

# A Tetrachotomy of Ontology-Mediated Queries with a Covering Axiom

Olga Gerasimova<sup>a</sup>, Stanislav Kikot<sup>b</sup>, Agi Kurucz<sup>c</sup>, Vladimir Podolskii<sup>d,a</sup>, Michael Zakharyashev<sup>e</sup>

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

<sup>b</sup>School of Computing and Digital Media, London Metropolitan University, U.K.

<sup>c</sup>Department of Informatics, King's College London, U.K.

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

<sup>e</sup>Department of Computer Science, Birkbeck, University of London, U.K.

---

## Abstract

We are interested in the problem of efficiently determining the data complexity of answering queries mediated by non-Horn description logic ontologies and constructing their optimal rewritings to standard database queries. In general, this problem is known to be extremely complex. In this article, we strip it to the bare bones and focus on conjunctive queries mediated by a *simple covering axiom* stating that one class is covered by the union of two other classes. We develop a novel technique to prove that, quite surprisingly, deciding first-order rewritability of even such simple ontology-mediated queries is PSPACE-hard. The main result of this article is a complete and transparent syntactic AC<sup>0</sup>/NL/P/coNP tetrachotomy of *path* queries under the assumption that the covering classes are disjoint. We also obtain a number of syntactic and semantic sufficient conditions (without the path query assumption) for membership in AC<sup>0</sup>, L, NL, and P.

**Keywords:** Ontology-based data access, description logic, first-order rewritability, data complexity.

---

## 1. Introduction

The general research problem we are concerned with in this article can be formulated as follows: for any fixed ontology-mediated query  $Q = (\mathcal{T}, q)$  with a description logic ontology  $\mathcal{T}$  and a conjunctive query  $q$ , determine the computational complexity of answering  $Q$  over any given input data instance  $\mathcal{A}$  and, if possible, reduce the task of finding certain answers to  $Q$  over  $\mathcal{A}$  to the task of evaluating a conventional database query with optimal complexity over  $\mathcal{A}$ .

Answering various types of queries mediated by a description logic (DL) ontology has been known as an important reasoning problem in knowledge representation since the early 1990s [1]. The proliferation of DLs and their applications [2, 3], the development of the (DL-underpinned) Web Ontology Language OWL<sup>1</sup>, and especially the paradigm of ontology-based data access (OBDA) [4, 5, 6] (proposed in the mid 2000s and recently nicknamed the virtual knowledge graph paradigm (VKG) [7]), have made theory and practice of answering ontology-mediated queries (OMQs) a hot research area lying at the crossroads of Knowledge Representation and Reasoning, Semantic Technologies and the Semantic Web, Knowledge Graphs, and Database Theory and Technologies.

In a nutshell, the idea underlying OBDA is as follows. The users of an OBDA system (such as Mastro<sup>2</sup> or Ontop<sup>3</sup>) may assume that the data they want to query is given in the form of a directed graph whose nodes are labelled with concepts (unary predicates or classes) and edges with roles (binary predicates or properties)—even though, in reality, the data can be physically stored in different and possibly heterogeneous data sources—hence the moniker VKG. The concept and role labels come from an ontology, designed by a domain expert, and should be familiar to the

---

*Email addresses:* ogerasimova@hse.ru (Olga Gerasimova), staskikotx@gmail.com (Stanislav Kikot), agi.kurucz@kc1.ac.uk (Agi Kurucz), podolskii@mi.ras.ru (Vladimir Podolskii), michael@dcs.bbk.ac.uk (Michael Zakharyashev)

<sup>1</sup><https://www.w3.org/TR/owl2-overview/>

<sup>2</sup><https://www.obdasystems.com>

<sup>3</sup><https://ontopic.biz>

intended users, who do not have to know anything about the real data sources. Apart from providing a user-friendly vocabulary for queries and a high-level conceptual view of the data, an important role of the ontology is to enrich possibly incomplete data with background knowledge. To illustrate, imagine that we are interested in the life of ‘scientists’ and would like to satisfy our curiosity by querying the data available on the Web (it may come from the universities’ databases, publishing companies, social networks, etc.). An ontology  $\mathcal{O}$  about scientists, provided by an OBDA system, might contain the following ‘axioms’ (given, for readability, both as DL concept inclusions and first-order sentences):

$$\text{BritishScientist} \sqsubseteq \exists \text{affiliatedWith}.\text{UniversityInUK} \quad (1)$$

$$\forall x [\text{BritishScientist}(x) \rightarrow \exists y (\text{affiliatedWith}(x, y) \wedge \text{UniversityInUK}(y))]$$

$$\exists \text{worksOnProject} \sqsubseteq \text{Scientist} \quad (2)$$

$$\forall x [\exists y \text{worksOnProject}(x, y) \rightarrow \text{Scientist}(x)]$$

$$\text{Scientist} \sqcap \exists \text{affiliatedWith}.\text{UniversityInUK} \sqsubseteq \text{BritishScientist} \quad (3)$$

$$\forall x [(\text{Scientist}(x) \wedge \exists y (\text{affiliatedWith}(x, y) \wedge \text{UniversityInUK}(y))) \rightarrow \text{BritishScientist}(x)]$$

$$\text{BritishScientist} \sqsubseteq \text{Brexiteer} \sqcup \text{Remainer} \quad (4)$$

$$\forall x [\text{BritishScientist}(x) \rightarrow (\text{Brexiteer}(x) \vee \text{Remainer}(x))]$$

Now, to find, for example, British scientists, we could execute a simple OMQ  $\mathcal{Q} = (\mathcal{O}, \mathbf{q})$  with the query

$$\mathbf{q}(x) = \text{BritishScientist}(x)$$

mediated by the ontology  $\mathcal{O}$ . The OBDA system is expected to return the members of the concept *BritishScientist* that are extracted from the original datasets by ‘mappings’ (database queries connecting the data with the ontology vocabulary) and also deduced from the data and axioms in  $\mathcal{O}$  such as (3). It is this latter reasoning task that makes OMQ answering non-trivial and potentially intractable both in practice and from the complexity-theoretic standpoint.

To ensure theoretical and practical tractability, the OBDA paradigm presupposes that the users’ OMQs are reformulated—or rewritten—by the OBDA system into conventional database queries over the original data sources, which have proved to be quite efficiently evaluated by the existing database management systems. Whether or not such a rewriting is possible and into which target query language naturally depends on the OMQ in question. One way to *uniformly* guarantee the desired rewritability is to delimit the language for OMQ ontologies and queries. Thus, the *DL-Lite* family of description logics [5] and the *OWL 2 QL* profile<sup>4</sup> of *OWL 2* were designed to guarantee rewritability of all OMQs with a *DL-Lite* ontology and a conjunctive query (CQ) into first-order (FO) queries, that is, essentially SQL queries [8]. In complexity-theoretic terms, FO-rewritability of an OMQ means that it can be answered in  $\text{LogTime}$  uniform  $\text{AC}^0$ , one of the smallest complexity classes. In our example above, (1) and (2) are the only axioms allowed by *OWL 2 QL*. Any OMQ with an  $\mathcal{EL}$ , *OWL 2 EL* or *hornSHIQ* ontology is datalog-rewritable [9], and so can be answered in P—polynomial time in the size of the data [10, 11]. Axioms (1)–(3) are admitted by the  $\mathcal{EL}$  syntax. On the other hand, OMQs with an  $\mathcal{ALC}$  (a notational variant of the multimodal logic  $\mathbf{K}_n$  [12]) or Schema.org ontology and a CQ are in general coNP-hard [1], and so regarded as intractable and not suitable for OBDA. For example, coNP-hard is the OMQ  $(\{(4)\}, \mathbf{q}_1)$  with the CQ

$$\mathbf{q}_1 = \exists w, x, y, z [\text{Brexiteer}(w) \wedge \text{hasCoAuthor}(w, x) \wedge \text{Remainer}(x) \wedge \text{hasCoAuthor}(x, y) \wedge \text{Brexiteer}(y) \wedge \text{hasCoAuthor}(y, z) \wedge \text{Remainer}(z)].$$

For various reasons, many existing ontologies do not comply with the restrictions imposed by the standard languages for OBDA. Notable examples include the large-scale medical ontology SNOMED CT<sup>5</sup>, which is mostly but not entirely in  $\mathcal{EL}$ , and the oil and gas NPD FactPages<sup>6</sup> ontology, which falls outside *OWL 2 QL* by a whisker. One way to resolve

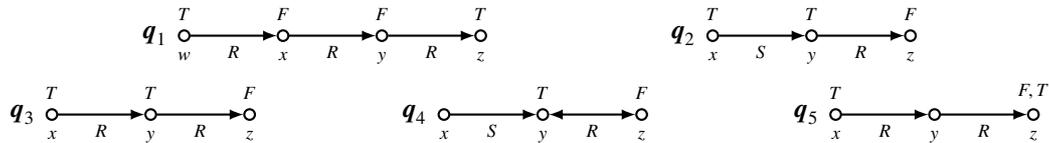
<sup>4</sup><https://www.w3.org/TR/owl2-profiles/>

<sup>5</sup><https://bioportal.bioontology.org/ontologies/SNOMEDCT>

<sup>6</sup><https://factpages.npd.no>

this issue is to compute an approximation of a given ontology within the required ontology language, which is an interesting and challenging reasoning problem by itself; see, e.g., [13, 14, 15, 16] and references therein.

There is a *non-uniform* alternative to the uniform approach (which delimits the language for OMQ ontologies and queries in order to make a general rewriting algorithm possible). One can allow OMQs in an expressive language, but at the same time provide the OBDA system with an algorithm that is capable of deciding the data complexity of each given OMQ, and (when it is possible) rewriting it to an equivalent database query of this optimal complexity. For example, while answering the OMQ  $(\{(4)\}, q_1)$  is coNP-complete, one can show that  $(\{(4)\}, q_2)$  with the same ontology and the CQ  $q_2$  shown in the picture below is P-complete and datalog-rewritable,  $(\{(4)\}, q_3)$  is NL- (non-deterministic logarithmic space) complete and linear-datalog-rewritable,  $(\{(4)\}, q_4)$  is L- (logarithmic space) complete and symmetric-datalog-rewritable, while  $(\{(4)\}, q_5)$  is in  $AC^0$  and FO-rewritable:



In the picture,  $F(u)$  stands for *Brexit*( $u$ ),  $T(u)$  stands for *Remainer*( $u$ ),  $R(u, v)$  for *hasCoAuthor*( $u, v$ ),  $S(u, v)$  for *hasBoss*( $x, y$ ), and all the variables  $w, x, y, z$  are assumed to be existentially quantified. (And yes, there exist British scientists who are Brexiteers in one aspect of life and Remainers in some other.) As another example, we refer to the experiments with the NPD FactPages ontology used for testing OBDA in industry [17, 18]. Although the ontology contains covering axioms of the form  $A \sqsubseteq B_1 \sqcup \dots \sqcup B_n$  not allowed in *OWL 2 QL*, one can show that the concrete queries provided by the industrial end-users do not ‘feel’ those dangerous axioms and are FO-rewritable.

Thus, in an ideal OBDA scenario, we would like the OBDA system to be able to recognise automatically the data complexity of any given OMQ and, whenever possible, rewrite it to a target query language with optimal complexity. The problem of determining the non-uniform data complexity and rewritability of OMQs was first systematically considered by Lutz and Wolter [19] for individual DL ontologies with varying CQs and by Bienvenu et al. [20] for individual OMQs (see also [21, 22]). In particular, the latter found a connection of OMQs to non-uniform constraint satisfaction problems (CSPs) with a fixed template [23] and used it to show that deciding FO- and datalog-rewritability of OMQs with an ontology in any DL between  $\mathcal{ALC}$  and  $\mathcal{SHIU}$  and an atomic query is  $NEXP$ -complete. The Feder-Vardi dichotomy of CSPs [24, 25] implies a P/coNP dichotomy of such OMQs, which is decidable in  $NEXP$ . For OMQs with an  $\mathcal{ALCI}$  ontology (that is,  $\mathcal{ALC}$  with inverse binary relations) and a CQ, deciding FO-rewritability rises and becomes  $2NEXP$ -complete; deciding whether such an OMQ is in P for data complexity is also  $2NEXP$ -complete [26, 27]. For OMQs with an  $\mathcal{EL}$  ontology and a CQ, Lutz and Sabellek [28] established a trichotomy according to which each OMQ is either in  $AC^0$  and FO-rewritable, or NL-complete and linear-datalog-rewritable, or P-complete and datalog-rewritable; deciding membership in this trichotomy is  $EXP$ -complete.

It should be also noted that, in the context of datalog and deductive databases, a similar 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 (i.e., intensional) predicates [29] and single rule programs [30, 31],  $2EXP$ -complete for monadic programs [32, 33], and  $PSPACE$ -complete for linear monadic programs [32]. Considerable efforts have been made to understand linearisability of datalog programs ensuring evaluation in NL [34, 35, 36, 37, 38], and datalog rewritability of disjunctive datalog programs [39].

To sum up, the general problem of recognising the data complexity of OMQs in standard DLs and the types of their rewritability turns out to be computationally very hard. Moreover, in spite of numerous attempts, very few practically useful partial algorithms or easily checkable syntactic conditions have been discovered so far. The natural idea [20] of using the connection with CSPs and taking advantage of algorithms and techniques developed for checking their complexity has not succeeded either: as reported in [40], the Polyanna program [41], designed to check tractability of CSPs, failed to recognise coNP-hardness of the very simple OMQ  $(\{(4)\}, q_1)$  above because the reduction to CSPs is unavoidably exponential and Polyanna simply ran out of memory.

*Our contribution.* In this paper, we contribute to the non-uniform approach, but we take a different direction. Rather than considering arbitrary ontologies in an expressive DL and providing a general complexity analysis, we single

out and fix one fundamental source of intractability in OMQ answering—the basic *covering axiom*  $A \sqsubseteq F \sqcup T$  with concept names (unary predicates)  $A, F, T$ —and investigate how the interplay between this axiom and the structure of the Boolean CQs  $q$  in the OMQs

$$\mathcal{Q} = (\text{cov}_A, q), \quad \text{with} \quad \text{cov}_A = \{A \sqsubseteq F \sqcup T\} \quad (5)$$

determines the computational behaviour of  $\mathcal{Q}$ . As the seemingly trivial OMQs  $(\{(4)\}, q_1)$ – $(\{(4)\}, q_4)$  above indicate, this interplay can be pretty subtle even in the case of Boolean path queries. From the practical point of view, covering (or union) constraints are indispensable to conceptual modelling [42]; for example, in Schema.org they are used to represent disjunctive property domains and ranges [43]. The first attempt to understand the complexity of answering OMQs with arbitrary Schema.org ontologies and unions of CQs (UCQs) was made by Hernich et al. [44].

We obtain a series of results on the complexity of answering OMQs of the form (5) 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  $\forall\exists\text{SAT}$  and the acyclicity problem for succinctly represented graphs. In particular,

- we show that, in general, answering OMQs (5) is  $\Pi_2^p$ -complete for combined complexity (in the size of  $q$  and the data), that is, harder than answering *DL-Lite* and  $\mathcal{EL}$  OMQs (unless  $\text{NP} = \Pi_2^p$ , and so  $\text{NP} = \text{PSPACE}$ );
- we prove that the problem of determining FO-rewritability of these OMQs is even harder, namely  $\text{PSPACE}$ -hard in the size of  $q$ , which indicates that a general syntactic classification of CQs  $q$  according to the data complexity of answering  $\mathcal{Q}$  and the type of its rewritability will be extremely difficult to find. This result is quite surprising in comparison with the  $\text{PSPACE}$ -hardness proofs for boundedness of linear monadic datalog programs [32] and FO-rewritability of OMQs with Schema.org ontologies and UCQs [44], 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 use just a single dag-shaped CQ to encode satisfying evaluations of a Boolean formula and apply a reduction of the acyclicity problem for the graph succinctly represented by this formula.

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.

