ, then $(\mathcal{T}, \boldsymbol{q})$ is in* AC^0.

(ii) *If $\boldsymbol{q}$ is twinless and contains at least one solitary F and at least one solitary T, then $(\mathsf{cov}_\top, \boldsymbol{q})$ and $(\mathsf{cov}_\top^{\perp}, \boldsymbol{q})$, and so $(\mathsf{cov}_A, \boldsymbol{q})$ and $(\mathsf{cov}_A^{\perp}, \boldsymbol{q})$ are* L*-hard.*

Proof. (i) We show that $\mathcal{T}, \mathcal{A} \models \boldsymbol{q}$ iff $\mathcal{A} \models \boldsymbol{q}$, and so $\boldsymbol{q}$ is an FO-rewriting of $(\mathcal{T}, \boldsymbol{q})$. $(\Rightarrow)$ Suppose $\mathcal{A} \not\models \boldsymbol{q}$ and $\boldsymbol{q}$ has no solitary F. Let $\mathcal{A}'$ be the result of adding a label F to every undecided A-node in $\mathcal{A}$. Clearly, $\mathcal{A}'$ is a model of $\mathcal{T}$ and $\mathcal{A}$ with $\mathcal{A}' \not\models \boldsymbol{q}$. $(\Leftarrow)$ is trivial.

(ii) The proof is similar to that of Theorem 13, using an FO-reduction of the L-complete reachability problem for undirected graphs. ❑

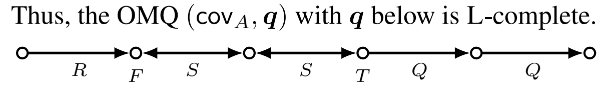
Theorem 2 (ii) is complemented by the following simple sufficient condition, which can be proved using symmetric datalog (which is in L for data complexity; see, e.g., (Egri, Larose, and Tesson 2007)). Call a CQ $\boldsymbol{q}'(x, y)$ with two free variables x and y *symmetric* if, for any ABox $\mathcal{A}$ and $a, b \in \mathsf{ind}(\mathcal{A})$, we have $\mathcal{A} \models \boldsymbol{q}'(a, b)$ iff $\mathcal{A} \models \boldsymbol{q}'(b, a)$.

Theorem 3. *Let $\boldsymbol{Q} = (\mathcal{T}, \boldsymbol{q})$ be any OMQ with*

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

for some CQs $\boldsymbol{q}'(x, y)$, $\boldsymbol{q}_1'(x)$ and $\boldsymbol{q}_2'(y)$ that do not contain solitary T and F, and symmetric $\boldsymbol{q}'(x, y)$, assuming $\boldsymbol{q}_1'(x)$ and $\boldsymbol{q}_2'(y)$ to be disjoint and have x and y as their only common variables with $\boldsymbol{q}'(x, y)$. Then $\boldsymbol{Q}$ is in L.

Thus, the OMQ $(\mathsf{cov}_A, \boldsymbol{q})$ with $\boldsymbol{q}$ below is L-complete.



Since $\mathrm{AC}^0 \subsetneq \mathrm{L}$, Theorem 2 gives a necessary and sufficient criterion in the presence of the disjointness axiom:

Corollary 4. *An OMQ* $(\mathsf{cov}_A^\perp, \boldsymbol{q})$ *is in* AC^0 *iff* $\boldsymbol{q}$ *is a* 0*-CQ or contains a twin. If* $\boldsymbol{q}$ *is a twinless* 0*-CQ, then* $\boldsymbol{q}$ *is an FO-rewriting of* $(\mathsf{cov}_A^\perp, \boldsymbol{q})$.

Proof. We only show the latter. Suppose $\boldsymbol{q}$ is F-free. As $(\mathsf{cov}_A^\perp, \boldsymbol{q})$ is FO-rewritable, it has a UCQ rewriting (Bienvenu et al. 2014, Proposition 5.9). Take any CQ $\boldsymbol{q}'$ in this UCQ and regard it as an ABox, $\mathcal{A}$. Clearly, $\mathsf{cov}_A^\perp, \mathcal{A} \models \boldsymbol{q}$. Let $\mathcal{A}'$ result from $\mathcal{A}$ by adding F to any undecided A-node. Then there is a homomorphism $h\colon \boldsymbol{q} \to \mathcal{A}'$. As $\boldsymbol{q}$ is F-free, h is also a homomorphism to $\boldsymbol{q}'$, and so $\boldsymbol{q}$ is an FO-rewriting of $(\mathsf{cov}_A^\perp, \boldsymbol{q})$. ❑

Example 8 below shows that Corollary 4 does not hold for CQs with twins. We next consider OMQs in the class P.

By a 1-*CQ* we mean any CQ with exactly one solitary F and at least one solitary T, or exactly one solitary T and at least one solitary F; cf. markable disjunctive datalog programs of (Kaminski, Nenov, and Grau 2016).

Theorem 5. *Any OMQ* $\boldsymbol{Q} = (\mathcal{T}, \boldsymbol{q})$ *with a* 1*-CQ* $\boldsymbol{q}$ *is datalog-rewritable, and so is in* P.

Proof. Suppose that $F(x)$ and $T(y_1), \dots, T(y_n)$ are all the solitary occurrences of F and T in $\boldsymbol{q}$. Let $\Pi_{\boldsymbol{q}}$ be a monadic datalog program with goal $\boldsymbol{G}$ and four rules

$$\boldsymbol{G} \leftarrow F(x), \boldsymbol{q}', P(y_1), \dots, P(y_n) \qquad (2)$$

$$P(x) \leftarrow T(x) \qquad (3)$$

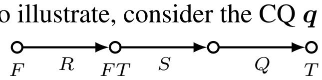
$$P(x) \leftarrow A(x), \boldsymbol{q}', P(y_1), \dots, P(y_n) \qquad (4)$$

$$\boldsymbol{G} \leftarrow F(x), T(x) \qquad (5)$$

where $\boldsymbol{q}' = \boldsymbol{q} \setminus \{F(x), T(y_1), \dots, T(y_n)\}$ and P is fresh. (If $\mathcal{T} = \mathsf{cov}_A$, rule (5) is omitted.) Then, for any ABox $\mathcal{A}$ (without P), we have $\mathcal{T}, \mathcal{A} \models \boldsymbol{q}$ iff $\Pi_{\boldsymbol{q}}, \mathcal{A} \models \boldsymbol{G}$. ❑

Theorem 5 makes it possible to use the 2ExpTime algorithm of (Cosmadakis et al. 1988) to decide whether $\Pi_{\boldsymbol{q}}$ is bounded, and so $\boldsymbol{Q}$ is in AC^0, and the results of (Ullman and Gelder 1988; Ramakrishnan et al. 1989; Afrati, Gergatsoulis, and Toni 2003) and many other techniques to understand whether $\Pi_{\boldsymbol{q}}$ can be transformed to a linear program, which would mean that $\boldsymbol{Q}$ is in NL. For OMQs $\boldsymbol{Q}$ whose 1-CQ $\boldsymbol{q}$ is a ditree with its unique solitary F-node as root, the program $\Pi_{\boldsymbol{q}}$ can be reformulated as an $\mathcal{EL}$ ontology, and so one can use the AC^0/NL/P trichotomy of (Lutz and Sabellek 2017), which is checkable in ExpTime.

Example 6. To illustrate, consider the CQ $\boldsymbol{q}$ below.



We have $\mathsf{cov}_A, \mathcal{A} \models \boldsymbol{q}$ iff $\mathcal{E}, \mathcal{A} \models \exists x\, B(x)$, where $\mathcal{E}$ is the $\mathcal{EL}$ TBox $\{F \sqcap C_{\boldsymbol{q}} \sqsubseteq B,\, T \sqsubseteq P,\, A \sqcap C_{\boldsymbol{q}} \sqsubseteq P\}$ with $C_{\boldsymbol{q}} = \exists R.(F \sqcap T \sqcap \exists S.\exists Q.P)$.

We now use some techniques from (Cosmadakis et al. 1988; Lutz and Sabellek 2017) to obtain handy semantic conditions for an OMQ to be in AC^0 or NL. Let $\boldsymbol{Q} = (\mathcal{T}, \boldsymbol{q})$ be an OMQ with a 1-CQ $\boldsymbol{q}$ and a single solitary $F(x)$. Define by induction a class $\mathfrak{K}_{\boldsymbol{Q}}$ of ABoxes called *cactuses for* $\boldsymbol{Q}$. We start by setting $\mathfrak{K}_{\boldsymbol{Q}} = \{\boldsymbol{q}\}$, regarding $\boldsymbol{q}$ as an ABox, and then recursively apply to $\mathfrak{K}_{\boldsymbol{Q}}$ the following two rules:

(bud) if $T(y) \in \mathcal{C} \in \mathfrak{K}_{\boldsymbol{Q}}$ with solitary $T(y)$, then we add to $\mathfrak{K}_{\boldsymbol{Q}}$ the ABox obtained by replacing $T(y)$ in $\mathcal{C}$ with $(\boldsymbol{q} \setminus \{F(x)\}) \cup \{A(x)\}$, in which x is renamed to y and all other variables are given *fresh* names;

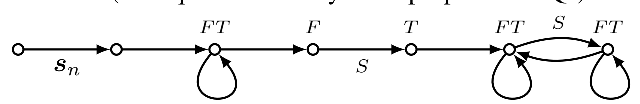
(prune) if $\mathcal{C} \in \mathfrak{K}_{\boldsymbol{Q}}$ and $\mathcal{T}, \mathcal{C}^- \models \boldsymbol{q}$, where $\mathcal{C}^- = \mathcal{C} \setminus \{T(y)\}$ for some solitary $T(y)$ in $\mathcal{C}$, then we add $\mathcal{C}^-$ to $\mathfrak{K}_{\boldsymbol{Q}}$.