- 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  $x$ , like the first four CQs depicted above) gives rise to an FO-rewritable OMQ (i.e., in  $\text{AC}^0$ ) if it does not contain occurrences of one of  $F$  or  $T$ ; otherwise the OMQ is L-hard and even NL-hard for path CQs. This simple criterion fails for CQs with twins, where the problem of finding a syntactic characterisation might turn 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 [28] gave a nice semantic criterion for OMQs with an  $\mathcal{EL}$  ontology. We combine their ideas with the automata-theoretic technique of Cosmadakis et al. [32] and prove a useful sufficient semantic condition for our OMQs to be linear-datalog-rewritable (and so in NL).
- We use some of these conditions to obtain the main result of this paper: a complete and transparent syntactic  $\text{AC}^0/\text{NL}/\text{P}/\text{coNP}$  and rewritability tetrachotomy of the OMQs (5) with a path CQ  $q$  that do not contain *FT*-twins. The latter restriction is redundant if the ontology is extended with the disjointness axiom  $F \sqcap T \sqsubseteq \perp$  (making covering axiom (4) a ‘British scientist’s dilemma’). We show that (i) such CQs  $q$  without occurrences of  $F$  (or  $T$ ) and only them give rise to FO-rewritable OMQs  $\mathcal{Q}$  (in  $\text{AC}^0$ ), that (ii)  $\mathcal{Q}$  is linear-datalog-rewritable and NL-complete just in case  $q$  has a certain periodic structure, and prove that (iii) otherwise  $\mathcal{Q}$  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  $F$ s and at least two  $T$ s, and only them give rise to  $\text{coNP}$ -hard OMQs.

*Structure of the paper.* In the next section, we provide the necessary background definitions. In Section 3, we first prove that answering our OMQs is  $\Pi_2^p$ -complete for combined complexity, and then obtain a few simple and general

results on OMQs in  $AC^0$ , L, and P. We also introduce a handy semantic construction for certain OMQs, which is similar to datalog expansion [45] and, in our context, called ‘growing cactuses’. Then, in Section 4, we use cactuses to show that deciding FO-rewritability of an OMQ is PSPACE-hard. Section 5 gives a sufficient semantic condition of linear-datalog-rewritability in terms of cactuses. Finally, in Section 6, we obtain a complete classification of OMQs with a path CQ without twins according to their data complexity and rewritability type. Future research and open problems are discussed in the concluding Section 7.

An extended abstract with some of the results from this article has been presented at the 17th International Conference on Principles of Knowledge Representation and Reasoning.

## 2. Preliminaries

Using the standard description logic syntax and semantics [3], we consider *ontology-mediated queries* (OMQs) of the form  $Q = (\mathcal{T}, q)$ , where  $\mathcal{T}$  is one of the two ontologies

$$\text{cov}_A = \{A \sqsubseteq F \sqcup T\}, \quad \text{cov}_A^\perp = \{A \sqsubseteq F \sqcup T, F \sqcap T \sqsubseteq \perp\}$$

(in which we sometimes set  $A = \top$ ) and  $q$  is a *Boolean conjunctive query* (CQ), i.e., an FO-sentence  $q = \exists x \varphi(x)$ , in which  $\varphi$  is a conjunction of atoms with variables from  $x$ . We often think of  $q$  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  $F(x), T(x) \in q$  are referred to as *FT-twins* in  $q$ . An *ABox* (data instance),  $\mathcal{A}$ , is a finite set of ground atoms with unary or binary predicates. We denote by  $\text{ind}(\mathcal{A})$  the set of constants (individuals) in  $\mathcal{A}$ . An *interpretation* is a structure of the form  $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$  with a domain  $\Delta^{\mathcal{I}} \neq \emptyset$  and an interpretation function  $\cdot^{\mathcal{I}}$  such that  $a^{\mathcal{I}} \in \Delta^{\mathcal{I}}$  for any constant  $a$ ,  $\top^{\mathcal{I}} = \Delta^{\mathcal{I}}$ ,  $\perp^{\mathcal{I}} = \emptyset$ ,  $P^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}}$  for any unary predicate  $P$ , and  $P^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$  for any binary  $P$ . The interpretation  $\mathcal{I}$  is a *model* of  $\mathcal{T}$  if  $A^{\mathcal{I}} \subseteq F^{\mathcal{I}} \cup T^{\mathcal{I}}$  and, for  $\mathcal{T} = \text{cov}_A^\perp$ , also  $F^{\mathcal{I}} \cap T^{\mathcal{I}} = \emptyset$ ; it is a *model* of  $\mathcal{A}$  if  $P(a) \in \mathcal{A}$  implies  $a^{\mathcal{I}} \in P^{\mathcal{I}}$  and  $P(a, b) \in \mathcal{A}$  implies  $(a^{\mathcal{I}}, b^{\mathcal{I}}) \in P^{\mathcal{I}}$ . The truth-relation  $\mathcal{I} \models q$  is defined as usual in FO-logic.

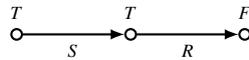
It is often convenient to regard CQs, ABoxes and interpretations as digraphs with labelled edges and partially labelled nodes (by  $F, T$  in CQs and  $F, T, A$  in ABoxes and interpretations). Without loss of generality, we assume that these graphs are *connected* as undirected graphs. A *path CQ* is a (simple) directed path each of whose edges is labelled by a single binary predicate.

The *certain answer* to an OMQ  $Q = (\mathcal{T}, q)$  over an ABox  $\mathcal{A}$  is ‘yes’ if  $\mathcal{I} \models q$  for all models  $\mathcal{I}$  of  $\mathcal{T}$  and  $\mathcal{A}$ —in which case we write  $\mathcal{T}, \mathcal{A} \models q$ —and ‘no’ otherwise.

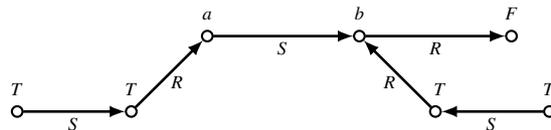
A *minimal model* of  $\mathcal{T}$  and  $\mathcal{A}$  is obtained from  $\mathcal{A}$  by adding to each ‘undecided’  $A$ -node (which is labelled by neither  $F$  nor  $T$ ) exactly one of  $F$  or  $T$  as label. Clearly,  $\mathcal{T}, \mathcal{A} \models q$  iff  $\mathcal{I} \models q$  for every minimal model  $\mathcal{I}$  of  $\mathcal{T}$  and  $\mathcal{A}$ . So, from now on, ‘model’ means ‘minimal model’. Finally, we note that  $\mathcal{I} \models q$  iff there is a digraph homomorphism  $h: q \rightarrow \mathcal{I}$  preserving the labels of nodes and edges.

The following example illustrates the ‘reasoning’ required to answer OMQs, which amounts to a ‘proof by cases’.

**Example 1.** Consider the OMQ  $Q = (\text{cov}_\top, q)$  with  $A = \top$  and the path CQ  $q$  shown in the picture below:



By analysing the four possible cases for  $a, b \in F^{\mathcal{I}}, T^{\mathcal{I}}$  in an arbitrary model  $\mathcal{I}$  of  $\text{cov}_\top$  and the ABox below, one can readily show that the certain answer to  $Q$  over this ABox is ‘yes’.



Indeed, if  $a^{\mathcal{I}} \in F^{\mathcal{I}}$ , then  $q$  is homomorphically embeddable into the  $S$ – $R$  path on the left-hand side of  $\mathcal{I}$ . Otherwise  $a^{\mathcal{I}} \in T^{\mathcal{I}}$ . If  $b^{\mathcal{I}} \in F^{\mathcal{I}}$ , then  $q$  is homomorphically embeddable into the  $S$ – $R$  path on the right-hand side of  $\mathcal{I}$ . In the remaining case  $b^{\mathcal{I}} \in T^{\mathcal{I}}$ , there is a homomorphism from  $q$  into the  $S$ – $R$  path on the top of  $\mathcal{I}$ .

Such proofs can be given as resolution refutations (derivations of the empty clause) in clausal logic.

**Example 2.** The certain answer to an OMQ  $\mathcal{Q} = (\mathcal{T}, \mathbf{q})$  over an ABox  $\mathcal{A}$  is ‘yes’ iff the following set  $\mathcal{S}_{\mathcal{Q}}$  of clauses is unsatisfiable:

$$\mathcal{S}_{\mathcal{Q}} = \{\neg A(x) \vee F(x) \vee T(x), \bigvee_{P(x) \in \mathbf{q}} \neg P(x)\} \cup \mathcal{A}.$$

If  $\mathcal{T} = \text{cov}_A^\perp$  then  $\mathcal{S}_{\mathcal{Q}}$  also contains the clause  $\neg F(x) \vee \neg T(x)$ . In other words, the certain answer to  $\mathcal{Q}$  over  $\mathcal{A}$  is ‘yes’ iff there is a resolution refutation for  $\mathcal{S}_{\mathcal{Q}}$  in the classical resolution calculus [46].

Our concern is the *combined* and *data complexity* of deciding, for a given OMQ  $\mathcal{Q} = (\mathcal{T}, \mathbf{q})$  and an ABox  $\mathcal{A}$ , whether  $\mathcal{T}, \mathcal{A} \models \mathbf{q}$ . In the former case,  $\mathbf{q}$  and  $\mathcal{A}$  are regarded as input; in the latter one,  $\mathbf{q}$  is fixed. It should be clear that  $\Pi_2^p = \text{coNP}^{\text{NP}}$  is an upper bound for the combined complexity of our problem, which amounts to checking that, for every model  $\mathcal{I}$  of  $\mathcal{T}$  and  $\mathcal{A}$ , there exists a homomorphism  $\mathbf{q} \rightarrow \mathcal{I}$ , with the latter being NP-complete. For data complexity, that is, when  $\mathbf{q}$  is fixed, checking the existence of a homomorphism  $\mathbf{q} \rightarrow \mathcal{I}$  can be done in P, and so the whole problem is in coNP.

We are also interested in various types of rewritability of OMQs. An OMQ  $\mathcal{Q} = (\mathcal{T}, \mathbf{q})$  is called *FO-rewritable* if there is an FO-sentence  $\Phi$  such that  $\mathcal{T}, \mathcal{A} \models \mathbf{q}$  iff  $\Phi$  is true in the structure  $\mathcal{A}$ . In terms of circuit complexity, FO-rewritability is equivalent to answering  $\mathcal{Q}$  in logtime-uniform  $\text{AC}^0$  [47]. Note that if  $\mathbf{q}$  contains *FT-twins*, then  $\exists x(F(x) \wedge T(x))$  is an FO-rewriting of  $\mathcal{Q} = (\text{cov}_A^\perp, \mathbf{q})$  because the certain answer to  $\mathcal{Q}$  over  $\mathcal{A}$  is ‘yes’ iff  $\text{cov}_A^\perp$  is inconsistent with  $\mathcal{A}$ , which can only happen when  $\mathcal{A}$  contains an *FT-twin*.

Recall from, say [8], that a *datalog program*,  $\Pi$ , is a finite set of *rules* of the form  $\forall \mathbf{x}(\gamma_0 \leftarrow \gamma_1 \wedge \dots \wedge \gamma_m)$ , where each  $\gamma_i$  is a (constant- and function-free) atom  $Q(\mathbf{y})$  with  $\mathbf{y} \subseteq \mathbf{x}$ . (As usual, we omit  $\forall \mathbf{x}$ .) The atom  $\gamma_0$  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 called *IDB predicates*, the rest *EDB predicates*. A *datalog query* in this paper takes the form  $(\Pi, \mathbf{G})$  with a 0-ary atom  $\mathbf{G}$ . The *answer* to  $(\Pi, \mathbf{G})$  over an ABox  $\mathcal{A}$  is ‘yes’ if  $\mathbf{G}$  is true in the structure  $\Pi(\mathcal{A})$  obtained by closing  $\mathcal{A}$  under the rules in  $\Pi$ , in which case we write  $\Pi, \mathcal{A} \models \mathbf{G}$ . We call  $(\Pi, \mathbf{G})$  a *datalog-rewriting* of an OMQ  $\mathcal{Q} = (\mathcal{T}, \mathbf{q})$  in case  $\mathcal{T}, \mathcal{A} \models \mathbf{q}$  iff  $\Pi, \mathcal{A} \models \mathbf{G}$ , for any ABox  $\mathcal{A}$  containing only EDB predicates of  $\Pi$ . If  $\mathcal{Q}$  is datalog-rewritable, then it can be answered in P for data complexity [48]; if there is a rewriting to a  $(\Pi, \mathbf{G})$  with a *linear* program  $\Pi$ , having at most one IDB predicate in the body of each of its rules, then  $\mathcal{Q}$  can be answered in NL (non-deterministic logarithmic space). The NL upper bound also holds for datalog queries with a linear-stratified program, which is defined as follows. A *stratified* program [8] is a sequence  $\Pi = (\Pi_0, \dots, \Pi_n)$  of datalog programs, called the *strata* of  $\Pi$ , such that each predicate in  $\Pi$  can occur in the head of a rule only in one stratum  $\Pi_i$  and can occur in the body of a rule only in strata  $\Pi_j$  with  $j \geq i$ . If, in addition, the body of each rule in  $\Pi$  contains at most one occurrence of a head predicate from the same stratum,  $\Pi$  is called *linear-stratified*. Every linear-stratified program can be converted to an equivalent linear datalog program [38], and so datalog queries with a linear-stratified program can be answered in NL for data complexity.

### 3. Initial Observations

In this section, we obtain a number of relatively simple complexity and rewritability results that are applicable to arbitrary (not necessarily path) CQs  $\mathbf{q}$ . By writing  $\mathcal{Q} = (\mathcal{T}, \mathbf{q})$  we mean ‘any  $\mathcal{T} \in \{\text{cov}_A, \text{cov}_A^\perp\}$ ’.

#### 3.1. Combined Complexity

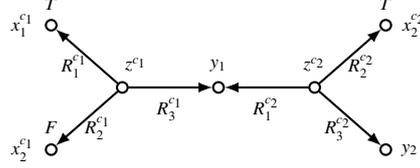
Our first result pushes to the limit [44, Theorem 5] according to which answering OMQs with Schema.org ontologies is  $\Pi_2^p$ -complete for combined complexity (the proof of that theorem uses an ontology with an enumeration definition  $E = \{0, 1\}$  and additional concept names, none of which is available in our case).

**Theorem 3.** (i) Answering OMQs  $(\mathcal{T}, \mathbf{q})$  is  $\Pi_2^p$ -complete for combined complexity.

(ii) Answering OMQs  $(\mathcal{T}, \mathbf{q})$  with a tree-shaped (or path) CQ  $\mathbf{q}$  is coNP-complete for combined complexity.

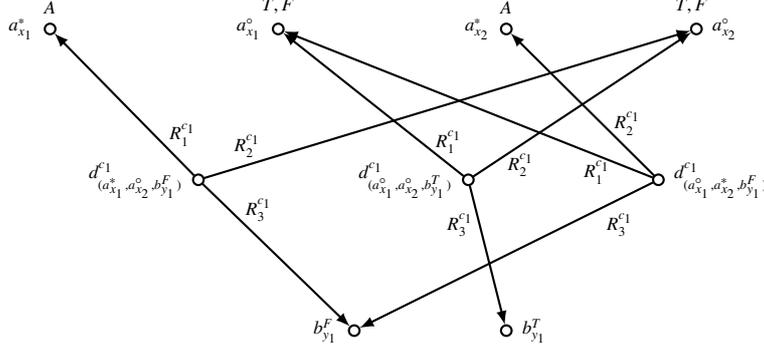
*Proof.* Deciding whether  $\mathcal{T}, \mathcal{A} \models \mathbf{q}$  can be done by a coNP Turing machine (checking all models  $\mathcal{I}$  of  $\mathcal{T}$  and  $\mathcal{A}$ ) with an NP-oracle (checking the existence of  $h: \mathbf{q} \rightarrow \mathcal{I}$ ); for tree-shaped  $\mathbf{q}$ , a P-oracle is enough. The lower bound in (ii) follows from Theorem 21. For (i), we prove it by reduction of  $\Pi_2^p$ -complete  $\forall \exists \text{SAT}$  [49].