We call a cactus *unpruned* if it can be obtained by applications of **(bud)** only. For $\mathcal{C} \in \mathfrak{K}_{\boldsymbol{Q}}$, we refer to the copies of (maximal subsets of) $\boldsymbol{q}$ that comprise $\mathcal{C}$ as *segments*. The *skeleton* $\mathcal{C}^s$ of $\mathcal{C}$ is the ditree whose nodes are the segments $\mathfrak{s}$ of $\mathcal{C}$ and edges $(\mathfrak{s}, \mathfrak{s}')$ mean that $\mathfrak{s}'$ was attached to $\mathfrak{s}$ by budding. The *depth of* $\mathcal{C}$ is the number of edges on the longest branch of $\mathcal{C}^s$. A *path-cactus* is a cactus whose skeleton has a single branch.

It is straightforward to see by structural induction that $\mathcal{T}, \mathcal{C} \models \boldsymbol{q}$, for all $\mathcal{C} \in \mathfrak{K}_{\boldsymbol{Q}}$. Further, for any ABox $\mathcal{A}$, we have $\mathcal{T}, \mathcal{A} \models \boldsymbol{q}$ iff either $\mathcal{T} = \mathsf{cov}_A^\perp$ and $\mathcal{A}$ contains an FT-twin, or there exists a homomorphism $h\colon \mathcal{C} \to \mathcal{A}$ for some unpruned $\mathcal{C} \in \mathfrak{K}_{\boldsymbol{Q}}$. Using these observations and (Bienvenu et al. 2014, Proposition 5.9), we obtain:

Theorem 7. *An OMQ* $\boldsymbol{Q} = (\mathcal{T}, \boldsymbol{q})$ *with a 1-CQ* $\boldsymbol{q}$ *is FO-rewritable iff there exists* $d < \omega$ *such that every* $\mathcal{C} \in \mathfrak{K}_{\boldsymbol{Q}}$ *contains a homomorphic image of some unpruned* $\mathcal{C}^- \in \mathfrak{K}_{\boldsymbol{Q}}$ *of depth* $\leq d$.

Example 8. Let $\boldsymbol{s}_n$ be a chain of n arrows labelled by S, for $n \geq 3$. Consider the CQ $\boldsymbol{q}_n$ shown below, where the omitted labels on edges are all R. It is not hard to check that $\boldsymbol{q}_n$ is minimal (not equivalent to any of its proper sub-CQs).



Let $\mathcal{C}_k$ be the cactus obtained by applying **(bud)** k-times to $\mathcal{C}_0 = \boldsymbol{q}_3$. Then there is a homomorphism $\boldsymbol{q}_3 \to \mathcal{C}_k$, for any $k \geq 2$: it uses the S-chain before the T-node to accommodate $\boldsymbol{s}_3$. However, there is no homomorphism from $\boldsymbol{q}_3$ to $\mathcal{C}_1$ as $\boldsymbol{s}_3$ is too long. It follows that $\boldsymbol{q}_3 \vee \mathcal{C}_1$ is an FO-rewriting of $(\mathsf{cov}_A, \boldsymbol{q}_3)$, where we treat $\mathcal{C}_1$ as a CQ. It is to be noted that $\mathcal{C}_1$ has an A-node. In general, the UCQ $\boldsymbol{q} \vee \mathcal{C}_1 \vee \dots \vee \mathcal{C}_{n-2}$ is an FO-rewriting of $(\mathsf{cov}_A, \boldsymbol{q}_n)$.

We use Theorem 7 to prove that checking FO-rewritability of OMQs with 1-CQs is PSpace-hard. This result should be compared to (Hernich et al. 2015, Theorem 11) showing PSpace-hardness of FO-rewritability of *UCQs* mediated by Schema.org ontologies. In our case, the expressive power of UCQs (used to capture TM-computations) is not available, and so we had to develop a brand new technique. It is to be noted that the proof of the following theorem constructs FO-rewritable OMQs that require cactuses $\mathcal{C}_i$ in Theorem 7 of *doubly exponential* size.

Theorem 9. *Checking FO-rewritability of OMQs with* (*dag*) *1-CQs is* PSpace*-hard.*

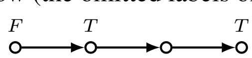
Proof. The proof is by reduction of the PSPACE-complete *acyclicity problem for succinct graphs* given by Boolean formulas (Papadimitriou and Yannakakis 1986). For each such formula $\varphi(\boldsymbol{x})$, $|\boldsymbol{x}| = n$, we construct a (dag) 1-CQ $\boldsymbol{q}_\varphi$ with one FT-twin, one solitary F- and two solitary T-nodes such that each skeleton $\mathcal{C}^s$ is of branching ≤ 2, where $\mathcal{C} \in \mathfrak{K}_{\boldsymbol{Q}}$ for $\boldsymbol{Q} = (\mathsf{cov}_A, \boldsymbol{q}_\varphi)$. Then every path-cactus $\mathcal{C}$ of depth n is encoded by a word $\mathfrak{a}_\mathcal{C} \in \{F, T\}^n$ indicating which of the T-nodes in $\boldsymbol{q}_\varphi$ was budded at each step. Using that such a word can be considered as an assignment to the variables $\boldsymbol{x}$ in φ, we then prove that the input graph given by φ has a cycle iff the set $\mathfrak{K}_{\boldsymbol{Q}}$ contains arbitrarily large cactuses without homomorphic images of unpruned cactuses in any given finite set. ❑

We next obtain a sufficient condition of linear-datalog-rewritability of $\boldsymbol{Q} = (\mathcal{T}, \boldsymbol{q})$ with 1-CQ $\boldsymbol{q}$. The (*branching*) *rank* $\boldsymbol{br}(\mathfrak{s})$ of a segment $\mathfrak{s}$ in a cactus $\mathcal{C}$ is defined inductively by taking $\boldsymbol{br}(\mathfrak{s}) = 0$ if $\mathfrak{s}$ is a leaf and, for non-leaf $\mathfrak{s}$,

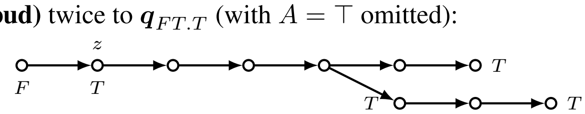
$$\boldsymbol{br}(\mathfrak{s}) = \begin{cases} m+1, & \text{if } \mathfrak{s} \text{ has} \geq 2 \text{ children of rank } m; \\ m, & \text{otherwise.} \end{cases}$$

The *branching number of* $\mathcal{C}$ is the rank of its root segment (see (Lutz and Sabellek 2017)). Let $\mathfrak{K}_{\boldsymbol{Q}}^{min}$ be the set of *minimal* cactuses in $\mathfrak{K}_{\boldsymbol{Q}}$ (to which **(prune)** is not applicable). We say that $\mathfrak{K}_{\boldsymbol{Q}}^{min}$ is *boundedly branching* if there is some $\mathbf{b} < \omega$ such that $\mathfrak{K}_{\boldsymbol{Q}}^{min}$ contains a cactus with branching number $\mathbf{b}$ but no cactus of greater branching number. Otherwise, we call $\mathfrak{K}_{\boldsymbol{Q}}^{min}$ *unboundedly branching*.

Example 10. Consider the OMQ $\boldsymbol{Q} = (\mathsf{cov}_\top, \boldsymbol{q}_{FT.T})$ with $\boldsymbol{q}_{FT.T}$ shown below (the omitted labels on edges are all R):



In the next picture, we show a cactus $\mathcal{C}$ obtained by applying **(bud)** twice to $\boldsymbol{q}_{FT.T}$ (with $A = \top$ omitted):



Clearly, $\mathsf{cov}_\top, \mathcal{C} \setminus \{T(z)\} \models \boldsymbol{q}_{FT.T}$, and so **(prune)** would remove $T(z)$ from $\mathcal{C}$. Using this fact, one can show that every cactus in $\mathfrak{K}_{\boldsymbol{Q}}^{min}$ has branching number ≤ 1. On the other hand, if $\boldsymbol{Q} = (\mathsf{cov}_A, \boldsymbol{q}_{FT.T})$, then $\mathfrak{K}_{\boldsymbol{Q}}^{min}$ is unboundedly branching by Theorems 11 and 15.

Theorem 11. *For any OMQ* $\boldsymbol{Q} = (\mathcal{T}, \boldsymbol{q})$ *with a* 1*-CQ* $\boldsymbol{q}$*, if* $\mathfrak{K}_{\boldsymbol{Q}}^{min}$ *is boundedly branching, then* $\boldsymbol{Q}$ *is linear-datalog-rewritable, and so is in* NL.

Proof. In a nutshell, the (rather involved) proof is as follows. Similarly to (Cosmadakis et al. 1988), we represent cactus-like ABoxes as terms of a tree alphabet and construct a tree automaton $\mathfrak{A}_{\boldsymbol{Q}}$ with $\mathfrak{K}_{\boldsymbol{Q}}^{min} \subseteq L(\mathfrak{A}_{\boldsymbol{Q}}) \subseteq \{\mathcal{A} \mid \mathcal{T}, \mathcal{A} \models \boldsymbol{q}\}$. Then, using ideas of (Lutz and Sabellek 2017), we show that if $\mathfrak{K}_{\boldsymbol{Q}}^{min}$ is boundedly branching, then the automaton $\mathfrak{A}_{\boldsymbol{Q}}$ can be transformed into a (monadic) linear-stratified datalog rewriting of $\boldsymbol{Q}$. As shown by (Afrati, Gergatsoulis, and Toni 2003), such a rewriting can further be converted into a linear datalog rewriting (at the expense of increasing the arity of IDB predicates in the program). ❑

4 AC^0/NL/P/coNP-Tetrachotomy

In this section, we focus on the OMQs $(\mathsf{cov}_A, \boldsymbol{q})$ with a twinless path CQ $\boldsymbol{q}$. So from now on, solitary F-nodes (T-nodes) in $\boldsymbol{q}$ will simply be called F*-nodes* (T*-nodes*). Our aim is to obtain a complete syntactic classification of these OMQs according to their data complexity and rewritability.

We begin by dividing the CQs in question into three disjoint classes: the 0-CQs and the 1-CQs, which have been defined earlier, and the 2*-CQs* that contain at least two F-nodes and at least two T-nodes. We split 1-CQs into two further classes that can be defined by an easily checkable syntactic condition as follows. We denote the first (root) node in $\boldsymbol{q}$ by s and the last (leaf) node by e. We write $x \preceq y$ to say that there is a path from x to y in $\boldsymbol{q}$, and $x \prec y$ whenever $x \preceq y$ and $x \neq y$. If $x \preceq y$, then $[x, y]$ comprises those atoms in $\boldsymbol{q}$ whose variables are in the interval $\{z \mid x \preceq z \preceq y\}$; further, $(x, y] = [x, y] \setminus \{T(x), F(x)\}$, $[x, y) = [x, y] \setminus \{T(y), F(y)\}$ and $(x, y) = [x, y) \setminus \{T(x), F(x)\}$.

Now let $x_{-l} \prec \cdots \prec x_{-1} \prec x_0 \prec x_1 \prec \cdots \prec x_r$ be all the F- or T-nodes in $\boldsymbol{q}$, with x_0 being the only F-node and $l + r \geq 1$. We denote this 1-CQ by $\boldsymbol{q}_{lr}$. Let $\boldsymbol{r}_i = (x_{i-1}, x_i)$, where $x_{-l-1} = s$ and $x_{r+1} = e$. We write $\boldsymbol{r}_i \rightsquigarrow \boldsymbol{r}_j$ if there is a homomorphism $h\colon \boldsymbol{r}_i \to \boldsymbol{r}_j$ with $h(x_{i-1}) = x_{j-1}$ and $h(x_i) = x_j$. We call $\boldsymbol{q}_{lr}$ *right-periodic* if $l = 0$ and $\boldsymbol{r}_i \rightsquigarrow \boldsymbol{r}_1$ for all $i = 1, \dots, r$. By taking a mirror image of this definition, we obtain the notion of *left-periodic* 1-CQ, in which case $r = 0$ and $\boldsymbol{r}_{-i} \rightsquigarrow \boldsymbol{r}_0$ for all $i = 1, \dots, l$. A 1-CQ $\boldsymbol{q}$ is *periodic* if it is either right- or left-periodic, and *non-periodic* otherwise.

Theorem 12 (tetrachotomy)**.** *For any* $\boldsymbol{Q} = (\mathsf{cov}_A, \boldsymbol{q})$ *with a twinless path CQ* $\boldsymbol{q}$*, the following hold*:

1. *if* $\boldsymbol{q}$ *is a* 0*-CQ, then* $\boldsymbol{Q}$ *is in* AC^0;
2. *if* $\boldsymbol{q}$ *is a periodic* 1*-CQ, then* $\boldsymbol{Q}$ *is* NL*-complete*;
3. *if* $\boldsymbol{q}$ *is a non-periodic* 1*-CQ, then* $\boldsymbol{Q}$ *is* P*-complete*;
4. *if* $\boldsymbol{q}$ *is a* 2*-CQ then* $\boldsymbol{Q}$ *is* coNP*-complete.*

Item 1 is shown in Theorem 2 (i). The other two upper bounds follow from Theorems 5 and 14. The lower bounds follow from Theorems 13, 15 and 16.

We begin with the following criterion:

Theorem 13. *If* $\boldsymbol{q}$ *is a path* 1*-CQ, then* $(\mathsf{cov}_A, \boldsymbol{q})$ *is* NL*-hard.*

Proof. The proof is by an FO-reduction of the NL-complete reachability problem for dags. We assume that there exist a T-node x and an F-node y in $\boldsymbol{q}$ with $x \prec y$ (the other case is symmetric) and without any F- or T-nodes between them. Given a digraph $G = (V, E)$ with nodes $\mathfrak{s}, \mathfrak{t} \in V$, we construct an ABox $\mathcal{A}_G$ as follows. Replace each edge $e = (\mathfrak{u}, \mathfrak{v}) \in E$ by a fresh copy $\boldsymbol{q}^e$ of $\boldsymbol{q}$ such that node x in $\boldsymbol{q}^e$ is renamed to $\mathfrak{u}$ with $T(\mathfrak{u})$ being replaced by $A(\mathfrak{u})$, and node y is renamed to $\mathfrak{v}$ with $F(\mathfrak{v})$ being replaced by $A(\mathfrak{v})$. Then $\mathcal{A}_G$ comprises all such $\boldsymbol{q}^e$, for $e \in E$, as well as $T(\mathfrak{s})$ and $F(\mathfrak{t})$. We show that $\mathfrak{s} \to_G \mathfrak{t}$ iff $\mathsf{cov}_A, \mathcal{A}_G \models \boldsymbol{q}$.

$(\Rightarrow)$ Suppose there is a path $\mathfrak{s} = \mathfrak{v}_0, \dots, \mathfrak{v}_n = \mathfrak{t}$ in G with $e_i = (\mathfrak{v}_i, \mathfrak{v}_{i+1}) \in E$, for $i < n$. Then, for any model $\mathcal{I}$ of cov_A and $\mathcal{A}_G$, there is some $i < n$ such that $\mathfrak{v}_i \in T^\mathcal{I}$ and

$\mathfrak{v}_{i+1} \in F^{\mathcal{I}}$. Thus, the identity map from $\boldsymbol{q}$ to its copy $\boldsymbol{q}^{e_i}$ is a $\boldsymbol{q} \to \mathcal{I}$ homomorphism, and so $\mathcal{I} \models \boldsymbol{q}$.

($\Leftarrow$) Suppose $\mathfrak{s} \not\to_G \mathfrak{t}$. Define a model $\mathcal{I}$ of cov_A and $\mathcal{A}_G$ by labelling by T the undecided A-nodes in $\mathcal{A}_G$ that are reachable from $\mathfrak{s}$ (via a directed path) and by F the remaining ones. It is easy to see that there is no homomorphism from $\boldsymbol{q}$ to $\mathcal{I}$. ❑

By Theorem 5, all OMQs $\boldsymbol{Q}$ with a 1-CQ $\boldsymbol{q}$ are datalog-rewritable and lie in P. Our next task is to show that every such OMQ with a twinless path $\boldsymbol{q}$ is either linear-datalog-rewritable, and so NL-complete, or P-hard.

Theorem 14. *If $\boldsymbol{q}$ is a periodic twinless path 1-CQ, then $\boldsymbol{Q} = (\mathsf{cov}_A, \boldsymbol{q})$ is linear-datalog-rewritable, and so is in* NL.

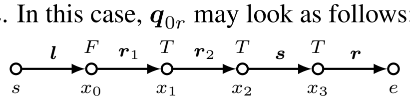
Proof. It is not hard to either construct an explicit linear datalog-rewriting of $\boldsymbol{Q}$ or show that every cactus in $\mathfrak{K}_{\boldsymbol{Q}}^{min}$ has branching number at most 1 and use Theorem 11. ❑

We next show that the OMQs with 1-CQs not covered by Theorem 14 are all P-hard.

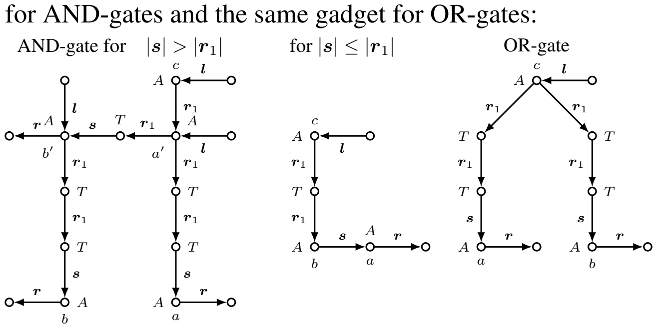
Theorem 15. *Let $\boldsymbol{q} = \boldsymbol{q}_{lr}$ be a twinless path 1-CQ such that one of the following conditions holds*: (i) $l, r \geq 1$, *or* (ii) $l = 0$ *and $\boldsymbol{q}_{0r}$ is not right-periodic, or* (iii) $r = 0$ *and $\boldsymbol{q}_{l0}$ is not left-periodic. Then* $(\mathsf{cov}_A, \boldsymbol{q}_{lr})$ *is* P*-hard.*

Proof. Each of the cases (i)–(iii) is proved by an FO-reduction of the P-complete monotone circuit evaluation problem. Below we only consider (ii). (The other two cases are proved using different gadgets.) We remind the reader that a *monotone Boolean circuit* is a dag $\boldsymbol{C}$ whose vertices are called *gates*. Gates with in-degree 0 are *input gates*. Each non-input gate g is either an AND-gate or an OR-gate, and has in-degree 2 (with the edges coming from the *inputs of* g). One of the non-input gates is distinguished as the *output gate*. Given an assignment α of F and T to the input gates of $\boldsymbol{C}$, we compute the value of each gate in $\boldsymbol{C}$ under α as usual in Boolean logic. The *output $\boldsymbol{C}(\alpha)$ of $\boldsymbol{C}$ on α* is the truth-value of the output gate.