Let  $\psi(x, y)$  be a 3CNF with propositional variables  $x$  and  $y$  and let  $\varphi = \forall x \exists y \psi(x, y)$ . We assume that each literal contains each variable at most once. Denote by  $\mathbf{q}_\varphi$  the CQ that, for each clause  $c = \ell_1 \vee \ell_2 \vee \ell_3$  in  $\psi$ , contains atoms  $R_i^c(z^c, u_i^c)$ ,  $i = 1, 2, 3$ , with  $u_i^c = y$  if  $y \in \mathbf{y}$  is in  $\ell_i$  and  $u_i^c = x^c$  if  $x \in \mathbf{x}$  is in  $\ell_i$ ; in the latter case,  $\mathbf{q}_\varphi$  also contains  $T(x^c)$  if  $\ell_i = x$  and  $F(x^c)$  if  $\ell_i = \neg x$ . For example, clauses  $c_1 = x_1 \vee \neg x_2 \vee y_1$  and  $c_2 = \neg y_1 \vee x_2 \vee y_2$  contribute the following atoms to  $\mathbf{q}_\varphi$ :



For  $\text{cov}_A$ , the ABox  $\mathcal{A}_\varphi$  is defined as follows. For  $x \in \mathbf{x}$ , we take individuals  $a_x^*$  and  $a_x^\circ$  and, for  $y \in \mathbf{y}$ , individuals  $b_y^F$  and  $b_y^T$ .  $\mathcal{A}_\varphi$  comprises the atoms  $A(a_x^*)$ ,  $F(a_x^\circ)$ ,  $T(a_x^\circ)$ , for  $x \in \mathbf{x}$ . For each  $c = \ell_1 \vee \ell_2 \vee \ell_3$ , 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  $x \in \mathbf{x}$  is in  $\ell_i$ , (ii)  $e_i = b_y^\nu$  for some  $\nu \in \{F, T\}$  whenever  $y \in \mathbf{y}$  is in  $\ell_i$ , 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  $\ell_i$  true. Now, for each  $c$  and each  $t = (e_1, e_2, e_3)$  in  $E^c$ , we take a fresh individual  $d_t^c$  (the *centre* of the pair  $(c, t)$ ), and add three atoms  $R_i^c(d_t^c, e_i)$ ,  $i = 1, 2, 3$ , to  $\mathcal{A}_\varphi$ .

For  $\text{cov}_A^+$ , we take  $a_x^F$  and  $a_x^T$  instead of each  $a_x^\circ$ , add the atoms  $F(a_x^F)$ ,  $T(a_x^T)$  instead of  $F(a_x^\circ)$ ,  $T(a_x^\circ)$ , and replace item (i) in the definition of  $E^c$  with (i)'  $e_i = a_x^\mu$  for some  $\mu \in \{*, F, T\}$  whenever  $x \in \mathbf{x}$  is in  $\ell_i$ ,



The number of atoms in  $\mathcal{A}_\varphi$  is polynomial in the size of  $\varphi$ .

**Lemma 3.1.** Suppose  $\alpha: \mathbf{x} \rightarrow \{F, T\}$  is any assignment and

$$\mathcal{A}_\varphi^\alpha = \mathcal{A}_\varphi \cup \{T(a_x^*) \mid \alpha(x) = T\} \cup \{F(a_x^*) \mid \alpha(x) = F\}.$$

There exists  $\beta: \mathbf{y} \rightarrow \{F, T\}$  such that  $\psi(\alpha(\mathbf{x}), \beta(\mathbf{y}))$  is true iff  $\mathcal{A}_\varphi^\alpha \models \mathbf{q}_\varphi$ .

*Proof.* ( $\Rightarrow$ ) Suppose  $\beta$  is such that  $\psi(\alpha(\mathbf{x}), \beta(\mathbf{y}))$  is true. We need to show that there is a homomorphism  $h: \mathbf{q}_\varphi \rightarrow \mathcal{A}_\varphi^\alpha$ .

*Case  $\text{cov}_A$ :* For every clause  $c = \ell_1 \vee \ell_2 \vee \ell_3$  in  $\psi$  and for all  $i = 1, 2, 3$ , we define  $e_i^c$  as follows. We let (i)  $e_i^c = a_x^*$  if  $x \in \mathbf{x}$  is in  $\ell_i$  and  $\alpha$  makes  $\ell_i$  true, (ii)  $e_i^c = a_x^\circ$  if  $x \in \mathbf{x}$  is in  $\ell_i$  and  $\alpha$  makes  $\ell_i$  false, and (iii)  $e_i^c = b_y^{\beta(y)}$  if  $y \in \mathbf{y}$  is in  $\ell_i$ . As  $\psi(\alpha(\mathbf{x}), \beta(\mathbf{y}))$  is true,  $(e_1^c, e_2^c, e_3^c)$  is in  $E^c$ . Then we define a map  $h$  by taking  $h(z^c)$  to be the centre of  $(c, (e_1^c, e_2^c, e_3^c))$  and  $h(u_i^c) = e_i^c$ . It follows from the construction that  $h$  is well-defined and a homomorphism from  $\mathbf{q}_\varphi$  to  $\mathcal{A}_\varphi$  with respect to the binary atoms. We show that it preserves the unary atoms as well. Indeed, for each  $c$  and each  $x \in \mathbf{x}$  occurring in  $c$ , there are two cases: (1) If  $x^c$  is labelled by  $T$  in  $\mathbf{q}_\varphi$ , then  $\ell_i = x$ . So if  $\alpha$  makes  $\ell_i$  true, then  $e_i^c = a_x^*$  is labelled by  $T$  in  $\mathcal{A}_\varphi^\alpha$ . And if  $\alpha$  makes  $\ell_i$  false, then  $e_i^c = a_x^\circ$  is labelled by both  $T$  and  $F$  in  $\mathcal{A}_\varphi^\alpha$ . (2) If  $x^c$  is labelled by  $F$  in  $\mathbf{q}_\varphi$ , then  $\ell_i = \neg x$ . So if  $\alpha$  makes  $\ell_i$  true, then  $e_i^c = a_x^*$  is labelled by  $F$  in  $\mathcal{A}_\varphi^\alpha$ . And if  $\alpha$  makes  $\ell_i$  false, then  $e_i^c = a_x^\circ$  is labelled by both  $T$  and  $F$  in  $\mathcal{A}_\varphi^\alpha$ .

*Case  $\text{cov}_A^+$ :* In the definition of  $e_i^c$ , we replace (ii) with (ii)'  $e_i^c = a_x^T$  if  $\ell_i = x$  for some  $x \in \mathbf{x}$  and  $\alpha(x) = F$ , and (ii)''  $e_i^c = a_x^F$  if  $\ell_i = \neg x$  for some  $x \in \mathbf{x}$  and  $\alpha(x) = T$ . Again, we claim that  $h$  as defined above preserves the unary atoms. Indeed, for each  $c$  and for each  $x \in \mathbf{x}$  occurring in  $c$ , there are two cases: (1) If  $x^c$  is labelled by  $T$  in  $\mathbf{q}_\varphi$ , then  $\ell_i = x$ . So if  $\alpha$  makes  $\ell_i$  true, then  $e_i^c = a_x^*$  is labelled by  $T$  in  $\mathcal{A}_\varphi^\alpha$ . And if  $\alpha$  makes  $\ell_i$  false, then  $e_i^c = a_x^T$  is labelled by

$T$  in  $\mathcal{A}_\varphi^a$ . (2) If  $x^c$  is labelled by  $F$  in  $\mathbf{q}_\varphi$ , then  $\ell_i = \neg x$ . So if  $a$  makes  $\ell_i$  true, then  $e_i^c = a_x^*$  is labelled by  $F$  in  $\mathcal{A}_\varphi^a$ . And if  $a$  makes  $\ell_i$  false, then  $e_i^c = a_x^F$  is labelled by  $F$  in  $\mathcal{A}_\varphi^a$ .

( $\Leftarrow$ ) Suppose that  $h: \mathbf{q}_\varphi \rightarrow \mathcal{A}_\varphi^a$ . Then, for any  $y \in \mathbf{y}$ ,  $h(y) = b_y^v$  for some  $v \in \{F, T\}$ . We then set  $b(y) = v$ . We claim that  $\psi(a(x), b(y))$  is true. Indeed, for every clause  $c = \ell_1 \vee \ell_2 \vee \ell_3$  in  $\psi$ , there is  $t = (e_1, e_2, e_3) \in E^c$  such that  $h$  maps the ‘contribution’ of  $c$  in  $\mathbf{q}_\varphi$  onto the ‘star’ with centre  $d_i^c$ . If  $t$  is in  $E^c$  because  $e_i = a_x^*$  for some  $i = 1, 2, 3$ ,  $x \in \mathbf{x}$ , then the label of  $a_x^*$  in  $\mathcal{A}_\varphi^a$  is  $a(x)$ . As  $h$  is a homomorphism, the label of  $x^c$  in  $\mathbf{q}_\varphi$  is also  $a(x)$ , and so  $a$  makes  $c$  true by the definition of  $\mathbf{q}_\varphi$ . And if  $t$  is in  $E^c$  because  $e_i = b_y^{b(y)}$  for some  $i = 1, 2, 3$ ,  $y \in \mathbf{y}$  and  $b(y)$  makes  $\ell_i$  true, then  $c$  is clearly true as well.  $\square$

Finally, we prove that  $\varphi$  is satisfiable iff  $\mathcal{T}, \mathcal{A}_\varphi \models \mathbf{q}_\varphi$  iff  $\mathcal{I} \models \mathbf{q}_\varphi$  for every model  $\mathcal{I}$  of  $\mathcal{T}$  and  $\mathcal{A}_\varphi$ . ( $\Rightarrow$ ) Given  $\mathcal{I}$ , define an assignment  $\alpha_{\mathcal{I}}: \mathbf{x} \rightarrow \{F, T\}$  by taking  $\alpha_{\mathcal{I}}(x) = T$  if  $a_x^* \in T^{\mathcal{I}}$  and  $\alpha_{\mathcal{I}}(x) = F$  if  $a_x^* \in F^{\mathcal{I}}$ . Then  $\mathcal{I} = \mathcal{A}_\varphi^{\alpha_{\mathcal{I}}}$ , and so we are done by Lemma 3.1. The implication ( $\Leftarrow$ ) also follows from Lemma 3.1, as  $\mathcal{A}_\varphi^a$  is a model of  $\mathcal{T}$  and  $\mathcal{A}_\varphi$ , for every assignment  $a: \mathbf{x} \rightarrow \{F, T\}$ .  $\square$

### 3.2. Data Complexity: $AC^0$ and L

We next focus on the *data* complexity of (answering) OMQs  $(\mathcal{T}, \mathbf{q})$ . If  $\mathbf{q}$  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). We call  $\mathbf{q}$  a  $0$ -CQ if it does not have a solitary  $F$  or a solitary  $T$ . Note that, for any twinless  $\mathbf{q}$ ,  $(\text{cov}_A, \mathbf{q})$  and  $(\text{cov}_A^\perp, \mathbf{q})$  have the same data complexity.

**Theorem 4.** (i) If  $\mathbf{q}$  is a  $0$ -CQ, then  $(\mathcal{T}, \mathbf{q})$  is in  $AC^0$ .

(ii) If  $\mathbf{q}$  is twinless and contains at least one solitary  $F$  and at least one solitary  $T$ , then  $(\text{cov}_T, \mathbf{q})$  and  $(\text{cov}_T^\perp, \mathbf{q})$ , and so  $(\text{cov}_A, \mathbf{q})$  and  $(\text{cov}_A^\perp, \mathbf{q})$  are L-hard.

*Proof.* (i) We show that  $\mathcal{T}, \mathcal{A} \models \mathbf{q}$  iff  $\mathcal{A} \models \mathbf{q}$ , and so  $\mathbf{q}$  is an FO-rewriting of  $(\mathcal{T}, \mathbf{q})$ . ( $\Rightarrow$ ) Suppose  $\mathcal{A} \not\models \mathbf{q}$  and  $\mathbf{q}$  has no solitary  $F$  (the other case is similar). 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  $\mathcal{A}' \not\models \mathbf{q}$ . ( $\Leftarrow$ ) is trivial.

(ii) The proof is by an FO-reduction of the L-complete reachability problem for undirected graphs. Denote by  $\mathbf{q}'$  the CQ obtained by gluing together all the  $T$ -nodes and by gluing together all the  $F$ -nodes in  $\mathbf{q}$ . Thus,  $\mathbf{q}'$  contains a single  $T$ -node,  $x$ , and a single  $F$ -node,  $y$ . Clearly, there is a homomorphism  $h: \mathbf{q} \rightarrow \mathbf{q}'$ . Let  $\mathbf{q}'' = \mathbf{q}' \setminus \{T(x), F(y)\}$ .

Suppose  $G = (V, E)$  is a graph with  $s, t \in V$ . We regard  $G$  as a directed graph such that  $(u, v) \in E$  iff  $(v, u) \in E$ , for any  $u, v \in V$ . Construct an ABox  $\mathcal{A}_G$  from  $G$  in the following way. Replace each edge  $e = (u, v) \in E$  by a copy  $\mathbf{q}_e''$  of  $\mathbf{q}''$  such that, in  $\mathbf{q}_e''$ , node  $x$  is renamed to  $u$ ,  $y$  to  $v$ , and all other nodes  $z$  to some fresh copy  $\mathfrak{z}_e$ . Then  $\mathcal{A}_G$  comprises all such  $\mathbf{q}_e''$ , for  $e \in E$ , as well as atoms  $T(s)$  and  $F(t)$ . We show that there is a path from  $s$  to  $t$  in  $G$  ( $s \rightarrow_G t$ , in symbols) iff  $\text{cov}_T, \mathcal{A}_G \models \mathbf{q}$  iff  $\text{cov}_T^\perp, \mathcal{A}_G \models \mathbf{q}$ .

( $\Rightarrow$ ) Suppose there is a path  $s = v_0, \dots, v_n = t$  in  $G$  with  $e_i = (v_i, v_{i+1}) \in E$ , for  $i < n$ . Consider an arbitrary model  $\mathcal{I}$  of  $\text{cov}_T$  and  $\mathcal{A}_G$ . Since  $\mathcal{I} \models \text{cov}_T$ , and  $T(s)$  and  $F(t)$  are in  $\mathcal{A}_G$ , we can find some  $i < n$  such that  $\mathcal{I} \models T(v_i)$  and  $\mathcal{I} \models F(v_{i+1})$ . As  $\mathbf{q}_{e_i}''$  is an isomorphic copy of  $\mathbf{q}''$ , we obtain  $\mathcal{I} \models \mathbf{q}''$ , and so  $\mathcal{I} \models \mathbf{q}$ .

( $\Leftarrow$ ) Suppose  $s \not\rightarrow_G t$ . Then, by the construction,  $t$  is not reachable from  $s$  in  $\mathcal{A}_G$  (not even via an undirected path). Define a model  $\mathcal{I}$  of  $\text{cov}_T^\perp$  and  $\mathcal{A}_G$  by taking  $T^{\mathcal{I}}$  to be the set of nodes in  $\mathcal{A}_G$  that are reachable from  $s$  (via an undirected path) and  $F^{\mathcal{I}}$  its complement. Clearly, no connected component of  $\mathcal{A}_G$  (as undirected graph) contains both  $T^{\mathcal{I}}$ - and  $F^{\mathcal{I}}$  nodes. Since  $\mathbf{q}$  is connected and contains at least one  $T$  and at least one  $F$ , it follows that  $\mathcal{I} \not\models \mathbf{q}$ .  $\square$

Theorem 4 (ii) is complemented by the following simple sufficient condition. Call a CQ  $\mathbf{q}'(x, y)$  with two free variables  $x$  and  $y$  *symmetric* if, for any ABox  $\mathcal{A}$  and  $a, b \in \text{ind}(\mathcal{A})$ , we have  $\mathcal{A} \models \mathbf{q}'(a, b)$  iff  $\mathcal{A} \models \mathbf{q}'(b, a)$ .