Let $\boldsymbol{l} = [s, x_0)$, let $n > 1$ be minimal with $\boldsymbol{r}_n \not\rightsquigarrow \boldsymbol{r}_1$, $\boldsymbol{s} = \boldsymbol{r}_n$, and let $\boldsymbol{r} = (x_n, e]$. Below we consider the case of $n = 3$ only, but it should be clear how to modify the proof for other n. In this case, $\boldsymbol{q}_{0r}$ may look as follows:



We distinguish between 2 cases: $|\boldsymbol{s}| > |\boldsymbol{r}_1|$ and $|\boldsymbol{s}| \leq |\boldsymbol{r}_1|$. Depending on the case, we use the following two gadgets for AND-gates and the same gadget for OR-gates:



Given a monotone circuit $\boldsymbol{C}$ and an assignment α, we construct an ABox $\mathcal{A}_{\boldsymbol{C},\alpha}$ as follows. With each non-input gate g we associate a fresh copy of its gadget. When the inputs of g are gates g_a and g_b then, for each $i = a, b$, if g_i is a non-input gate then we merge node c of the gadget for g_i with the i-node in the gadget for g; and if g_i is an input gate, we replace the label A of i and i' (if available) in the gadget for g with $\alpha(g_i)$. Finally, we replace the label A of node c in the gadget for the output gate with F. We claim that $\mathsf{cov}_A, \mathcal{A}_{\boldsymbol{C},\alpha} \models \boldsymbol{q}_{0r}$ iff $\boldsymbol{C}(\alpha) = T$.

($\Leftarrow$) is proved by induction on the number of non-input gates in $\boldsymbol{C}$. The basis is obvious. For the induction step, suppose the output gate g in $\boldsymbol{C}$ is an AND-gate with inputs g_a and g_b, at least one of which is a non-input gate. Let $\mathcal{I}$ be an arbitrary model of cov_A and $\mathcal{A}_{\boldsymbol{C},\alpha}$. If both a and b in the gadget for g are in $T^{\mathcal{I}}$, then it is easy to check that we always have a $\boldsymbol{q}_{0r} \to \mathcal{I}$ homomorphism, no matter what the labels of a' and b' (if available) are. It remains to consider the case when either a or b is in $F^{\mathcal{I}}$, and so the corresponding g_i is not an input gate. Take the subcircuit $\boldsymbol{C}^-$ of $\boldsymbol{C}$ whose output gate is g_i. Then $\mathcal{A}_{\boldsymbol{C}^-,\alpha}$ is the sub-ABox of $\mathcal{A}_{\boldsymbol{C},\alpha}$ with the c-node in the gadget for g_i as its topmost node, and $A(c)$ replaced by $F(c)$. Now if $\mathcal{I}^-$ is the restriction of $\mathcal{I}$ to $\mathcal{A}_{\boldsymbol{C}^-,\alpha}$ (and so $c \in F^{\mathcal{I}^-}$), then by IH there is a $\boldsymbol{q}_{0r} \to \mathcal{I}^-$ homomorphism, and so $\mathcal{I} \models \boldsymbol{q}_{0r}$ as well. The case when the output gate g in $\boldsymbol{C}$ is an OR-gate is similar.

($\Rightarrow$) Suppose $\boldsymbol{C}(\alpha) = F$. To show $\mathsf{cov}_A, \mathcal{A}_{\boldsymbol{C},\alpha} \not\models \boldsymbol{q}_{0r}$, we define a model $\mathcal{I}$ of cov_A and $\mathcal{A}_{\boldsymbol{C},\alpha}$ inductively by labelling the A-nodes in the gadget for each non-input gate g of $\boldsymbol{C}$ by $F^{\mathcal{I}}$ or $T^{\mathcal{I}}$ as follows: node c is labelled by the truth-value of g under α, while node i (and node i' if applicable), for $i = a, b$, is labelled by the truth-value of g_i under α, where g_a and g_b are the inputs of g. Suppose, on the contrary, that there is a homomorphism $h\colon \boldsymbol{q}_{0r} \to \mathcal{I}$ and consider possible locations of $h(x_0) \in F^{\mathcal{I}}$. Suppose first that $|\boldsymbol{s}| > |\boldsymbol{r}_1|$ and $h(x_0)$ is in some AND-gadget.

Case $a, a' \in T^{\mathcal{I}}$, $b, b', c \in F^{\mathcal{I}}$: If $h(x_0) = c$, then $h(x_1) = a'$ and, since $b' \in F^{\mathcal{I}}$, the node $h(x_2)$ is the T-node just below a'. But then, since $|\boldsymbol{s}| > |\boldsymbol{r}_1|$, the node $h(x_3)$ must be strictly between a and the T-node above it, which is impossible because there are no T-nodes there. We obviously cannot have $h(x_0) = b'$ because $b \in F^{\mathcal{I}}$.

Case $a, a', c \in F^{\mathcal{I}}$, $b, b' \in T^{\mathcal{I}}$: If $h(x_0) = a'$, then $h(x_1)$ is the central T-node. But then, since $|\boldsymbol{s}| > |\boldsymbol{r}_1|$, node $h(x_2)$ must be strictly between b' and the central T-node, which is impossible because there are no T-nodes there.

Case $a, a', b, b', c \in F^{\mathcal{I}}$ is covered by the previous ones.

Suppose next that $|\boldsymbol{s}| \leq |\boldsymbol{r}_1|$ and $h(x_0) = c$ is in some AND-gadget. Then $h(x_2) = b$, provided that $b \in T^{\mathcal{I}}$ (otherwise such h is impossible), which means that $a \in F^{\mathcal{I}}$, and so $h(x_3)$ is located in some other gadget. However, this is impossible because of the following. In every gadget, the 'edges' leaving node c are labelled by $\boldsymbol{r}_1$. So if $|\boldsymbol{s}| < |\boldsymbol{r}_1|$ then $h(x_3)$ must be strictly between the c node of the gadget and the end-node of an $\boldsymbol{r}_1$-edge, but there are no T-nodes there. If $|\boldsymbol{s}| = |\boldsymbol{r}_1|$ then $\boldsymbol{s} \rightsquigarrow \boldsymbol{r}_1$, contrary to $\boldsymbol{s} \not\rightsquigarrow \boldsymbol{r}_1$.

Finally, if $h(x_0) = c$ is in some OR-gadget, then both a and b of the gadget are in F^I, and so $h(x_3) \in F^{\mathcal{I}}$, which is a contradiction. ❑

To complete our tetrachotomy, it remains to consider OMQs with 2-CQs.

Theorem 16. *If $\boldsymbol{q}$ is a twinless path 2-CQ, then $(\mathsf{cov}_A, \boldsymbol{q})$ is* coNP*-hard.*

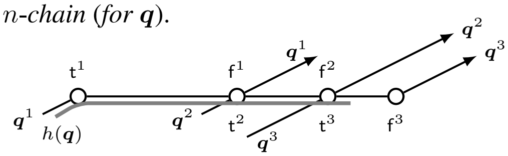
The proof is sketched in the remainder of this section. It is by a polynomial reduction of the complement of 3SAT: for every 3CNF ψ, we define (via a series of steps) an ABox $\mathcal{A}_\psi$ whose size is polynomial in the sizes of $\boldsymbol{q}$ and ψ, and then show that ψ is satisfiable iff $\mathsf{cov}_A, \mathcal{A}_\psi \not\models \boldsymbol{q}$.

We begin by introducing two general tools that will be used throughout. The following generalisation of homomorphisms will allow us to regard our CQs as if they contained a single binary predicate only. Given a model $\mathcal{I}$ of some ABox $\mathcal{A}$, we call a map $h\colon \boldsymbol{q} \to \mathcal{I}$ a *subhomomorphism* if the following hold:

– $h(x) \in T^{\mathcal{I}}$, for every T-node x in $\boldsymbol{q}$, and $h(x) \in F^{\mathcal{I}}$, for every F-node x in $\boldsymbol{q}$;

– for any nodes x, y in $\boldsymbol{q}$, if $R(x, y)$ is in $\boldsymbol{q}$ for some R, then $S\big(h(x), h(y)\big)$ is in $\mathcal{A}$ for some S.