**Theorem 5.** Let  $\mathbf{Q} = (\mathcal{T}, \mathbf{q})$  be any OMQ with

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

for some CQs  $\mathbf{q}'(x, y)$ ,  $\mathbf{q}'_1(x)$  and  $\mathbf{q}'_2(y)$  that do not contain solitary  $T$  and  $F$ , and symmetric  $\mathbf{q}'(x, y)$ . Here we assume that  $\mathbf{q}'_1(x)$  and  $\mathbf{q}'_2(y)$  are disjoint, and that  $x$  and  $y$  are their only common variables with  $\mathbf{q}'(x, y)$ . Then  $\mathbf{Q}$  is in L.

*Proof.* It is not hard to show that, for any ABox  $\mathcal{A}$ , we have  $\mathcal{T}, \mathcal{A} \models \mathbf{q}$  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:

(S1)  $F(v_0), A(v_1), \dots, A(v_{n-1}), T(v_n) \in \mathcal{A}$ ,

(S2)  $\mathcal{A} \models \mathbf{q}'(v_i, v_{i+1})$ , for  $0 \leq i < n$ ,

(S3)  $\mathcal{A} \models \mathbf{q}'_1(v_i)$ , for  $0 \leq i < n$ ,

(S4)  $\mathcal{A} \models \mathbf{q}'_2(v_i)$ , for  $1 \leq i \leq n$ .

Indeed, suppose that there are  $v_0, v_1, \dots, v_n \in \text{ind}(\mathcal{A})$  such that (S1)–(S4) hold. Consider a model  $\mathcal{I}$  of  $\mathcal{T}$  and  $\mathcal{A}$ . By discrete continuity, there must be  $i$  such that both  $F(v_i)$  and  $T(v_{i+1})$  are true in  $\mathcal{I}$ . Now, (S2)–(S4) guarantee that  $\mathcal{I} \models \mathbf{q}$ . Conversely, suppose there are no  $v_0, v_1, \dots, v_n$  in  $\text{ind}(\mathcal{A})$  that satisfy (S1)–(S4). We define inductively sets  $F_i$  for  $i \geq 0$  and  $F'_i$  for  $i \geq 1$  by setting  $F_0 = F^{\mathcal{A}}$ ,  $F'_{i+1} = \{y \mid \mathcal{A} \models \mathbf{q}'(x, y) \wedge \mathbf{q}'_1(x) \wedge \mathbf{q}'_2(y) \text{ where } x \in F_i\}$  and  $F_i = \{y \in F'_{i+1} \mid A(y) \in \mathcal{A}\}$  for  $i \geq 1$ . We define a model  $\mathcal{I}$  of  $\mathcal{T}$  and  $\mathcal{A}$  by extending  $F^{\mathcal{A}}$  and  $T^{\mathcal{A}}$  to  $F^{\mathcal{I}} = \bigcup_{i=0}^{\infty} F_i$  and  $T^{\mathcal{I}} = T^{\mathcal{A}} \cup A^{\mathcal{A}} \setminus \bigcup_{i=1}^{\infty} F_i$ . We claim that  $\mathcal{I} \not\models \mathbf{q}$ . Indeed, suppose there is a homomorphism  $h: \mathcal{q} \rightarrow \mathcal{I}$ . Then there is  $i_0$  such that  $h(x) \in F_{i_0}$ . Moreover,  $h(y)$  must be in the intersection of  $F_{i_0+1}$  and  $T^{\mathcal{I}}$ . However, due to our assumption, no node in  $F'_{i_0+1}$  can be in  $T^{\mathcal{A}}$ , and by construction, no node in  $F'_{i_0+1}$  can be in  $T^{\mathcal{I}} \setminus T^{\mathcal{A}}$ , which is impossible.

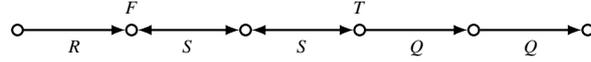
A linear datalog program  $\Pi$  is *symmetric* if, for any recursive rule  $I(\mathbf{x}) \leftarrow J(\mathbf{y}) \wedge E(\mathbf{z})$  in  $\Pi$  (except the goal rules), where  $E(\mathbf{z})$  is a shorthand for the conjunction of the EDBs of the rule, its symmetric counterpart  $J(\mathbf{y}) \leftarrow I(\mathbf{x}) \wedge E(\mathbf{z})$  is also in  $\Pi$ . It is known (see, e.g., [50]) that symmetric programs can be evaluated in L for data complexity.

It remains to observe that checking whether there are  $v_0, v_1, \dots, v_n \in \text{ind}(\mathcal{A})$  such that (S1)–(S4) hold can be done by the following symmetric datalog program, in which  $B(x) = A(x) \wedge \mathbf{q}'_1(x) \wedge \mathbf{q}'_2(x)$ :

$$\begin{aligned} \mathbf{G} &\leftarrow \mathbf{q} \\ \mathbf{G} &\leftarrow F(x), \mathbf{q}'_1(x), \mathbf{q}'(x, y), P(y) \\ P(x) &\leftarrow B(x), \mathbf{q}'(x, y), \mathbf{q}'_2(y), T(y) \\ P(x) &\leftarrow B(x), \mathbf{q}'(x, y), P(y), B(y). \end{aligned}$$

where the only recursive rule  $P(x) \leftarrow B(x), \mathbf{q}'(x, y), P(y), B(y)$  is equivalent to its symmetric counterpart due to the symmetry of  $\mathbf{q}'(x, y)$ .  $\square$

**Example 6.** By Theorems 5 and 4 (ii), the OMQ  $(\text{cov}_{\top}, \mathbf{q})$  with  $\mathbf{q}$  shown below is L-complete.



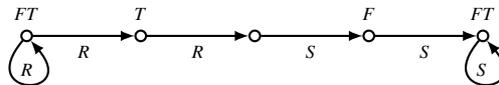
Since  $\text{AC}^0 \subsetneq \text{L}$ , Theorem 4 gives a sufficient and necessary criterion in the presence of the disjointness axiom:

**Corollary 7.** An OMQ  $(\text{cov}_A^{\perp}, \mathbf{q})$  is in  $\text{AC}^0$  iff  $\mathbf{q}$  is a 0-CQ or contains a twin. If  $\mathbf{q}$  is a twinless 0-CQ, then  $\mathbf{q}$  is an FO-rewriting of  $(\text{cov}_A^{\perp}, \mathbf{q})$ .

*Proof.* If  $\mathbf{q}$  has a twin, then  $\exists x (F(x) \wedge T(x))$  is an FO-rewriting of  $\mathbf{Q}$ . So suppose  $\mathbf{q}$  is twinless. By Theorem 4 and since  $\text{AC}^0 \subsetneq \text{L}$ ,  $\mathbf{Q}$  is in  $\text{AC}^0$  iff  $\mathbf{q}$  is a 0-CQ. Suppose  $\mathbf{Q}$  is in  $\text{AC}^0$  and  $\mathbf{q}$  has no solitary  $F$  (the other case is similar). Then  $\mathbf{Q}$  is FO-rewritable, and so, by [20, Proposition 5.9], it must have a rewriting in the form of a union (disjunction) of CQs. Consider any CQ  $\mathbf{q}'$  in this rewriting. Let  $\mathcal{A}$  be an ABox isomorphic to  $\mathbf{q}'$  (as a labelled digraph). Then  $\text{cov}_A^{\perp}, \mathcal{A} \models \mathbf{q}$ . Let  $\mathcal{A}^F$  result from  $\mathcal{A}$  by adding a label  $F$  to any  $A$ -node that is not labelled by  $F$  or  $T$ . Then there is a homomorphism  $h: \mathbf{q} \rightarrow \mathcal{A}^F$ . As  $\mathbf{q}$  does not have  $F$ - and  $A$ -nodes,  $h$  is also a homomorphism from  $\mathbf{q}$  to  $\mathbf{q}'$ . It follows that  $\mathbf{q}$  is an FO-rewriting of  $\mathbf{Q}$ .  $\square$

The next example shows that the criterion of Corollary 7 does not hold for CQs with twins (see also Example 13).

**Example 8.** It is not hard to check (directly or using Theorem 12 below that  $(\text{cov}_A, \mathbf{q})$  with  $\mathbf{q}$  shown below has  $\mathbf{q}$  as its FO-rewriting, and so is in  $\text{AC}^0$ . Note that  $\mathbf{q}$  is *minimal* in the sense that it is not equivalent to any of its proper sub-CQs.



### 3.3. Datalog Rewritability of OMQs with a 1-CQ

In this section, we introduce some technical tools for dealing with 1-CQs. Here, 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  $F$ . The tools are an adaptation of known (disjunctive) datalog techniques to OMQs  $\mathcal{Q}$  with a 1-CQ. More specifically, we observe that every such  $\mathcal{Q}$  can be rewritten to a very simple datalog query—nearly a sirup in the sense of [51]—which can be regarded as an adaptation of the idea of markability for disjunctive datalog programs from [39]. We also adapt the datalog expansion technique [52, 32, 45] to characterise those datalog queries semantically.

Throughout this section, we assume that  $\mathbf{q}$  is a 1-CQ such that  $F(x)$  and  $T(y_1), \dots, T(y_n)$  are all of the solitary occurrences of  $F$  and  $T$  in  $\mathbf{q}$ , and let  $\mathcal{Q} = (\mathcal{T}, \mathbf{q})$ . For each such  $\mathcal{Q}$ , we define a monadic datalog program  $\Pi_{\mathcal{Q}}$  with goal  $\mathbf{G}$  and four rules

$$\mathbf{G} \leftarrow F(x), \mathbf{q}', P(y_1), \dots, P(y_n) \quad (6)$$

$$P(x) \leftarrow T(x) \quad (7)$$

$$P(x) \leftarrow A(x), \mathbf{q}', P(y_1), \dots, P(y_n) \quad (8)$$

$$\mathbf{G} \leftarrow F(x), T(x) \quad (9)$$

where  $\mathbf{q}' = \mathbf{q} \setminus \{F(x), T(y_1), \dots, T(y_n)\}$  and  $P$  is a fresh predicate symbol which never occurs in ABoxes. If  $\mathcal{T} = \text{cov}_A$ , rule (9) is omitted.

We also define by induction a class  $\mathfrak{R}_{\mathcal{Q}}$  of ABoxes called *cactuses for  $\mathcal{Q}$* . We start by setting  $\mathfrak{R}_{\mathcal{Q}} = \{\mathbf{q}\}$ , regarding  $\mathbf{q}$  as an ABox, and then recursively apply to  $\mathfrak{R}_{\mathcal{Q}}$  the following two rules:

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

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

It is straightforward to see by structural induction that

$$\mathcal{T}, C \models \mathbf{q}, \text{ for every } C \in \mathfrak{R}_{\mathcal{Q}}. \quad (10)$$

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

**Theorem 9.** *For every OMQ  $\mathcal{Q} = (\mathcal{T}, \mathbf{q})$  with a 1-CQ  $\mathbf{q}$  and every ABox  $\mathcal{A}$ , the following are equivalent:*

- (i)  $\mathcal{T}, \mathcal{A} \models \mathbf{q}$ ;
- (ii)  $\Pi_{\mathcal{Q}}, \mathcal{A} \models \mathbf{G}$ ;
- (iii) either  $\mathcal{T} = \text{cov}_A^{\perp}$  and  $\mathcal{A}$  contains an FT-twin, or there exists a homomorphism  $h: C \rightarrow \mathcal{A}$  for some unpruned  $C \in \mathfrak{R}_{\mathcal{Q}}$ .

*Proof.* We show (i)  $\Rightarrow$  (ii)  $\Rightarrow$  (iii)  $\Rightarrow$  (i).

(i)  $\Rightarrow$  (ii) If  $\mathcal{T} = \text{cov}_A^{\perp}$  and  $\mathcal{A}$  contains a node labelled by both  $T$  and  $F$ , then  $\mathbf{G}$  holds in the closure  $\Pi_{\mathcal{Q}}(\mathcal{A})$  of  $\mathcal{A}$  under  $\Pi_{\mathcal{Q}}$  by rule (9). In any other case, we define a model  $\mathcal{I}$  based on  $\mathcal{A}$  by labelling each ‘undecided’  $A$ -node  $a$  by  $T^{\mathcal{I}}$  if  $P(a)$  holds in  $\Pi_{\mathcal{Q}}(\mathcal{A})$ , and by  $F^{\mathcal{I}}$  otherwise. As  $\mathcal{I}$  is a model of  $\mathcal{T}$  and  $\mathcal{A}$ , there is a homomorphism  $h: \mathbf{q} \rightarrow \mathcal{I}$ . Then  $h(y_i) \in T^{\mathcal{I}}$ , and so  $P(h(y_i))$  holds in  $\Pi_{\mathcal{Q}}(\mathcal{A})$ , for every  $i \leq n$  (by rule (7) and the definition of  $\mathcal{I}$ ). We claim that  $h(x)$  is an  $F$ -node in  $\Pi_{\mathcal{Q}}(\mathcal{A})$ , and so  $\mathbf{G}$  holds in  $\Pi_{\mathcal{Q}}(\mathcal{A})$  by rule (6). Indeed, otherwise by  $h(x) \in F^{\mathcal{I}}$  and the definition of  $\mathcal{I}$ ,  $h(x)$  is an  $A$ -node but not a  $P$ -node in  $\Pi_{\mathcal{Q}}(\mathcal{A})$ , contrary to rule (8).

(ii)  $\Rightarrow$  (iii) Suppose that  $\mathcal{T} = \text{cov}_A$  or  $\mathcal{A}$  does not contain a node labelled by both  $T$  and  $F$ . Then rule (9) is either not in  $\Pi_{\mathcal{Q}}$  or not used. We define inductively (on the applications of rule (8) in the derivation of  $\mathbf{G}$ ) an unpruned cactus  $C \in \mathfrak{R}_{\mathcal{Q}}$  and a homomorphism  $h: C \rightarrow \mathcal{A}$ . To begin with, there are objects  $x^a, y_1^a, \dots, y_n^a$  for which rule (6) was triggered. So  $x^a$  is an  $F$ -node in  $\Pi_{\mathcal{Q}}(\mathcal{A})$ , and so it is an  $F$ -node in  $\mathcal{A}$ . Take a function  $h_0: \mathbf{q} \rightarrow \mathcal{A}$  such that it

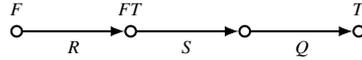
preserves binary predicates,  $h_0(x) = x^a$  and  $h_0(y_i) = y_i^a$  for  $i \leq n$ . If  $y_i^a$  is a  $T$ -node in  $\mathcal{A}$  for every  $i \leq n$ , then  $h = h_0$  is the required homomorphism from  $\mathbf{q} \in \mathfrak{R}_Q$  to  $\mathcal{A}$ . If  $y_i^a$  is not a  $T$ -node in  $\mathcal{A}$  for some  $i$ , then  $y_i^a$  is a  $P$ -node in  $\Pi_Q(\mathcal{A})$  obtained by rule (8), and so  $y_i^a$  is an  $A$ -node in  $\mathcal{A}$ . Also, there are objects  $x^b = y_i^a$  and  $y_1^b, \dots, y_n^b$  such that rule (8) was triggered for  $x^b, y_1^b, \dots, y_n^b$ . Let  $C$  be the cactus obtained from  $\mathbf{q}$  by budding at  $y_i$ , and extend  $h_0$  to a function  $h_1: C \rightarrow \mathcal{A}$  such that it preserves binary predicates and  $h_1(y_j^s) = y_j^b$  for all  $T$ -nodes  $y_j^s$  of the new segment  $s$ . If  $y_j^b$  is a  $T$ -node in  $\mathcal{A}$  for every  $j \leq n$ , then  $h = h_1$  is the required homomorphism from  $C \in \mathfrak{K}_Q$  to  $\mathcal{A}$ . Otherwise, we bud  $C$  again and repeat the above argument. As the derivation of  $\mathbf{G}$  from  $\mathcal{A}$  using  $\Pi_Q$  is finite, sooner or later the procedure stops with a cactus and a homomorphism.