Second, we define some ABoxes that are 'built up' from copies of $\boldsymbol{q}$ in a particular way. If $x \preceq y$, we let $\delta(x, y)$ denote the distance between x and y in $\boldsymbol{q}$, that is, the number of edges in the path from x to y, and set $|\boldsymbol{q}| = \delta(s, e)$. Given any path CQ $\boldsymbol{q}'$, we write $\prec_{\boldsymbol{q}'}$ and $\preceq_{\boldsymbol{q}'}$ for the ordering of nodes in $\boldsymbol{q}'$, and $\delta_{\boldsymbol{q}'}$ for the distance in $\boldsymbol{q}'$. We omit the subscripts when $\boldsymbol{q}' = \boldsymbol{q}$. Now let $\boldsymbol{q}^1, \dots, \boldsymbol{q}^n$, $n \geq 2$, be disjoint copies of $\boldsymbol{q}$. For any j and node x in $\boldsymbol{q}$, we let x^j denote the copy of x in $\boldsymbol{q}^j$, and let $\iota^j : \boldsymbol{q}^j \to \boldsymbol{q}$ be the identity map. We assume that $\boldsymbol{q}$ contains a T-node $\prec$-preceding an F-node (as the other case is symmetric). For each j, $1 \leq j \leq n$, we pick a T-node t^j and an F-node f^j in $\boldsymbol{q}^j$ such that $\mathsf{t}^j \prec_{\boldsymbol{q}^j} \mathsf{f}^j$; we call the selected nodes *contacts*. We replace the T- and F-labels of all the contacts with A, and then glue f^j together with t^{j+1} for every j with $1 \leq j < n$. We call the resulting contacts *glue-contacts* and the resulting ABox $\mathcal{H}$ an *n-chain* (*for $\boldsymbol{q}$*).



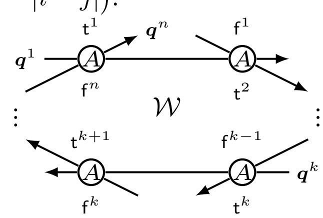
The following general criterion will give us flexibility in designing the ABox $\mathcal{A}_\psi$, and it will be used in the proofs of Lemmas 18, 19 and 21:

Lemma 17. *Suppose that $\mathcal{H}$ is an n-chain, for some $n \geq 2$. (i) If $h\colon \boldsymbol{q} \to \mathsf{ind}(\mathcal{H})$ is a function with $s^1 \prec_{\boldsymbol{q}^1} h(s) \prec_{\boldsymbol{q}^1} \mathsf{f}^1$, and $\mathcal{I}$ a model of cov_A and $\mathcal{H}$ whose glue-contacts are all in $F^{\mathcal{I}}$, then h is not a $\boldsymbol{q} \to \mathcal{I}$ subhomomorphism. (ii) If $h\colon \boldsymbol{q} \to \mathsf{ind}(\mathcal{H})$ is a function with $\mathsf{t}^n \prec_{\boldsymbol{q}^n} h(e) \prec_{\boldsymbol{q}^n} e^n$, and $\mathcal{I}$ a model of cov_A and $\mathcal{H}$ whose glue-contacts are all in $T^{\mathcal{I}}$, then h is not a $\boldsymbol{q} \to \mathcal{I}$ subhomomorphism.*

Proof. (i) Suppose on the contrary that $h\colon \boldsymbol{q} \to \mathcal{I}$ is a subhomomorphism. For each j, let $\iota^j : \boldsymbol{q}^j \to \boldsymbol{q}$ be the identity map. We define a 'shift' function $g_\leftarrow\colon \mathsf{ind}(\mathcal{H}) \to \mathsf{ind}(\mathcal{H})$ by taking $g_\leftarrow(x) = h\big(\iota^j(x)\big)$ whenever x is a node in $\boldsymbol{q}^j$, where we consider each glue-contact $\mathsf{c} = \mathsf{f}^i = \mathsf{t}^{i+1}$, for $1 \leq i < n$, as a node in $\boldsymbol{q}^{i+1}$, that is, $g_\leftarrow(\mathsf{c}) = g_\leftarrow(\mathsf{t}^{i+1}) = h\big(\iota^{i+1}(\mathsf{t}^{i+1})\big)$. As $\mathcal{H}$ is finite, there exists a 'fixpoint' of $g_\leftarrow$: a node x in $\mathcal{H}$ and a number $N > 0$ such that $g_\leftarrow^N(x) = x$. It is not hard to see that the assumption in (i) implies that the 'fixpoint-cycle' $x, g_\leftarrow(x), g_\leftarrow^2(x), \dots, g_\leftarrow^{N-1}(x), x$ can be 'shifted to the left' in the sense that there exists a glue-contact c with $g_\leftarrow^N(\mathsf{c}) = \mathsf{c}$. But this leads to a contradiction. On the one hand, $\mathsf{c} \in F^{\mathcal{I}}$ by our assumption, and so c cannot be in $T^{\mathcal{I}}$ by the minimality of $\mathcal{I}$. On the other hand, it can be shown by induction on j that $g_\leftarrow^j(\mathsf{c}) \in T^{\mathcal{I}}$, and so $\mathsf{c} = g_\leftarrow^N(\mathsf{c}) \in T^{\mathcal{I}}$.

The proof of (ii) is similar, using a shift function $g_\rightarrow$ where we consider each contact $\mathsf{c} = \mathsf{f}^i = \mathsf{t}^{i+1}$ as a node in $\boldsymbol{q}^i$, that is, $g_\rightarrow(\mathsf{c}) = g_\rightarrow(\mathsf{f}^i) = h\big(\iota^i(\mathsf{f}^i)\big)$. Then we shift the fixpoint-cycle of $g_\rightarrow$ to the right. ❑

Given a 3CNF ψ, we now start building the ABox $\mathcal{A}_\psi$ from copies of the 2-CQ $\boldsymbol{q}$. We begin with structures that will be used to encode the truth-values of literals (variables and negations thereof) in the clauses of ψ. We take an n-chain for $\boldsymbol{q}$ and some $n \geq |\boldsymbol{q}|$, and glue together its t^1 and f^n contacts, replacing their respective T- and F-labels with A. If the contacts are such that $\iota^{j+1}(\mathsf{t}^{j+1}) \prec \iota^j(\mathsf{f}^j)$ for every j, then the resulting ABox $\mathcal{W}$ is called an *n-cogwheel* (throughout we assume that $\pm$ is modulo n). For each j, the nodes preceding t^j in $\boldsymbol{q}^j$ form its *initial cog*, while the nodes succeeding f^j in $\boldsymbol{q}^j$ form its *final cog*. Given two contacts $\mathsf{c}_1 = \mathsf{f}^i = \mathsf{t}^{i+1}$ and $\mathsf{c}_2 = \mathsf{f}^j = \mathsf{t}^{j+1}$, we define the *contact-distance between c_1 and c_2 in $\mathcal{W}$* as $\min\big(|i - j|, n - |i - j|\big)$.

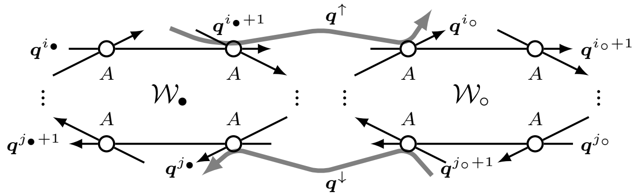


Lemma 18. *Suppose $\mathcal{W}$ is an n-cogwheel for some $n \geq |\boldsymbol{q}|$. For any model $\mathcal{I}$ of cov_A and $\mathcal{W}$, we have $\mathcal{I} \not\models \boldsymbol{q}$ iff the contacts in $\mathcal{I}$ are either all in $T^{\mathcal{I}}$ or all in $F^{\mathcal{I}}$.*

Next, for each variable p occurring in the 3CNF ψ, we take a fresh pair of cogwheels and make sure that they always encode the opposite truth-values of the literals p and $\neg p$. To achieve this, we connect the cogwheels in each pair with two additional copies of $\boldsymbol{q}$ in a special way.

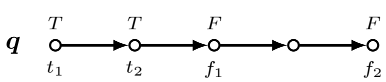
Let $\mathcal{W}_\bullet$ and $\mathcal{W}_\circ$ be two disjoint n-cogwheels for some $n > 4|\boldsymbol{q}| + 2$, and let $\boldsymbol{q}^\uparrow$, $\boldsymbol{q}^\downarrow$ be two more fresh and disjoint copies of $\boldsymbol{q}$. For $j = \uparrow, \downarrow$ and node x in $\boldsymbol{q}$, we let x^j denote the copy of x in $\boldsymbol{q}^j$. We pick two contacts $\mathsf{c}_\bullet^\uparrow = \mathsf{f}^{i_\bullet} = \mathsf{t}^{i_\bullet+1}$ and $\mathsf{c}_\bullet^\downarrow = \mathsf{f}^{j_\bullet} = \mathsf{t}^{j_\bullet+1}$ in $\mathcal{W}_\bullet$ such that they are 'far' from each other either way, that is, the contact-distance between them in $\mathcal{W}_\bullet$ is $> 2|\boldsymbol{q}|$. Similarly, we pick two contacts $\mathsf{c}_\circ^\uparrow = \mathsf{f}^{i_\circ} = \mathsf{t}^{i_\circ+1}$ and $\mathsf{c}_\circ^\downarrow = \mathsf{f}^{j_\circ} = \mathsf{t}^{j_\circ+1}$ in $\mathcal{W}_\circ$ such that the contact-distance between them in $\mathcal{W}_\circ$ is $> 2|\boldsymbol{q}|$. Then we glue together the contact $\mathsf{c}_\bullet^\uparrow$ in $\mathcal{W}_\bullet$ with $f_1^\uparrow$, and also the contact $\mathsf{c}_\circ^\uparrow$ in $\mathcal{W}_\circ$ with $f_2^\uparrow$, having the F-labels of $f_1^\uparrow$ and $f_2^\uparrow$

replaced with A. Finally, we glue together the contact $\mathsf{c}_\circ^\downarrow$ in $\mathcal{W}_\circ$ with $t_1^\downarrow$, and also the contact $\mathsf{c}_\bullet^\downarrow$ in $\mathcal{W}_\bullet$ with $t_2^\downarrow$, having the T-labels of $t_1^\downarrow$ and $t_2^\downarrow$ replaced with A. The resulting ABox $\mathcal{B}$ is called an *n-bike*. We call the contacts $\mathsf{c}_\bullet^\uparrow = \mathsf{f}^{i_\bullet} = \mathsf{t}^{i_\bullet+1} = f_1^\uparrow$ and $\mathsf{c}_\circ^\uparrow = \mathsf{f}^{j_\circ} = \mathsf{t}^{j_\circ+1} = f_2^\uparrow$ *F-connections* and the contacts $\mathsf{c}_\circ^\downarrow = \mathsf{f}^{j_\circ} = \mathsf{t}^{j_\circ+1} = t_1^\downarrow$ and $\mathsf{c}_\bullet^\downarrow = \mathsf{f}^{j_\bullet} = \mathsf{t}^{j_\bullet+1} = t_2^\downarrow$ *T-connections* in $\mathcal{B}$.