(iii)  $\Rightarrow$  (i) If  $\mathcal{T} = \text{cov}_A^+$  and  $\mathcal{A}$  contains a node labelled by both  $T$  and  $F$  then  $\mathcal{T}, \mathcal{A} \models \mathbf{q}$  obviously holds. Otherwise, take an arbitrary model  $\mathcal{I}$  of  $\mathcal{T}$  and  $\mathcal{A}$ . We define a model  $\mathcal{I}^+$  of  $\mathcal{T}$  and  $C$  by ‘pulling back  $\mathcal{I}$ ’ via the homomorphism  $h$ : for every node  $x$  in  $C$ ,  $x \in A^{\mathcal{I}^+}$  iff  $h(x) \in A^{\mathcal{I}}$ . By (10), there is a homomorphism  $g: \mathbf{q} \rightarrow \mathcal{I}^+$ . Thus, the composition of  $g$  and  $h$  is a  $\mathbf{q} \rightarrow \mathcal{I}$  homomorphism, as required.  $\square$

**Corollary 10.** Any  $OMQ \mathbf{Q} = (\mathcal{T}, \mathbf{q})$  with a 1-CQ  $\mathbf{q}$  is datalog-rewritable, and so is in P.

Corollary 10 makes it possible to use the 2ExpTIME algorithm of [32] to decide whether  $\Pi_Q$  is bounded, and so  $\mathbf{Q}$  is in  $AC^0$ , and the results of [34, 35, 38] and many other techniques to understand whether  $\Pi_Q$  can be transformed to a linear program, which would mean that  $\mathbf{Q}$  is in NL. For  $OMQs \mathbf{Q}$  whose 1-CQ  $\mathbf{q}$  is a ditree with its unique solitary  $F$ -node as root, the program  $\Pi_Q$  can be reformulated as an  $\mathcal{EL}$  ontology, and so one can use the  $AC^0/NL/P$  trichotomy of [28, 53], which is checkable in ExpTIME.

**Example 11.** To illustrate, consider the 1-CQ  $\mathbf{q}$  below.



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

#### 4. Deciding FO-rewritability of $OMQs$ with a 1-CQ

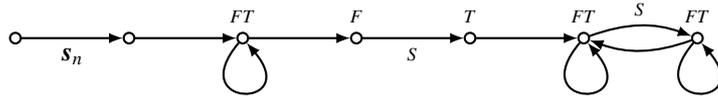
The following semantic criterion of FO-rewritability is standard; cf. [32] in the datalog setting:

**Theorem 12.** An  $OMQ \mathbf{Q} = (\mathcal{T}, \mathbf{q})$  with a 1-CQ  $\mathbf{q}$  is FO-rewritable iff there exists  $d < \omega$  such that every  $C \in \mathfrak{R}_Q$  contains a homomorphic image of some unpruned  $C^- \in \mathfrak{R}_Q$  of depth  $\leq d$ .

*Proof.* ( $\Rightarrow$ ) By the proof of Corollary 7,  $\mathbf{Q}$  has an FO-rewriting of the form  $\mathbf{q}_1 \vee \dots \vee \mathbf{q}_n$ , where the  $\mathbf{q}_i$  are CQs (possibly containing  $A$ -nodes). Treating the  $\mathbf{q}_i$  as ABoxes, we obviously have  $\mathcal{T}, \mathbf{q}_i \models \mathbf{q}$ , and so, by Theorem 9, there is a homomorphism from some unpruned  $C_i \in \mathfrak{R}_Q$  to  $\mathbf{q}_i$ . Now let  $d$  be the maximum of the depths of the  $C_i$ , and take any  $C \in \mathfrak{R}_Q$  of depth  $> d$ . Then there are homomorphisms  $C_i \rightarrow \mathbf{q}_i \rightarrow C$ , for some  $i$ ,  $1 \leq i \leq n$ , as required.

( $\Leftarrow$ ) Given  $d < \omega$ , we take all the unpruned cactuses  $C_1, \dots, C_n$  of depth  $\leq d$  (up to isomorphism). Now we consider each  $C_i$  as a CQ. Then  $C_1 \vee \dots \vee C_n$  is an FO-rewriting of  $\mathbf{Q}$ . Indeed, if  $\mathcal{T}, \mathcal{A} \models \mathbf{q}$  then there are homomorphisms  $C_i \rightarrow C \rightarrow \mathcal{A}$ , for some  $C$  and  $i$ , again by Theorem 9.  $\square$

**Example 13.** Let  $s_n$  be a chain of  $n \geq 3$  arrows labelled by  $S$ . Consider the CQ  $\mathbf{q}_n$  shown below, where the omitted labels on edges are all  $R$ . It is not hard to check that  $\mathbf{q}_n$  is minimal (not equivalent to any of its proper sub-CQs).



Let  $C_k$  be the cactus obtained by applying (**bud**)  $k$ -times to  $C_0 = \mathbf{q}_3$ . Then there is a homomorphism  $\mathbf{q}_3 \rightarrow C_k$ , for any  $k \geq 2$ : it uses the  $S$ -chain before the  $T$ -node to accommodate  $s_3$ . However, there is no homomorphism from  $\mathbf{q}_3$  to  $C_1$  as  $s_3$  is too long. It follows that  $\mathbf{q}_3 \vee C_1$  is an FO-rewriting of  $(\text{cov}_A, \mathbf{q}_3)$ , where we treat  $C_1$  as a CQ. It is to be noted that  $C_1$  has an  $A$ -node. In general, the UCQ  $\mathbf{q} \vee C_1 \vee \dots \vee C_{n-2}$  is an FO-rewriting of  $(\text{cov}_A, \mathbf{q}_n)$ .

The following result should be compared to [44, Theorem 11] showing PSPACE-hardness of FO-rewritability of UCQs mediated by Schema.org ontologies. In our case, the expressive power of UCQs (used in [44] to polynomially capture various different aspects of exponentially large structures) is not available, and so we had to develop a brand new way of capturing *all* of these aspects by a *single* CQ.

**Theorem 14.** *It is PSPACE-hard to decide whether a given OMQ  $\mathcal{Q} = (\text{cov}_A, \mathbf{q})$  with a (dag) 1-CQ  $\mathbf{q}$  (having one solitary  $F$  and two solitary  $T$ s) is FO-rewritable.*

The remainder of this section is dedicated to the proof of Theorem 14, which is by reduction of the PSPACE-complete *acyclicity problem for succinct digraphs* encoded as Boolean formulas [54]; see also [55, Claim 4.4]. We will use the criterion for FO-rewritability of Theorem 12.

We remind the reader that a *Boolean formula with variables  $\mathbf{x}$*  is a ditree  $\varphi(\mathbf{x})$  whose vertices are called *gates*. Leaf gates are labelled by the variables from  $\mathbf{x} = (x_1, \dots, x_n)$ , where each variable  $x_i$  can label several leaves of  $\varphi(\mathbf{x})$ . Each non-leaf gate  $g$  is either an AND-gate (having 2 children) or a NOT-gate (having 1 child), with the outgoing edge(s) leading to the *input(s)* of  $g$ . Given an assignment  $\alpha$  of  $F$  and  $T$  to the variables  $\mathbf{x}$  of  $\varphi$ , we compute the value of each gate in  $\varphi$  under  $\alpha$  as usual in Boolean logic. We let  $\varphi[\alpha]$  denote the truth-value of the root gate. The size  $|\varphi|$  of  $\varphi$  is the number of its gates.

A digraph  $G_\psi$  on  $2^k$  nodes from  $\{F, T\}^k$  is *succinctly represented by a Boolean formula  $\psi(\mathbf{x}, \mathbf{y})$*  with  $|\mathbf{x}| = |\mathbf{y}| = k$  in case  $(\alpha, \beta)$  is an edge in  $G_\psi$  iff  $\psi(\alpha, \beta) = T$ , for any  $\alpha, \beta \in \{F, T\}^k$ . The *acyclicity problem for succinct graphs* is to decide, given a succinct representation  $\psi$  of  $G_\psi$ , whether  $G_\psi$  is acyclic or not. Clearly,  $G_\psi$  is acyclic iff, for any sequence  $\alpha_1, \dots, \alpha_m$  of more than  $2^k$  nodes in  $G_\psi$ , there is  $i < m$  such that  $(\alpha_i, \alpha_{i+1})$  is not an edge in  $G_\psi$ . Our aim is, given such a formula  $\psi$ , to construct a 1-CQ  $\mathbf{q}_\psi$  such that  $|\mathbf{q}_\psi|$  is polynomial in  $|\psi|$  and the following holds:

**Lemma 14.1.** *For every sequence  $\alpha_1, \dots, \alpha_m$  of more than  $2^k$  nodes in  $G_\psi$ , there is  $i < m$  with  $\psi(\alpha_i, \alpha_{i+1}) = F$  iff there is  $d < \omega$  such that every  $C \in \mathfrak{R}_{\mathbf{q}_\psi}$  contains a homomorphic image of some unpruned  $C^- \in \mathfrak{R}_{\mathbf{q}_\psi}$  of depth  $\leq d$ .*

Theorem 14 is an immediate consequence of Theorem 12 and Lemma 14.1.

#### 4.1. General plan

We fix  $\psi$ , and let  $\mathbf{q} = \mathbf{q}_\psi$  and  $\mathcal{Q} = (\text{cov}_A, \mathbf{q})$ . The 1-CQ  $\mathbf{q}$  will be dag-shaped with one solitary  $F$ -node (called the *centre* of  $\mathbf{q}$ ), two solitary  $T$ -nodes ( $t_F$  and  $t_T$ ) and one  $FT$ -twin. The CQ  $\mathbf{q}^-$  is obtained from  $\mathbf{q}$  by replacing the  $F$ -label of its centre by  $A$ . All leaf segments in unpruned cactuses different from  $\mathbf{q}$  are of this form. Also,  $\mathbf{q}$  will be such that

if  $h: C \rightarrow C'$  is a homomorphism for some cactuses  $C, C' \in \mathfrak{R}_{\mathcal{Q}}$ , then  
the only solitary  $F$ -node in  $C$  is mapped by  $h$  to the only solitary  $F$ -node in  $C'$ . (11)

The (copy of the) centre of  $\mathbf{q}$  is also called the *centre of  $\mathfrak{s}$* , for any segment  $\mathfrak{s}$  occurring in cactuses in  $\mathfrak{R}_{\mathcal{Q}}$ . Since our 1-CQs do not contain  $A$ -nodes, (11) implies that if  $h: C \rightarrow C'$  is a homomorphism for some cactuses  $C, C' \in \mathfrak{R}_{\mathcal{Q}}$ , then for every segment  $\mathfrak{s}$  in  $C$ , the centre of  $\mathfrak{s}$  should be mapped by  $h$  to the centre of some segment  $\mathfrak{s}'$  in  $C'$ . So we say that  *$h$  maps  $\mathfrak{s}$  into  $\mathfrak{s}'$*  if  $h$  maps the centre of  $\mathfrak{s}$  to the centre of  $\mathfrak{s}'$ . Observe that (11) also implies that

if  $h: C \rightarrow C'$  is a homomorphism for some cactuses  $C, C' \in \mathfrak{R}_{\mathcal{Q}}$ , then  
 $h$  maps the root segment of  $C$  into the root segment of  $C'$ . (12)

#### 4.2. Encoding path-cactuses and graph-node sequences by $FT$ -sequences

An  *$FT$ -sequence* (of length  $k \leq \omega$ ) is any element of  $\{F, T\}^k$ . Given  $1 \leq i \leq j \leq k$ , we call  $(\alpha_i, \alpha_{i+1}, \dots, \alpha_{j-1}, \alpha_j)$  a *subsequence* of  $(\alpha_1, \dots, \alpha_k)$ . Since  $\mathbf{q}$  has two solitary  $T$ -nodes ( $t_F$  and  $t_T$ ), we can regard the skeleton  $C^{\mathfrak{s}}$  of any unpruned path-cactus  $C \in \mathfrak{R}_{\mathcal{Q}}$  as an  $FT$ -sequence, according to which  $T$ -nodes were budded at each segment. Conversely, for every finite  $FT$ -sequence  $\delta$ , there is a uniquely determined unpruned path-cactus  $C \in \mathfrak{R}_{\mathcal{Q}}$  such that  $\delta = C^{\mathfrak{s}}$ .

Given a (finite or infinite) sequence  $\alpha = (\alpha_1, \alpha_2, \dots)$  of nodes in  $G_\psi$  (with each of  $\alpha_i$  being a  $k$ -long  $FT$ -sequence), we encode  $\alpha$  by an  $FT$ -sequence as follows. First, for each  $i$ , let  $\alpha'_i$  be obtained from  $\alpha_i$  by writing each letter

twice. Then we encode  $\alpha$  by the  $FT$ -sequence  $FT\alpha'_1FT\alpha'_2\dots$ . We call a  $4k + 4$ -long  $FT$ -sequence *correct* if it is a subsequence of an encoding of some graph-node sequence  $\alpha$ .

Given  $\psi$  (with  $2k$  variables), it is straightforward to define another Boolean formula  $\varphi_\psi$  of size polynomial in  $|\psi|$  and with  $4k + 4$  variables such that, for every  $4k + 4$ -long  $FT$ -sequence  $\delta$ , we have  $\varphi_\psi(\delta) = T$  iff either  $\delta$  is incorrect, or  $\delta = FT\alpha'FT\beta'$  for some  $\alpha, \beta \in \{F, T\}^k$  with  $\psi(\alpha, \beta) = F$  (that is,  $(\alpha, \beta)$  not being an edge of the graph  $G_\psi$ ).

The *co-depth* of a segment  $\mathfrak{s}$  in a path-cactus  $C$  is the length (number of edges) of the path in  $C^s$  starting at  $\mathfrak{s}$ . We want to ensure the following:

**Lemma 14.2.** *For any unpruned path-cactus  $C \in \mathfrak{R}_Q$  and segment  $\mathfrak{s}$  of co-depth  $\geq 4k + 4$  in  $C^s$ , the following hold:*

- (i) *There is a homomorphism  $h: \mathfrak{q}^- \rightarrow C$  mapping  $\mathfrak{q}^-$  into  $\mathfrak{s}$  iff  $\varphi_\psi(P_{\mathfrak{s}, C}^{4k+4}) = T$  for the  $4k + 4$ -long  $FT$ -sequence  $P_{\mathfrak{s}, C}^{4k+4}$  starting at the child of  $\mathfrak{s}$  in  $C^s$ .*
- (ii) *If there is a homomorphism  $h: \mathfrak{q}^- \rightarrow C$  mapping  $\mathfrak{q}^-$  into some  $\mathfrak{s}$  with node  $t_i$  budded and  $t_j$  not budded in  $\mathfrak{s}$ , for  $i, j \in \{F, T\}$ , then  $h$  can be chosen so that  $h(t_i) = \theta$  and  $h(t_j) = t_j$ , where  $\theta$  is the  $FT$ -node in  $\mathfrak{s}$ .*

#### 4.3. Proof of Lemma 14.1 using Lemma 14.2

( $\Leftarrow$ ) Suppose there is a sequence  $\alpha_1, \dots, \alpha_m$  of more than  $2^k$  nodes in  $G_\psi$  such that  $\psi(\alpha_i, \alpha_{i+1}) = T$  for every  $i < m$ . As  $G_\psi$  has  $2^k$  nodes and  $m > 2^k$ , there is  $i < m$  with  $\alpha_m = \alpha_i$ . Let  $\beta^\infty$  be the infinite periodic graph-node sequence obtained by repeating  $\alpha_i, \alpha_{i+1}, \dots, \alpha_m$ , and let  $\delta^\infty$  encode  $\beta^\infty$  as above. Given  $d < \omega$ , let  $\delta$  be the  $d + 4k + 5$ -long prefix of  $\delta^\infty$ . Let  $C_d$  be the unpruned path-cactus such that  $C_d^s = \delta$ . Then  $\varphi_\psi(P_{\mathfrak{s}, C_d}^{4k+4}) = F$  holds for every segment  $\mathfrak{s}$  of co-depth  $\geq 4k + 4$  in  $C_d^s$ . As any segment  $\mathfrak{s}'$  of depth  $\leq d$  in  $C_d^s$  is of co-depth  $\geq 4k + 4$ , by Lemma 14.2 (i), there is no homomorphism from  $\mathfrak{q}^-$  to  $C_d$  mapping  $\mathfrak{q}^-$  into any segment  $\mathfrak{s}'$  of  $C_d$  of depth  $\leq d$ .