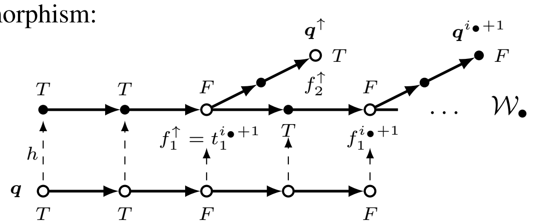


Throughout, for any k, we let t_k (f_k) denote the kth T-node (F-node) in $\boldsymbol{q}$. In particular, t_{last-1} (f_{last-1}) denotes the last but one T-node (F-node) in $\boldsymbol{q}$, and t_{last} (f_{last}) the last T-node (F-node). We assume that $t_1 \prec f_1$ (the other case is symmetric).

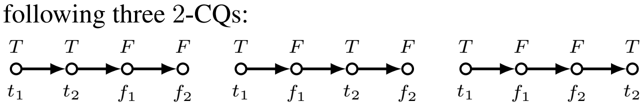
We want to achieve that, for any model $\mathcal{I}$ of cov_A and $\mathcal{B}$, we have $\mathcal{I} \not\models \boldsymbol{q}$ iff the contacts in $\mathcal{W}_\bullet$ are all in $T^\mathcal{I}$ while the contacts in $\mathcal{W}_\circ$ are all in $F^\mathcal{I}$, or the other way round. Using Lemma 18 and the fact that the F-connections are F-nodes in $\boldsymbol{q}^\uparrow$ while the T-connections are T-nodes in $\boldsymbol{q}^\downarrow$, it is straightforward to see that the implication ($\Rightarrow$) always holds for any n-bike $\mathcal{B}$. However, for the ($\Leftarrow$) direction to hold, we need to choose the contacts in the '$\pm|\boldsymbol{q}|$-size environments' of the F- and T-connections in the n-cogwheels $\mathcal{W}_\bullet$ and $\mathcal{W}_\circ$ carefully, in such a way that all possible locations in $\mathcal{B}$ for the image $h(\boldsymbol{q})$ of a potential homomorphism $h\colon \boldsymbol{q} \to \mathcal{I}$ are excluded. Our choices depend on the particular 2-CQ. For example, consider



If we choose $\mathsf{t}^{i_\bullet+1} = t_1^{i_\bullet+1}$ and $\mathsf{f}^{i_\bullet+1} = f_1^{i_\bullet+1}$, and $\mathcal{I}$ is such that all contacts in $\mathcal{W}_\bullet$ are in $F^\mathcal{I}$ (and all contacts in $\mathcal{W}_\circ$ are in $T^\mathcal{I}$), then we have the following $h\colon \boldsymbol{q} \to \mathcal{I}$ homomorphism:



On the other hand, By Lemma 19 below, the choices of $\mathsf{t}^{i_\bullet+1} = t_1^{i_\bullet+1}$ and $\mathsf{f}^{i_\bullet+1} = f_1^{i_\bullet+1}$ are good for any of the following three 2-CQs:



For each particular 2-CQ $\boldsymbol{q}$, there might be different ways of choosing the contacts so that all potential homomorphisms are excluded. Sometimes the choices are straightforward, some other times not so. In Lemma 19 below, we describe a system of choices that works for *every* 2-CQ (in other words, we give an algorithm that provides suitable choices for any $\boldsymbol{q}$). The different potential locations of a homomorphic image impose different constraints on the possible choices of contacts. Our 'meta-heuristics' in finding a solution to such a constraint system is to keep the contacts 'as close as possible' to each other so that most non-contact T- and F-nodes must be in the cogs of the cogwheels. This way potential homomorphic images are 'forced' to intersect with cogs, where there are fewer options for them: say, if h maps a node x of $\boldsymbol{q}$ to the initial cog of a copy $\boldsymbol{q}^j$, then we must have $x \preceq \iota^j\big(h(x)\big)$, for otherwise there is not enough room for the whole $h(\boldsymbol{q})$ in the cog.

Lemma 19. *Suppose $\mathcal{B}$ is an n-bike such that the following conditions hold for its F-connections:*

- *if $t_2 \prec f_1$ and $\delta(f_1, f_2) \geq \delta(t_1, f_1)$ then $\mathsf{t}^{i_\bullet+k} = t_2^{i_\bullet+k}$ and $\mathsf{f}^{i_\bullet+k} = f_2^{i_\bullet+k}$, for all k, $1 \leq k \leq |\boldsymbol{q}|$; otherwise, $\mathsf{t}^{i_\bullet+k} = t_1^{i_\bullet+k}$ and $\mathsf{f}^{i_\bullet+k} = f_1^{i_\bullet+k}$, for all k, $1 \leq k \leq |\boldsymbol{q}|$;*
- *$\mathsf{f}^{i_\bullet} = f_2^{i_\bullet}$ and $\mathsf{f}^{i_\bullet-k} = f_1^{i_\bullet-k}$, for all k, $0 < k \leq |\boldsymbol{q}|$;*
- *$\mathsf{t}^{i_\bullet-k} = t_1^{i_\bullet-k}$ and $\mathsf{t}^{i_\circ-k} = t_1^{i_\circ-k}$, for all k, $k \leq |\boldsymbol{q}|$;*
- *$\mathsf{t}^{i_\circ+1} = t_1^{i_\circ+1}$ and $\mathsf{f}^{i_\circ+1} = f_1^{i_\circ+1}$;*
- *$\mathsf{f}^{i_\circ} = f_2^{i_\circ}$ and $\mathsf{f}^{i_\circ-k} = f_1^{i_\circ-k}$, for all k, $0 < k \leq |\boldsymbol{q}|$;*

and the following conditions hold for its T-connections:

- *$\mathsf{t}^{j_\circ+1} = t_1^{j_\circ+1}$, $\mathsf{t}^{j_\circ-k} = t_1^{j_\circ-k}$ and $\mathsf{f}^{j_\circ-k} = f_1^{j_\circ-k}$, for all $k \leq |\boldsymbol{q}|$;*
- *if $t_2 \prec f_1$ and $\delta(f_1, f_2) \geq \delta(t_1, f_1)$ then $\mathsf{t}^{j_\bullet+k} = t_2^{j_\bullet+k}$ and $\mathsf{f}^{j_\bullet+k} = f_2^{j_\bullet+k}$, for all k, $1 \leq k \leq |\boldsymbol{q}|$; otherwise, $\mathsf{t}^{j_\bullet+k} = t_1^{j_\bullet+k}$ and $\mathsf{f}^{j_\bullet+k} = f_1^{j_\bullet+k}$, for all k, $1 \leq k \leq |\boldsymbol{q}|$.*

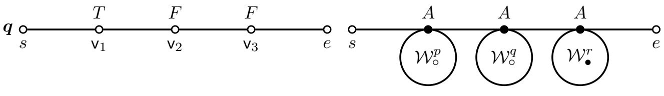
Then, for any model $\mathcal{I}$ of cov_A and $\mathcal{B}$, we have $\mathcal{I} \not\models \boldsymbol{q}$ iff the contacts in $\mathcal{W}_\bullet$ are all in $T^\mathcal{I}$ while the contacts in $\mathcal{W}_\circ$ are all in $F^\mathcal{I}$, or the other way round.

Note that the contact choices above are well-defined in any n-bike $\mathcal{B}$ as, for each of the cogwheels in $\mathcal{B}$, the contact-distance between its F- and T-connections is $> 2|q|$.