It follows that no unpruned cactus  $C^-$  of depth  $\leq d$  can be homomorphically mapped to  $C_d$ . Indeed, suppose on the contrary that there is a homomorphism  $h$  from  $C^-$  to  $C_d$ . Take some leaf segment  $\mathfrak{s}$  of  $C^-$ . By (12),  $\mathfrak{s}$  (and so  $\mathfrak{q}^-$ ) should be mapped by  $h$  into some segment  $\mathfrak{s}'$  of  $C_d$  whose depth is  $\leq d$  in  $C_d^s$ , which is a contradiction.

( $\Rightarrow$ ) Suppose that, for any sequence  $\alpha_1, \dots, \alpha_m$  of more than  $2^k$  nodes in  $G_\psi$ , there is  $i < m$  with  $\psi(\alpha_i, \alpha_{i+1}) = F$ . Then any  $\alpha \in \{F, T\}^\ell$  for  $\ell \geq (2^k + 1) \cdot (2k + 2)$  must contain a  $4k + 4$ -long subsequence  $\delta$  such that  $\varphi_\psi(\delta) = T$  (as either there is a  $4k + 4$ -long subsequence of  $\delta$  that is incorrect, or there is one encoding a pair  $(\alpha, \beta)$  of nodes in  $G_\psi$  with  $\psi(\alpha, \beta) = F$ ). Set  $d = (2^k + 1) \cdot (2k + 2)$ , and consider some cactus  $C$  of depth  $> d$ . We cut each of its branches at some depth  $\leq d$ , and show that the resulting cactus  $C^-$  can be mapped homomorphically into  $C$ .

To this end, let  $\mathcal{B}$  be a path-cactus corresponding to some branch of  $C$  such that  $\mathcal{B}^s$  is longer than  $d$ . We cut  $\mathcal{B}$  at some depth  $\leq d$ , and show that the resulting cactus  $C'$  can be mapped homomorphically into  $C$ . As  $\mathcal{B}^s$  is longer than  $d$ , the  $d$ -long prefix of  $\mathcal{B}^s$  must contain a  $4k + 4$ -long subsequence  $\delta$  such that  $\varphi_\psi(\delta) = T$ . Let  $\mathfrak{s}_\mathcal{B}$  be the segment in  $\mathcal{B}^s$  corresponding to the letter preceding the first letter of  $\delta$ . Then the depth of  $\mathfrak{s}_\mathcal{B}$  is  $< d$  in  $\mathcal{B}^s$ , its co-depth is  $\geq 4k + 4$ , and  $\delta = P_{\mathfrak{s}_\mathcal{B}, \mathcal{B}}^{4k+4}$ . Thus, by Lemma 14.2 (i), there is a homomorphism  $h_\mathcal{B}: \mathfrak{q}^- \rightarrow \mathcal{B}$  mapping  $\mathfrak{q}^-$  into  $\mathfrak{s}_\mathcal{B}$ . Let  $C'$  be obtained from  $C$  by cutting  $\mathcal{B}$  at  $\mathfrak{s}_\mathcal{B}$ . Let  $\mathfrak{s}_{C'}$  and  $\mathfrak{s}_C$  be the respective segments in  $C'$  and  $C$  corresponding to  $\mathfrak{s}_\mathcal{B}$ . If  $\mathfrak{s}_C = \mathfrak{s}_\mathcal{B}$  then, by taking  $h_\mathcal{B}$  on  $\mathfrak{s}_{C'} = \mathfrak{q}^-$  and the identity map on all other segments, we obtain a  $C' \rightarrow C$  homomorphism. So suppose that  $\mathfrak{s}_{C'} \neq \mathfrak{q}^-$ ,  $t_j$  is the budded  $T$ -node and  $t_i$  is the unbudded  $T$ -node in  $\mathfrak{s}_{C'}$ . Then  $t_i$  is budded and  $t_j$  is unbudded in  $\mathfrak{s}_\mathcal{B}$ , and both  $t_i$  and  $t_j$  are budded in  $\mathfrak{s}_C$ . By Lemma 14.2 (ii),  $h_\mathcal{B}$  can be chosen such that it is a  $\mathfrak{s}_{C'} \rightarrow C$  homomorphism mapping  $\mathfrak{s}_{C'}$  into  $\mathfrak{s}_C$ . So, by taking  $h_\mathcal{B}$  on  $\mathfrak{s}_{C'}$  and the identity map on all other segments, we again obtain a  $C' \rightarrow C$  homomorphism. If  $C'$  still has branches longer than  $d$ , we repeat the above process for a long branch in  $C'$  to obtain a  $C'' \rightarrow C'$  homomorphism for some  $C''$ , and so on. Sooner or later, we obtain a cactus  $C^-$  of depth  $\leq d$  homomorphically mapping into  $C$ .

#### 4.4. Query-design

Now we design a 1-CQ  $q$  satisfying (11) and Lemma 14.2. Apart from one  $F$ -node, two  $T$ -nodes and one  $FT$ -twin, it has nodes labelled by unary predicates  $B, D, E$ , and  $B_{ij}$ , for some  $i, j$ , and edges labelled by binary predicates  $R$  and  $S$ . The extra unary labels are just syntactic sugar in the sense that any node  $a$  labelled by, say  $B$ , can be regarded as a shorthand for a  $B$ -edge  $(a, a')$  to some fresh node  $a'$ . To simplify notation, in our pictures we omit the  $R$ -labels from  $R$ -arrows. Letters other than upper case italics (lower case italics and bold) are used as pointers to certain nodes, and do not denote predicates in  $q$ .

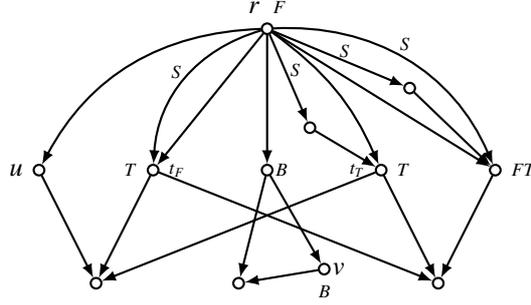


Figure 1: Block  $q^{base}$  of the 1-CQ  $q$ .

We assemble  $q$  from three blocks shown in Figures 1–3 by glueing node  $u$  of  $q^{var}$  in Fig. 3 to node  $u$  in  $q^{base}$  in Fig. 1, and node  $v$  of  $q^{form}$  in Fig. 2 to node  $v$  in  $q^{base}$ .

The block  $q^{form}$  encodes the formula  $\varphi_\psi(x_1, \dots, x_n)$  for  $n = 4k + 4$  as follows. With each non-leaf gate  $g$  in  $\varphi_\psi$  we associate a fresh copy of its gadget in the lower part of Fig. 2. Here, by  $(D)$  we mean that the label  $D$  is only present when the gate in question is the root gate of  $\varphi_\psi$ . For each  $i$ , we let  $k_i$  denote the number of leaves in the ditree  $\varphi_\psi$  with label  $x_i$ . Then each branch of  $\varphi_\psi$  can be characterised by a pair  $(i, j)$  such that the leaf node of the branch is labelled by the  $j$ th copy  $x_i^j$  of the variable  $x_i$  for some  $1 \leq i \leq n$  and  $1 \leq j \leq k_i$ . When the inputs of some AND-gate  $g$  are gates  $g_1$  and  $g_2$  then, for each  $m = 1, 2$ , if  $g_m$  is a non-leaf gate, we merge node  $\mathfrak{o}$  of the  $g_m$ -gadget with node  $\mathfrak{i}_m$  of the  $g$ -gadget; and if  $g_m$  is labelled by  $x_i^j$ , we merge node  $\mathfrak{i}_m$  of the  $g$ -gadget with the lower  $B_{ij}$ -node in the upper part of Fig. 2. We proceed similarly with NOT-gates as well.

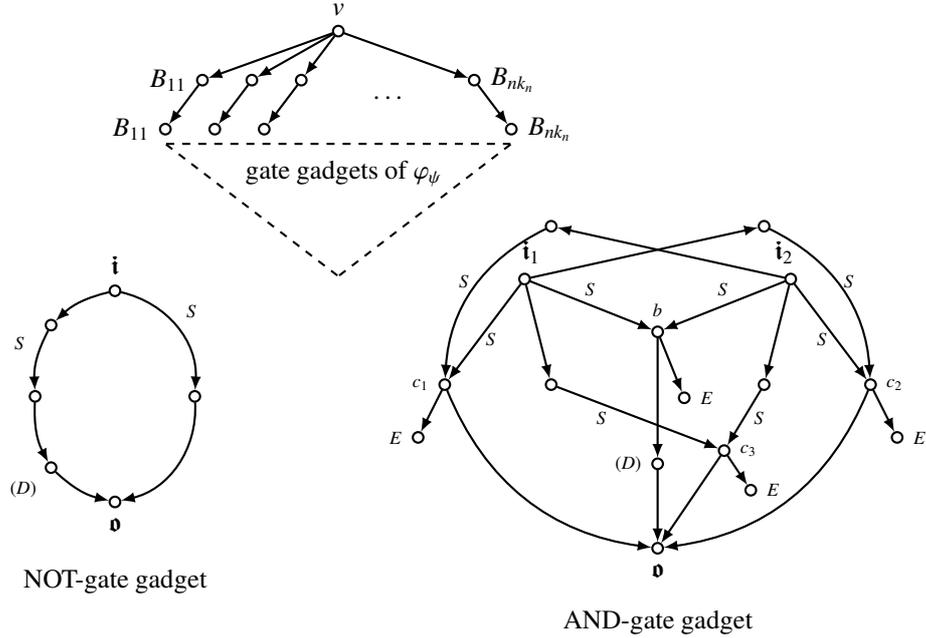


Figure 2: Block  $q^{form}$  of the 1-CQ  $q$ .

The third block  $q^{var}$  encodes the input for the variables  $x_1, \dots, x_n$  in  $\varphi_\psi$  as follows. First, its root  $u$  branches out to  $n + 1$  branches  $b_1, \dots, b_{n+1}$  in Fig. 3:

- (a) Branch  $b_{n+1}$  begins with a chain of  $n + 2$   $R$ -edges whose last two nodes are labeled by  $B$ . Each of these two  $B$ -nodes has an  $R$ -edge to the same node  $z$ .

- (b) Branch  $\mathbf{b}_i$  starts with  $i - 1$   $R$ -edges, then one  $S$ -edge, followed by a chain of  $n - i + 2$   $R$ -edges whose last node is labeled by  $B$ . This last  $B$  node also has an  $R$ -edge leading to the  $z$  node in (a), and it also branches out to further  $k_i$  branches  $\mathbf{b}_i^1, \dots, \mathbf{b}_i^{k_i}$ , with  $\mathbf{b}_i^j$  corresponding to the occurrence  $x_i^j$  of  $x_i$  among the leaf-labels in  $\varphi_\psi$ , for  $1 \leq j \leq k_i$ . For each  $j$ , let  $g_{ij}^1, \dots, g_{ij}^{d_{ij}}$  be the sequence of non-leaf gates from leaf to root on the branch with leaf  $x_i^j$ . Then  $\mathbf{b}_i^j$  starts with two  $R$ -edges with the end-node of the second one being labelled by  $B_{ij}$ , and it continues with a three-step RSR edge-pattern repeated  $d_{ij}$  times; see the left-hand side of Fig. 3. The first pattern corresponds to the gate  $g_{ij}^1$ , the second one to  $g_{ij}^2$  etc., so the last pattern corresponds to the root gate of  $\varphi$ . The last node of this last pattern is labelled by  $D$ .
- (c) For each AND-gate  $g$  of the formula  $\varphi_\psi$ , we add a fresh node  $w_g$  labelled by  $E$  to  $\mathbf{q}^{var}$ . For every RSR-pattern corresponding to some occurrence  $g_{ij}^\ell$  of  $g$ , we add an edge from the end-node of the  $S$ -edge of the pattern to  $w_g$  (and so occurrences corresponding to the two inputs of the same AND-gate are ‘connected’).

It is not hard to see that  $|\mathbf{q}|$  is polynomial in  $|\varphi_\psi|$ , and so in  $|\psi|$ . Property (11) holds because the  $F$ -node does have  $S$ -successors, while the  $FT$ -node does not.

#### 4.5. Proof of Lemma 14.2

Fix some unpruned path-cactus  $C \in \mathcal{R}_Q$  and a segment  $\mathfrak{s}$  of co-depth  $\geq 4k + 4$  in  $C^s$ , and let  $P_{\mathfrak{s}, C}^{4k+4}$  be the  $4k + 4$ -long  $FT$ -sequence starting at the child of  $\mathfrak{s}$  in  $C^s$ .

(i) ( $\Leftarrow$ ) and (ii): Suppose  $h: \mathbf{q}^- \rightarrow C$  is a homomorphism mapping  $\mathbf{q}^-$  into  $\mathfrak{s}$ . By (11), the centre of  $\mathbf{q}^-$  should be mapped by  $h$  to the centre of  $\mathfrak{s}$  (which is node  $r$  in its base block  $\mathbf{q}^{base}$ ). Since one of the nodes  $t_F$  or  $t_T$  in  $\mathfrak{s}$  is the bud  $t_i$  (and so not labelled by  $T$  in  $\mathfrak{s}$ ), we have  $h(t_i) = \theta$  for the  $FT$ -node  $\theta$  of  $\mathfrak{s}$ . As node  $u$  has a common successor with both  $t_F$  and  $t_T$ , and it also has (many) two-step successors (in  $\mathbf{q}^{var}$ ), we have  $h(u) = t_i$ . Observe that, for the unbudded  $T$ -node  $t_j$  in  $\mathfrak{s}$ , we have a choice:  $h(t_j)$  might be either  $t_j$  or  $\theta$ . We obtain (ii) by choosing the former.

Next, we examine the  $h$ -image of the descendants of  $u$  in  $\mathbf{q}^{var}$ . As  $n = 4k + 4$ , we consider the  $n$ -long  $FT$ -sequence  $P_{\mathfrak{s}, C}^{4k+4}$  as an assignment for the variables  $x_1, \dots, x_n$  in  $\varphi_\psi$ . As branch  $\mathbf{b}_{n+1}$  in  $\mathbf{q}^{var}$  contains two  $R$ -consecutive  $B$ -nodes at  $R$ -distance  $n + 1$  from  $u$ ,  $h$  must map these two  $B$ -nodes to the two  $B$ -nodes in the  $\mathbf{q}^{base}$ -block of the segment  $\mathfrak{s}'$  of  $C$  at the end of  $P_{\mathfrak{s}, C}^{4k+4}$ . Thus,  $h(z)$  must be the common  $R$ -successor of these two  $B$ -nodes in  $\mathfrak{s}'$ . Therefore, the  $B$ -node in  $\mathbf{b}_i$ , for every  $i \leq n$ , must also be mapped to one of the two  $B$ -nodes in the  $\mathbf{q}^{base}$ -block of  $\mathfrak{s}'$ . However, which of these two  $B$ -nodes is the image depends on the truth-value of  $P_{\mathfrak{s}, C}^{4k+4}$  on  $x_i$ :

- (a) If  $P_{\mathfrak{s}, C}^{4k+4}$  maps  $x_i$  to  $F$ , then the  $i$ th application of **(bud)** in  $P_{\mathfrak{s}, C}^{4k+4}$  is at some copy of  $t_F$  that is reachable from  $r$  via an  $S$ -step. So the  $B$ -node in  $\mathbf{b}_i$  is mapped to the lower  $B$ -node  $v$  in the  $\mathbf{q}^{base}$ -block of  $\mathfrak{s}'$ .
- (b) If  $P_{\mathfrak{s}, C}^{4k+4}$  maps  $x_i$  to  $T$ , then the  $i$ th application of **(bud)** in  $P_{\mathfrak{s}, C}^{4k+4}$  is at a copy of  $t_T$  that is reachable from  $r$  via an  $S$ -step followed by an  $R$ -step only. So the  $B$ -node in  $\mathbf{b}_i$  is mapped to the upper  $B$ -node in the  $\mathbf{q}^{base}$ -block of  $\mathfrak{s}'$ .