Let ψ be a 3CNF (a conjunction of n_ψ clauses of the form $\ell_1 \vee \ell_2 \vee \ell_3$, where each ℓ_i is a *literal*, that is, a propositional variable or its negation) and let $n > (n_\psi + 2) \cdot |\boldsymbol{q}|$. For each propositional variable p in ψ, we take a fresh n-bike $\mathcal{B}^p$ having n-cogwheels $\mathcal{W}_\bullet^p$, $\mathcal{W}_\circ^p$ and satisfying the conditions of Lemma 19. We pick three nodes v_1, v_2 and v_3 in $\boldsymbol{q}$ such that each v_a is a T-node or an F-node, and $\mathsf{v}_1 \prec \mathsf{v}_2 \prec \mathsf{v}_3$. Then, for every clause $c = (\ell_1^c \vee \ell_2^c \vee \ell_3^c)$ in ψ, we proceed as follows. We take a fresh copy $\boldsymbol{q}^c$ of $\boldsymbol{q}$, consider the copies v_1^c, v_2^c and v_3^c of the chosen nodes in $\boldsymbol{q}^c$, and replace their F- or T-labels with A. Further, for $a = 1, 2, 3$, we glue v_a^c to a fresh (unused as T- or F-connections) contact

- in $\mathcal{W}_\bullet^p$ iff either $\ell_a^c = p$ and v_a is an F-node in $\boldsymbol{q}$, or $\ell_a^c = \neg p$ and v_a is a T-node in $\boldsymbol{q}$;
- in $\mathcal{W}_\circ^p$ iff either $\ell_a^c = p$ and v_a is an T-node in $\boldsymbol{q}$, or $\ell_a^c = \neg p$ and v_a is a F-node in $\boldsymbol{q}$.

We call the chosen contacts in the three n-cogwheels the *wheel-contacts for c*. For example, if $\boldsymbol{q}$ looks like on the left-hand side of the picture below and $c = (p \vee \neg q \vee r)$, then we obtain the graph shown on the right-hand side of the picture with the n-cogwheels depicted as circles:



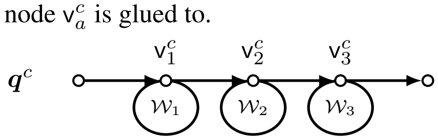
We pick the wheel-contacts for different clauses in each n-cogwheel in such a way that their contact-distance from each other and from the F- and T-connections of the n-cogwheel is $> 2|\boldsymbol{q}|$. We treat the resulting labelled graph as an ABox and denote it by $\mathcal{A}_\psi$.

The following lemma is a consequence of the definition of $\mathcal{A}_\psi$, and the 'easy' ($\Rightarrow$) direction of Lemma 19.

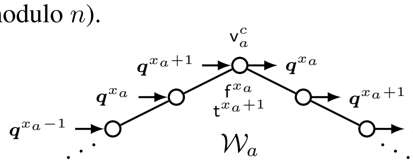
Lemma 20. *If* $\mathsf{cov}_A, \mathcal{A}_\psi \not\models \boldsymbol{q}$*, then* ψ *is satisfiable.*

It remains to find some conditions on $\mathcal{A}_\psi$ that guarantee the converse of Lemma 20. So suppose ψ is satisfiable under an assignment $\mathfrak{a}$. We define a model $\mathcal{I}_\mathfrak{a}$ of cov_A and $\mathcal{A}_\psi$ as follows: For every variable p in ψ, if $\mathfrak{a}(p) = T$ then we put all contacts of the n-cogwheel $\mathcal{W}^p_\bullet$ to $T^{\mathcal{I}_\mathfrak{a}}$ and all contacts of the n-cogwheel $\mathcal{W}^p_\circ$ to $F^{\mathcal{I}_\mathfrak{a}}$; if $\mathfrak{a}(p) = F$, we put all contacts of $\mathcal{W}^p_\bullet$ to $F^{\mathcal{I}_\mathfrak{a}}$ and all contacts of $\mathcal{W}^p_\circ$ to $T^{\mathcal{I}_\mathfrak{a}}$. We aim to find some conditions on $\mathcal{A}_\psi$ that would imply $\mathcal{I}_\mathfrak{a} \not\models \boldsymbol{q}$.

To formulate these conditions, we introduce a new notation for the three wheel-contacts, uniformly for any given clause c in ψ (not depending on c, but only on $a = 1, 2, 3$ and $\boldsymbol{q}$). For each $a = 1, 2, 3$, we let $\mathcal{W}_a$ denote the n-cogwheel the node v^c_a is glued to.



The wheel-contact for c in $\mathcal{W}_a$ was obtained (when forming the n-cogwheel $\mathcal{W}_a$) by glueing together the F-node f^{x_a} of some copy $\boldsymbol{q}^{x_a}$ and the T-node t^{x_a+1} of some copy $\boldsymbol{q}^{x_a+1}$ ($\pm$ is modulo n).



For each $a = 1, 2, 3$, we need to choose the contacts v_a, $\mathsf{f}^{x_a \pm k}$ and $\mathsf{t}^{x_a \pm k}$, for $k \le |\boldsymbol{q}|$, in such a way that $\mathcal{I}_\mathfrak{a} \not\models \boldsymbol{q}$ (and so the converse of Lemma 20 holds). There might be different ways of choosing these contacts so that all potential $\boldsymbol{q} \to \mathcal{I}_\mathfrak{a}$ homomorphisms are excluded. Our algorithm below selects contacts that are suitable for ψ and $\boldsymbol{q}$ uniformly, depending only on the particular 2-CQ $\boldsymbol{q}$, but not on the satisfying assignment $\mathfrak{a}$. While this 'heuristic' choice results in a case-distinction with fewer cases, in each case our task now is a bit harder than in the proof of Lemma 19. We do not have any information about the particular labelings of v^c_1, v^c_2 and v^c_3 in $\mathcal{I}_\mathfrak{a}$ other than the fact that the cogwheel attached to each of them represents a truth-value: for each $a = 1, 2, 3$, the contacts in $\mathcal{W}_a$ are either all in $T^{\mathcal{I}_\mathfrak{a}}$ or all in $F^{\mathcal{I}_\mathfrak{a}}$ (see the definition of $\mathcal{I}_\mathfrak{a}$ above).

Below, we use the following notation: $t_\Box$ denotes the last T-node preceding f_1, $t_\diamond$ denotes the last T-node preceding f_2, and $t_\sharp$ denotes the last T-node preceding f_{last}. (These all are well-defined, as $t_1 \prec f_1$ by our assumption.) We call $\mathcal{A}_\psi$ a *ψ-gadget* if the following conditions hold for all clauses c in ψ, all $k \le |\boldsymbol{q}|$, and all ℓ with $1 \le \ell \le |\boldsymbol{q}|$:

– $\mathsf{v}_1 = t_1$, $\mathsf{t}^{x_1+1} = t_1^{x_1+1}$, $\mathsf{t}^{x_1-k} = t_\Box^{x_1-k}$;

– if $t_{last} \prec f_1$ then $\mathsf{f}^{x_1-k} = f_2^{x_1-k}$, $\mathsf{v}_2 = t_{last}$, $\mathsf{t}^{x_2+\ell} = t_{last-1}^{x_2+\ell}$, $\mathsf{f}^{x_2+\ell} = f_1^{x_2+\ell}$, $\mathsf{t}^{x_2-k} = t_1^{x_2-k}$,

$$\mathsf{f}^{x_2-k} = \begin{cases} f_1^{x_2-k}, & \text{if } \delta(t_1, t_2) = \dots = \delta(t_{last-1}, t_{last}) \\ & \qquad = \delta(t_{last}, f_2), \\ f_2^{x_2-k}, & \text{otherwise;} \end{cases}$$

– if $f_1 \prec t_{last}$ then $\mathsf{f}^{x_1-k} = f_1^{x_1-k}$, $\mathsf{v}_2 = f_1$, $\mathsf{t}^{x_2+\ell} = t_\Box^{x_2+\ell}$, $\mathsf{f}^{x_2+\ell} = f_1^{x_2+\ell}$, $\mathsf{f}^{x_2-k} = f_2^{x_2-k}$, and there are two cases:

(i) if $f_1 \prec t_\diamond$ and there exist some T-node $t \prec t_\diamond$ and $k_t \ge 1$ with $\delta(t_\Box, f_1) = \delta(t, t_\diamond) + k_t \cdot \delta(t_\diamond, f_2)$, then let $t_\star$ be such a t with the smallest k_t, $\mathsf{t}^{x_2-(k_{t_\star}-1)} = t_\star^{x_2-(k_{t_\star}-1)}$, and $\mathsf{t}^{x_2-k} = t_\diamond^{x_2-k}$ for $k \ne k_{t_\star} - 1$;

(ii) otherwise, $\mathsf{t}^{x_2-k} = t_\diamond^{x_2-k}$;

– $\mathsf{v}_3 = f_{last}$, $\mathsf{t}^{x_3-k} = t_\sharp^{x_3-k}$, $\mathsf{f}^{x_3-k} = f_{last}^{x_3-k}$, $\mathsf{t}^{x_3+\ell} = t_\sharp^{x_3+\ell}$, $\mathsf{f}^{x_3+\ell} = f_{last}^{x_3+\ell}$.

Lemma 21. *If* $\mathcal{A}_\psi$ *is a* ψ*-gadget, then* $\mathcal{I}_\mathfrak{a} \not\models \boldsymbol{q}$*.*

Note that the contact choices above are well-defined in any ψ-gadget $\mathcal{A}_\psi$, as the wheel-contacts for different clauses in each cogwheel are such that their contact-distance from each other and from the F- and T-connections of the cogwheel is always $> 2|\boldsymbol{q}|$.

Finally, given a 3CNF ψ, take some ψ-gadget $\mathcal{A}_\psi$. By Lemmas 20 and 21, we have $\mathsf{cov}_A, \mathcal{A}_\psi \not\models \boldsymbol{q}$ iff ψ is satisfiable. This completes the proof of Theorem 16.