It remains to see how  $h$  maps the remaining part of  $\mathbf{q}^{var}$  into the  $\mathbf{q}^{form}$ -block of  $\mathfrak{s}'$ . We claim that for every non-leaf gate  $g$  in  $\varphi_\psi$ , if  $g_{ij}^\ell$  is an occurrence of  $g$  on some branch, then the end-node  $p_{ij}^\ell$  of the  $RSR$ -pattern corresponding to  $g_{ij}^\ell$  in  $\mathbf{q}^{var}$  is mapped to the  $\mathbf{q}^{form}$ -block of  $\mathfrak{s}'$  in such a way that

- (c)  $h(p_{ij}^\ell)$  is the  $\mathbf{o}$ -node of the gadget for  $g$  whenever the value of  $g$  under  $P_{\mathfrak{s}, C}^{4k+4}$  is  $F$ ;
- (d)  $h(p_{ij}^\ell)$  is the  $(D)$ -node of the gadget for  $g$  whenever the value of  $g$  under  $P_{\mathfrak{s}, C}^{4k+4}$  is  $T$ .

We prove this by induction on the tree-structure of  $\varphi_\psi$ , going from leaves to root. Consider an occurrence  $g_{ij}^\ell$  of a gate  $g$ . Suppose first that  $g$  is a NOT-gate.

Case  $\ell = 1$ . If the value of  $g$  under  $P_{\mathfrak{s}, C}^{4k+4}$  is  $F$ , then  $P_{\mathfrak{s}, C}^{4k+4}$  maps  $x_i$  to  $T$ , and so by (b) the  $B_{ij}$ -node of branch  $\mathbf{b}_i^j$  is mapped by  $h$  to the upper  $B_{ij}$ -node in the  $\mathbf{q}^{form}$ -block of  $\mathfrak{s}'$ . So the first  $R$ -edge of the  $RSR$ -pattern corresponding to  $g_{ij}^1$  is mapped to the  $R$ -edge connecting the two  $B_{ij}$ -nodes. And then the subsequent  $S$ - and  $R$ -edges of the pattern must be mapped to the right-hand side of the  $g$ -gadget starting from its  $\mathbf{i}$ -node. So  $h(p_{ij}^1)$  is its  $\mathbf{o}$ -node. If

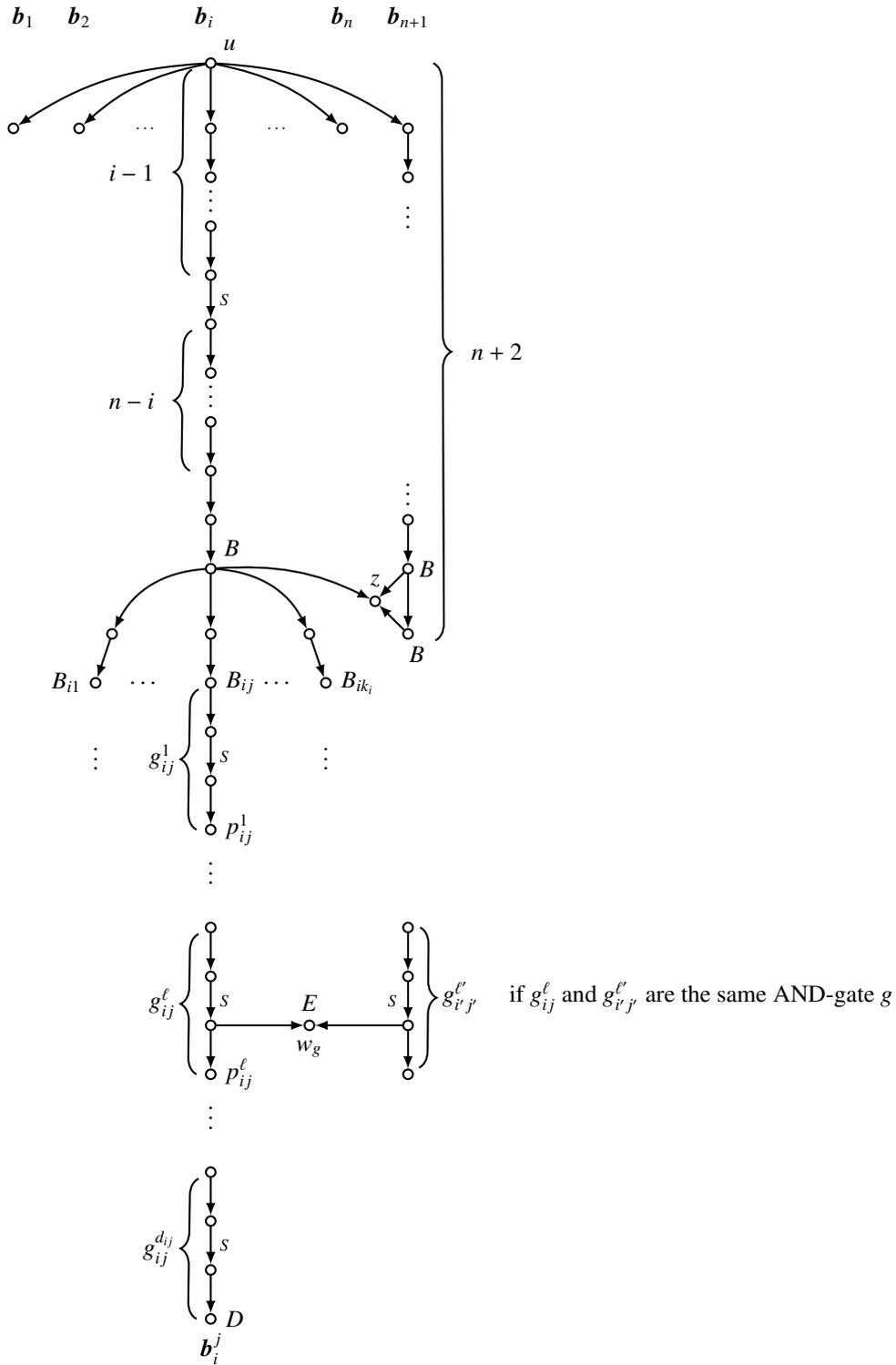


Figure 3: Block  $q^{var}$  of the 1-CQ  $q$ .

the value of  $g$  under  $P_{s,C}^{4k+4}$  is  $T$ , then  $P_{s,C}^{4k+4}$  maps  $x_i$  to  $F$ , and so by (a) the  $B_{ij}$ -node of branch  $b_i^j$  is mapped by  $h$  to the lower  $B_{ij}$ -node in the  $q^{form}$ -block of  $s'$ . So the whole  $RSR$ -pattern corresponding to  $g_{ij}^1$  must be mapped to the left-hand side of the  $g$ -gadget starting from its  $i$ -node, and so  $h(p_{ij}^1)$  is its  $(D)$ -node (otherwise the mapping cannot be ‘continued’ when  $g_{ij}^1$  is a non-root gate, and  $h$  would not preserve  $D$  when  $g_{ij}^1$  is the root-gate).

Case  $\ell > 1$ . Then  $g_{ij}^{\ell-1}$  is an occurrence of the non-leaf gate  $g^-$  that is the input of  $g$ . So if the value of  $g$  under  $P_{s,C}^{4k+4}$  is  $F$ , then the value of  $g^-$  under  $P_{s,C}^{4k+4}$  is  $T$ . By IH,  $h(p_{ij}^{\ell-1})$  is the  $(D)$ -node of the gadget for  $g^-$ . Thus, the first  $R$ -edge of the  $RSR$ -pattern corresponding to  $g_{ij}^\ell$  is mapped to the  $R$ -edge connecting the  $(D)$ - and  $\mathbf{o}$ -nodes of the  $g^-$ -gadget. And then the subsequent  $S$ - and  $R$ -edges of the pattern must be mapped to the right-hand side of the  $g^-$ -gadget. So  $h(p_{ij}^\ell)$  is its  $\mathbf{o}$ -node. If the value of  $g$  under  $P_{s,C}^{4k+4}$  is  $T$ , then the value of  $g^-$  under  $P_{s,C}^{4k+4}$  is  $F$ . Then by the IH,  $h(p_{ij}^{\ell-1})$  is the  $\mathbf{o}$ -node of the gadget for  $g^-$ . So the whole  $RSR$ -pattern corresponding to  $g_{ij}^\ell$  must be mapped to the left-hand side of the  $g$ -gadget starting from its  $i$ -node, and so  $h(p_{ij}^\ell)$  is its  $(D)$ -node.

Now suppose  $g$  is an AND-gate. There are many cases, depending on the truth-values of  $g$  and its inputs  $g_1$  and  $g_2$  under  $P_{s,C}^{4k+4}$ , and on whether each of the  $g_i$  is a leaf gate or not. We consider two of them as the other ones are similar.

- Suppose the value of  $g$  under  $P_{s,C}^{4k+4}$  is  $F$ ,  $\ell = 1$  (and so  $g_1$  is a leaf labelled by  $x_i$ ), and  $P_{s,C}^{4k+4}$  maps  $x_i$  to  $T$ . Suppose  $g_2$  is also a leaf gate, and so  $g_2$  has value  $F$  under  $P_{s,C}^{4k+4}$ . Let  $g_{i'j'}$  be an occurrence of  $g_2$ . By (b), the  $B_{ij}$ -node of branch  $b_i^j$  is mapped by  $h$  to the upper  $B_{ij}$ -node in the  $q^{form}$ -block of  $s'$ . So the first  $R$ -edge of the  $RSR$ -pattern corresponding to  $g_{ij}^1$  is mapped to the  $R$ -edge connecting the two  $B_{ij}$ -nodes. Thus, the  $S$ -edge of the  $RSR$ -pattern corresponding to  $g_{ij}^1$  must be mapped to an  $S$ -edge starting at the  $i_1$ -node of the  $g$ -gadget. Similarly, by (a), the  $B_{i'j'}$ -node of branch  $b_{i'}^{j'}$  is mapped by  $h$  to the lower  $B_{i'j'}$ -node in the  $q^{form}$ -block of  $s'$ . So the  $S$ -edge of the  $RSR$ -pattern corresponding to  $g_{i'j'}^1$  must be mapped to an  $S$ -edge following an  $R$ -edge starting at the  $i_2$ -node of the  $g$ -gadget. As  $h$  preserves  $E$ , the end-nodes of these two  $S$ -edges in the  $g$ -gadget must coincide, and so it must be node  $c_1$ . So  $h(p_{ij}^1)$  is the  $\mathbf{o}$ -node of the  $g$ -gadget.
- Suppose the value of  $g$  under  $P_{s,C}^{4k+4}$  is  $T$ , and both of its inputs are non-leaf gates having value  $T$  under  $P_{s,C}^{4k+4}$ . Suppose  $g_{ij}^{\ell-1}$  is an occurrence of  $g_1$  and  $g_{i'j'}^{\ell-1}$  is an occurrence of  $g_2$ . By IH,  $h(p_{ij}^{\ell-1})$  is the  $(D)$ -node of the gadget for  $g_1$ , and  $h(p_{i'j'}^{\ell-1})$  is the  $(D)$ -node of the gadget for  $g_2$ . Then the  $S$ -edges of the  $RSR$ -patterns corresponding to  $g_{ij}^{\ell-1}$  and  $g_{i'j'}^{\ell-1}$  must be mapped, respectively, to  $S$ -edges starting at the  $i_1$ - and  $i_2$ -nodes of the  $g$ -gadget. As  $h$  preserves  $E$ , the end-nodes of these two  $S$ -edges in the  $g$ -gadget must coincide, and so it must be node  $b$ . Thus,  $h(p_{ij}^\ell)$  is the  $(D)$ -node of the  $g$ -gadget, as required.

This completes the proof of (c) and (d). As  $h$  preserves  $D$ , it follows that  $\varphi_\psi(P_{s,C}^{4k+4}) = T$ .

(i) ( $\Rightarrow$ ): If  $\varphi_\psi(P_{s,C}^{4k+4}) = T$ , then we define a function  $h: q^- \rightarrow C$  by taking  $h(t) = \theta$ , where the copy of  $t$  in  $s$  is budded, and  $\theta$  is the  $FT$ -node in  $s$ ;  $h(u) = t$  for the budded  $T$ -node  $t$  in  $s$ ; mapping the descendants of  $u$  in  $q^{var}$  to segments subsequent to  $s$  in  $C$  following the structure of  $\varphi_\psi$ , as described in the ( $\Leftarrow$ ) direction above; and mapping any other nodes in  $q^-$  to their own copies in  $s$ . It is easy to see that  $h$  is a homomorphism mapping  $q^-$  into  $s$ .

## 5. Linear-Datalog-Rewritability of OMQs with a 1-CQ

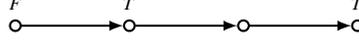
We next obtain a sufficient semantic condition of linear-datalog-rewritability of OMQs  $Q = (\mathcal{T}, q)$  with a 1-CQ  $q$ . The branching number [28] of a rooted tree  $\mathfrak{T}$  is defined as follows. For any node  $u$  in  $\mathfrak{T}$ , we define inductively its *branching rank*  $br(u)$  by taking  $br(u) = 0$  if  $u$  is a leaf and, for a non-leaf  $u$ ,

$$br(u) = \begin{cases} m + 1, & \text{if } u \text{ has } \geq 2 \text{ children of branching rank } m; \\ m, & \text{otherwise.} \end{cases} \quad (13)$$

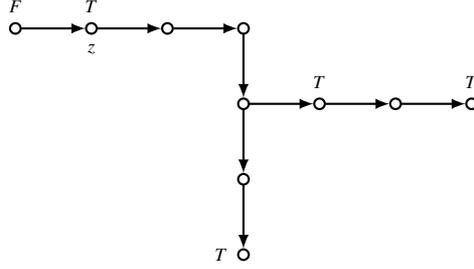
Then the *branching number* of  $\mathfrak{T}$  is the branching rank of its root node. (In other words, the branching number of  $\mathfrak{T}$  is  $b$  if the largest full binary tree that is a minor of  $\mathfrak{T}$  is of depth  $b$ .) The *branching number of a cactus*  $C \in \mathfrak{R}_Q$  is the branching number of  $C^s$ .

A cactus  $C \in \mathfrak{R}_Q$  is called *minimal* if we cannot apply (**prune**) to it (that is, if by omitting any of the  $T$ -labels from  $C$ , the resulting ABox  $C^-$  is such that  $\mathcal{T}, C^- \not\models q$ ). Let  $\mathfrak{R}_Q^{\min}$  be the set of minimal cactuses in  $\mathfrak{R}_Q$ . We say that  $\mathfrak{R}_Q^{\min}$  is *boundedly branching* if there is some  $b < \omega$  such that  $\mathfrak{R}_Q^{\min}$  contains a cactus with branching number  $b$  but no cactus of greater branching number. Otherwise, we call  $\mathfrak{R}_Q^{\min}$  *unboundedly branching*.

**Example 15.** Consider  $Q = (\text{cov}_\top, q_{FT.T})$  with  $q_{FT.T}$  depicted below (the omitted labels on the edges are all  $R$ ):



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



Clearly,  $\text{cov}_\top, C \setminus \{T(z)\} \models q_{FT.T}$ , and so (**prune**) would remove  $T(z)$  from  $C$ . Using this fact, one can show that every cactus in  $\mathfrak{R}_Q^{\min}$  has branching number  $\leq 1$ . On the other hand, if  $Q = (\text{cov}_A, q_{FT.T})$  then  $\mathfrak{R}_Q^{\min}$  is unboundedly branching by Theorems 16 and 20.

**Theorem 16.** For any OMQ  $Q = (\mathcal{T}, q)$  with a 1-CQ  $q$ , if  $\mathfrak{R}_Q^{\min}$  is boundedly branching, then  $Q$  is linear-datalog-rewritable, and so is in NL.