5 Conclusions

This paper contributes to the area of research into the non-uniform complexity of OMQ answering. Although there exist algorithms that are capable of deciding various types of rewritability of a given OMQ, they are of so high complexity that complete syntactic and practical general classifications of OMQs are hardly possible. We take a different route to understanding the problem by stripping it to the bare bones: we fix the ontology cov_A with a single axiom saying that A is covered by the union of F and T, and consider CQs with unary predicates F, T and arbitrary binary predicates (in fact, one binary predicate is already extremely challenging). This 'primitivisation' pays off as we obtain a number of useful sufficient conditions for membership in AC^0/L/NL/P. However, to our great surprise, it turns out that checking FO-rewritability of OMQs with a 1-CQ still remains in the range between PSPACE and 2EXPTIME. We finally obtain a remarkably transparent syntactic AC^0/NL/P/coNP tetrachotomy (requiring a pretty complex proof) for path CQs that do not contain occurrences of FT-twins.

As a next step, we would like to extend our tetrachotomy to the ontology $\mathsf{cov}_\top$, which is in *DL-Lite*$_{krom}$ (Artale et al. 2009), and to (possibly tree-shaped) CQs containing FT-twins (adding unary predicates seems less problematic). We are working on pinpointing the exact complexity of classifying OMQs with covering. In particular, we believe that deciding FO-rewritability for OMQs with a 1-CQ is actually 2EXPTIME-complete. Finally, we would also like to understand the connection of our problem to CSPs.

Acknowledgements

The work of O. Gerasimova was supported by the Russian Academic Excellence Project 5-100. The work of M. Zakharyaschev was supported by the EPSRC U.K. grant EP/S032282. We thank the anonymous referees for their careful reading and expert criticism of the manuscript.

References

Afrati, F. N.; Gergatsoulis, M.; and Toni, F. 2003. Linearisability on datalog programs. *Theor. Comput. Sci.* 308(1-3):199–226.

Artale, A.; Calvanese, D.; Kontchakov, R.; and Zakharyaschev, M. 2009. The DL-Lite family and relations. *Journal of Artificial Intelligence Research (JAIR)* 36:1–69.

Benedikt, M.; ten Cate, B.; Colcombet, T.; and Vanden Boom, M. 2015. The complexity of boundedness for guarded logics. In *30th Annual ACM/IEEE Symposium on Logic in Computer Science, LICS 2015, Kyoto, Japan, July 6-10, 2015*, 293–304. IEEE Computer Society.

Bienvenu, M.; ten Cate, B.; Lutz, C.; and Wolter, F. 2014. Ontology-based data access: A study through disjunctive datalog, CSP, and MMSNP. *ACM Transactions on Database Systems* 39(4):33:1–44.

Bulatov, A. A. 2017. A dichotomy theorem for nonuniform CSPs. In Umans, C., ed., *58th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2017, Berkeley, CA, USA, October 15-17, 2017*, 319–330. IEEE Computer Society.

Calvanese, D.; De Giacomo, G.; Lembo, D.; Lenzerini, M.; and Rosati, R. 2007. Tractable reasoning and efficient query answering in description logics: the *DL-Lite* family. *Journal of Automated Reasoning* 39(3):385–429.

Cosmadakis, S. S.; Gaifman, H.; Kanellakis, P. C.; and Vardi, M. Y. 1988. Decidable optimization problems for database logic programs (preliminary report). In *STOC*, 477–490.

Egri, L.; Larose, B.; and Tesson, P. 2007. Symmetric datalog and constraint satisfaction problems in logspace. In *Logic in Computer Science, 2007. LICS 2007. 22nd Annual IEEE Symposium on*, 193–202. IEEE.

Eiter, T.; Ortiz, M.; Šimkus, M.; Tran, T.; and Xiao, G. 2012. Query rewriting for Horn-$\mathcal{SHIQ}$ plus rules. In Hoffmann, J., and Selman, B., eds., *Proceedings of the Twenty-Sixth AAAI Conference on Artificial Intelligence, July 22-26, 2012, Toronto, Ontario, Canada*. AAAI Press.

Gault, R., and Jeavons, P. 2004. Implementing a test for tractability. *Constraints* 9(2):139–160.

Gerasimova, O.; Kikot, S.; Kurucz, A.; Podolskii, V. V.; and Zakharyaschev, M. 2020. A data complexity and rewritability tetrachotomy of ontology-mediated queries with a covering axiom. *CoRR* abs/2006.04167.

Hernich, A.; Lutz, C.; Ozaki, A.; and Wolter, F. 2015. Schema.org as a description logic. In Yang, Q., and Wooldridge, M. J., eds., *Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence, IJCAI 2015, Buenos Aires, Argentina, July 25-31, 2015*, 3048–3054. AAAI Press.

Hovland, D.; Kontchakov, R.; Skjæveland, M. G.; Waaler, A.; and Zakharyaschev, M. 2017. Ontology-based data access to Slegge. In d'Amato, C.; Fernández, M.; Tamma, V. A. M.; Lécué, F.; Cudré-Mauroux, P.; Sequeda, J. F.; Lange, C.; and Heflin, J., eds., *The Semantic Web - ISWC 2017 - 16th International Semantic Web Conference, Vienna, Austria, October 21-25, 2017, Proceedings, Part II*, volume 10588 of *Lecture Notes in Computer Science*, 120–129. Springer.

Hustadt, U.; Motik, B.; and Sattler, U. 2005. Data complexity of reasoning in very expressive description logics. In Kaelbling, L. P., and Saffiotti, A., eds., *IJCAI-05, Proceedings of the Nineteenth International Joint Conference on Artificial Intelligence, Edinburgh, Scotland, UK, July 30 - August 5, 2005*, 466–471. Professional Book Center.

Immerman, N. 1999. *Descriptive Complexity*. Springer.

Kaminski, M.; Nenov, Y.; and Grau, B. C. 2016. Datalog rewritability of disjunctive datalog programs and non-Horn ontologies. *Artif. Intell.* 236:90–118.

Kharlamov, E.; Hovland, D.; Skjæveland, M. G.; Bilidas, D.; Jiménez-Ruiz, E.; Xiao, G.; Soylu, A.; Lanti, D.; Rezk, M.; Zheleznyakov, D.; Giese, M.; Lie, H.; Ioannidis, Y. E.; Kotidis, Y.; Koubarakis, M.; and Waaler, A. 2017. Ontology based data access in Statoil. *J. Web Sem.* 44:3–36.

Lutz, C., and Sabellek, L. 2017. Ontology-mediated querying with the description logic EL: trichotomy and linear datalog rewritability. In Sierra, C., ed., *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI 2017, Melbourne, Australia, August 19-25, 2017*, 1181–1187. ijcai.org.

Lutz, C., and Wolter, F. 2012. Non-uniform data complexity of query answering in description logics. In Brewka, G.; Eiter, T.; and McIlraith, S. A., eds., *Principles of Knowledge Representation and Reasoning: Proceedings of the Thirteenth International Conference, KR 2012, Rome, Italy, June 10-14, 2012*. AAAI Press.

Marcinkowski, J. 1996. DATALOG sirups uniform boundedness is undecidable. In *Proceedings, 11th Annual IEEE Symposium on Logic in Computer Science, New Brunswick, New Jersey, USA, July 27-30, 1996*, 13–24. IEEE Computer Society.

Papadimitriou, C. H., and Yannakakis, M. 1986. A note on succinct representations of graphs. *Information and Control* 71(3):181–185.

Poggi, A.; Lembo, D.; Calvanese, D.; De Giacomo, G.; Lenzerini, M.; and Rosati, R. 2008. Linking data to ontologies. *Journal on Data Semantics* X:133–173.

Ramakrishnan, R.; Sagiv, Y.; Ullman, J. D.; and Vardi, M. Y. 1989. Proof-tree transformation theorems and their applications. In *Proceedings of the eighth ACM SIGACT-SIGMOD-SIGART symposium on Principles of database systems*, 172–181. ACM.

Rosati, R. 2007. On conjunctive query answering in EL.

In Calvanese, D.; Franconi, E.; Haarslev, V.; Lembo, D.; Motik, B.; Turhan, A.; and Tessaris, S., eds., *Proceedings of the 2007 International Workshop on Description Logics (DL2007), Brixen-Bressanone, near Bozen-Bolzano, Italy, 8-10 June, 2007*, volume 250 of *CEUR Workshop Proceedings*. CEUR-WS.org.

Saraiya, Y. P. 1989. Linearizing nonlinear recursions in polynomial time. In Silberschatz, A., ed., *Proceedings of the Eighth ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems, March 29-31, 1989, Philadelphia, Pennsylvania, USA*, 182–189. ACM Press.

Stockmeyer, L. J. 1976. The polynomial-time hierarchy. *Theor. Comput. Sci.* 3(1):1–22.

Ullman, J. D., and Gelder, A. V. 1988. Parallel complexity of logical query programs. *Algorithmica* 3:5–42.

Vardi, M. Y. 1988. Decidability and undecidability results for boundedness of linear recursive queries. In Edmondson-Yurkanan, C., and Yannakakis, M., eds., *Proceedings of the Seventh ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems, March 21-23, 1988, Austin, Texas, USA*, 341–351. ACM.

Zhang, W.; Yu, C. T.; and Troy, D. 1990. Necessary and sufficient conditions to linearize double recursive programs in logic databases. *ACM Trans. Database Syst.* 15(3):459–482.

Zhuk, D. 2017. A proof of CSP dichotomy conjecture. In Umans, C., ed., *58th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2017, Berkeley, CA, USA, October 15-17, 2017*, 331–342. IEEE Computer Society.