*Proof.* Similarly to [32], we represent cactus-like ABoxes as terms of a tree alphabet and construct a tree automaton  $\mathfrak{A}_Q$  such that (i) cactuses in  $\mathfrak{R}_Q^{\min}$  are accepted by  $\mathfrak{A}_Q$ , and (ii) for every ABox  $\mathcal{A}$  accepted by  $\mathfrak{A}_Q$ , we have  $\mathcal{T}, \mathcal{A} \models q$ . Then, using ideas of [28], we show that if  $\mathfrak{R}_Q^{\min}$  is boundedly branching, then the automaton  $\mathfrak{A}_Q$  can be transformed into a (monadic) linear-stratified datalog rewriting of  $Q$ . As shown in [38], 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).

We only consider the case  $\mathcal{T} = \text{cov}_A$ , as the case when  $\mathcal{T} = \text{cov}_A^\perp$  is similar. Recall from [56] that a *tree alphabet* is a finite set  $\Sigma$  of symbols, each of which is associated with a natural number, its *arity*. A  $\Sigma$ -tree is any ground term built up inductively, using the symbols of  $\Sigma$  as functions: 0-ary symbols in  $\Sigma$  are  $\Sigma$ -trees and, for any  $k$ -ary  $\alpha$  in  $\Sigma$  and  $\Sigma$ -trees  $C_1, \dots, C_k$ , the term  $\alpha(C_1, \dots, C_k)$  is a  $\Sigma$ -tree. We define a tree alphabet  $\Sigma_Q$  as follows. Consider cactus-like ABoxes that are built from  $q$  using (**bud**) and (**prune**), with applications of the latter also being allowed when  $\text{cov}_A, C^- \not\models q$  for the resulting ABox  $C^-$ . The symbols of  $\Sigma_Q$  are the segments  $s$  of such cactus-like ABoxes, with the arity of  $s$  being the number of its budding nodes, and with the  $x$ -node of  $s$  being either labelled by  $F$  or not. Then each cactus in  $\mathfrak{R}_Q$  can be encoded by some  $\Sigma_Q$ -tree; see Fig. 4 for an example. On the other hand, every  $\Sigma_Q$ -tree represents some cactus-like ABox. So, with a slight abuse of terminology, from now on by a  $\Sigma_Q$ -tree we mean either the term or the corresponding ABox.

However, such an ABox  $C$  is not necessarily a cactus for two kinds of reasons: either  $\text{cov}_A, C \not\models q$  or  $C$  might have  $F$ -nodes in some ‘wrong’ segments (in every cactus, there is a unique  $F$ -node: the  $x$  node of its root segment). We are interested in those  $\Sigma_Q$ -trees  $C$  for which  $\text{cov}_A, C \models q$ . To capture them, we use tree automata [56]. A *nondeterministic finite tree automaton* (NTA) over a tree alphabet  $\Sigma$  is a quadruple  $\mathfrak{A} = (Q, Q_f, \Delta, \Sigma)$ , where

- $Q$  is a finite set of *states*,
- $Q_f \subseteq Q$  is a set of *final states*, and
- $\Delta$  is a set of *transitions* of the form  $q_1, \dots, q_k \Rightarrow^a q$ , where  $k$  is the arity of  $\alpha \in \Sigma$  and  $q_1, \dots, q_k, q \in Q$ ; for symbols  $\alpha$  of arity 0, we might have *initial transitions* of the form  $\Rightarrow^a q$ .

A *run* of  $\mathfrak{A}$  on a  $\Sigma$ -tree  $C$  is a labelling function  $r$  from the subterms of  $C$  to  $Q$  satisfying the following condition: for any subterm  $C^- = \alpha(C_1, \dots, C_k)$  of  $C$ , there is a transition  $q_1, \dots, q_k \Rightarrow^a q$  in  $\Delta$  such that  $r(C_1) = q_1, \dots, r(C_k) = q_k$

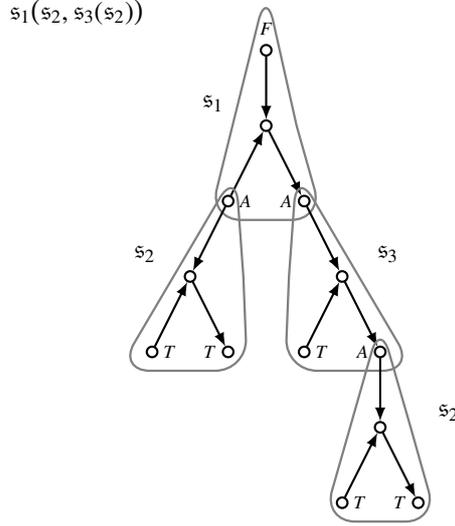
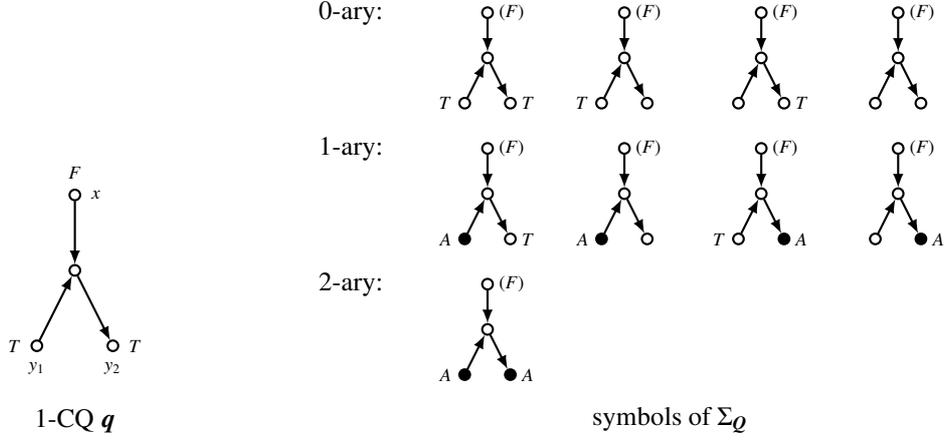


Figure 4: An example of a tree alphabet  $\Sigma_Q$ , and a cactus as a  $\Sigma_Q$ -tree.

and  $r(C^-) = q$  (in which case we say that the transition is *used in*  $r$ ). A  $\Sigma$ -tree  $C$  is *accepted* by  $\mathfrak{A}$  if there is a run of  $\mathfrak{A}$  on  $C$  that labels  $C$  with a final state. Let  $L(\mathfrak{A})$  be the set of all  $\Sigma$ -trees accepted by  $\mathfrak{A}$ . A set  $L$  of  $\Sigma$ -trees is called a *regular tree language* if  $L = L(\mathfrak{A})$ , for some NTA  $\mathfrak{A}$  over  $\Sigma$ .

**Lemma 16.1.**  $L_Q = \{C \mid C \text{ is a } \Sigma_Q\text{-tree with } \text{cov}_A, C \models q\}$  is a regular tree language.

*Proof.* We proceed via a series of steps. Suppose  $q$  is a 1-CQ such that  $F(x)$  and  $T(y_1), \dots, T(y_n)$  are all of the solitary occurrences of  $F$  and  $T$  in  $q$ . We want to use Theorem 9 for describing  $L_Q$ . Recall the datalog program  $\Pi_Q$  given by (6)–(8) above. We extend the tree alphabet  $\Sigma_Q$  to a tree alphabet  $\Sigma_Q^e$  as follows. For each symbol  $s$  in  $\Sigma_Q$ , we label some (possibly none) of the nodes in segment  $s$  by  $P$ . We call each resulting ‘segment’  $s^e$  an *extension of*  $s$ . (Each symbol in  $\Sigma_Q$  might have several extensions, and each of them has the same arity as  $s$ .) Let  $\Sigma_Q^e$  consist of all possible extensions of every  $s$  in  $\Sigma_Q$ . We say that a  $\Sigma_Q^e$ -tree  $C^e$  is an *extension* of a  $\Sigma_Q$ -tree  $C$  if they have isomorphic tree structures, and each symbol  $s^e$  in  $C^e$  is an extension of the corresponding symbol  $s$  in  $C$ . For example, the closure  $\Pi_Q(C)$  of any  $\Sigma_Q$ -tree  $C$  under  $\Pi_Q$  is an extension of  $C$ .

For any  $\Sigma_Q^e$ -tree  $C^e$ , we write  $C^e \models G$ , for the goal predicate  $G$  of  $\Pi_Q$ , if there is a homomorphism from  $q^e$  to  $C^e$ , where  $q^e = q \setminus \{T(y_1), \dots, T(y_n)\} \cup \{P(y_1), \dots, P(y_n)\}$ . We claim that each of the following is a regular tree language:

- (a) the set of  $\Sigma_Q^e$ -trees  $C^e$  with  $C^e \neq \Pi_Q(C^e)$ ;
- (b) the set of  $\Sigma_Q^e$ -trees  $C^e$  with  $C^e \models G$ ;
- (c) the set of  $\Sigma_Q$ -trees  $C$  that have some extension  $C^e$  with  $C^e = \Pi_Q(C^e)$  and  $C^e \not\models G$ ;
- (d) the set of  $\Sigma_Q$ -trees  $C$  with  $\Pi_Q, C \models G$ .

Indeed, to show (a), we need an NTA ‘detecting a pattern’ in the ABox  $C^e$  falsifying one of rules (7)–(8) in  $\Pi_Q$ . Similarly, to show (b), we need an NTA ‘detecting a pattern’ in  $C^e$  corresponding to an application of rule (6) in  $\Pi_Q$ . Now, (c) follows from (a), (b) and the fact that regular tree languages are closed under taking complements, intersections and linear homomorphisms [56] (as the ‘forgetting’ function substituting  $\mathfrak{s}$  for each  $\mathfrak{s}^e$  is a linear tree homomorphism from  $\Sigma_Q^e$ -trees to  $\Sigma_Q$ -trees, mapping any extension  $C^e$  to  $C$ .) To show (d), take the complement of (c), and observe that  $\Pi_Q, C \models G$  iff, for every extension  $C^e$  of  $C$ , whenever  $C^e = \Pi_Q(C^e)$  then  $C^e \models G$ .

Finally, it follows from (d) and Theorem 9 that  $L_Q$  is a regular tree language.  $\square$

An NTA  $\mathfrak{A} = (Q, Q_f, \Delta, \Sigma)$  is *stratified* if there is a function  $st: Q \rightarrow \omega$  such that, for any transition  $q_1, \dots, q_k \Rightarrow^a q$  in  $\Delta$ ,

- $st(q_i) \leq st(q)$ , for every  $i$ ,  $1 \leq i \leq k$ , and
- there is at most one  $i$  such that  $1 \leq i \leq k$  and  $st(q_i) = st(q)$ .

**Lemma 16.2.** *For any NTA  $\mathfrak{A}$  and any  $\mathbf{b} < \omega$ , there is a stratified NTA  $\mathfrak{A}^s$  such that*

$$\{C \in L(\mathfrak{A}) \mid \text{the branching number of } C \text{ is } \leq \mathbf{b}\} \subseteq L(\mathfrak{A}^s) \subseteq L(\mathfrak{A}). \quad (14)$$

*Proof.* Suppose  $\mathfrak{A} = (Q, Q_f, \Delta, \Sigma)$ . We define  $\mathfrak{A}^s = (Q^s, Q_f^s, \Delta^s, \Sigma)$  as follows. First, set  $Q^s = Q \times \{0, \dots, \mathbf{b}\}$  and  $Q_f^s = Q_f \times \{0, \dots, \mathbf{b}\}$ . Then, for any transition of the form  $\Rightarrow^a q$  in  $\Delta$ , we add the transition  $\Rightarrow^a (q, 0)$  to  $\Delta^s$ . For any transition  $q_1, \dots, q_k \Rightarrow^a q$  in  $\Delta$  and any  $m \leq \mathbf{b}$ , we add to  $\Delta^s$  all transitions  $(q_1, m_1), \dots, (q_k, m_k) \Rightarrow^a (q, m)$  such that

- either  $m_1, \dots, m_k < m$  and  $m_i = m_j = m - 1$ , for some  $i \neq j$ ;
- or  $m_i = m$ , for some  $i$ , and  $m_j < m$ , for all  $j \neq i$ .

$\mathfrak{A}^s$  is stratified as one can set  $st((q, m)) = m$ , for  $q \in Q$ ,  $m \leq \mathbf{b}$ . To show (14), observe that  $L(\mathfrak{A}^s) \subseteq L(\mathfrak{A})$  since from every run  $r$  of  $\mathfrak{A}^s$  on  $C$  we obtain a run of  $\mathfrak{A}$  on  $C$  by replacing each transition  $(q_1, m_1), \dots, (q_k, m_k) \Rightarrow^a (q, m)$  used in  $r$  with  $q_1, \dots, q_k \Rightarrow^a q$ . For the other inclusion, given a run  $r$  of  $\mathfrak{A}$  on some  $C$  with branching number  $\leq \mathbf{b}$ , we obtain a run of  $\mathfrak{A}^s$  on  $C$  by labelling each subterm  $C^-$  of  $C$  with state  $(r(C^-), \mathbf{b}^-)$ , where  $\mathbf{b}^-$  is the branching number of  $C^-$ .  $\square$

We can now complete the proof of Theorem 16. Indeed, suppose that every cactus in  $\mathfrak{R}_Q^{min}$  has branching number  $\leq \mathbf{b} < \omega$ . By Lemmas 16.1 and 16.2, there is a stratified NTA  $\mathfrak{A} = (Q, Q_f, \Delta, \Sigma_Q)$  such that

$$\{C \in L_Q \mid \text{the branching number of } C \text{ is } \leq \mathbf{b}\} \subseteq L(\mathfrak{A}) \subseteq L_Q.$$

Using  $\mathfrak{A}$ , we construct a (monadic) linear-stratified program  $\Pi_{\mathfrak{A}}$  with goal predicate  $\mathbf{G}_{\mathfrak{A}}$  as follows. For every  $q \in Q$ , we introduce a fresh unary predicate  $P_q$ . For every final state  $q \in Q_f$ ,  $\Pi_{\mathfrak{A}}$  contains the rule

$$\mathbf{G}_{\mathfrak{A}} \leftarrow P_q(x). \quad (15)$$

For every transition  $q_1, \dots, q_k \Rightarrow^s q$  in  $\Delta$ , where the budding nodes in the  $k$ -ary segment  $\mathfrak{s}$  are  $y_{i_1}, \dots, y_{i_k}$ ,  $\Pi_{\mathfrak{A}}$  contains

$$P_q(x) \leftarrow \mathfrak{s}, P_{q_1}(y_{i_1}), \dots, P_{q_k}(y_{i_k}). \quad (16)$$

As  $\mathfrak{A}$  is stratified, it is easy to see that the program  $\Pi_{\mathfrak{A}}$  is linear-stratified. We claim that  $(\Pi_{\mathfrak{A}}, \mathbf{G}_{\mathfrak{A}})$  is a datalog-rewriting of  $Q$ , that is, for any ABox  $\mathcal{A}$  (without the  $P_q$ ), we have  $\Pi_{\mathfrak{A}}, \mathcal{A} \models \mathbf{G}_{\mathfrak{A}}$  iff  $\text{cov}_{\mathcal{A}}, \mathcal{A} \models q$ .

( $\Leftarrow$ ) By Theorem 9, there is a homomorphism  $h: C \rightarrow \mathcal{A}$ , for some  $C \in \mathfrak{R}_Q$ . Clearly, we may assume  $C \in \mathfrak{R}_Q^{min}$ , and so  $C$  has branching number  $\leq \mathbf{b}$ . As  $\text{cov}_{\mathcal{A}}, C \models q$  by (10), it follows that  $C \in L_Q$ , and so  $C \in L(\mathfrak{A})$ . Let  $r$  be

an accepting run of  $\mathfrak{A}$  on  $C$ . We construct a derivation of  $G_{\mathfrak{A}}$  in  $\Pi_{\mathfrak{A}}(\mathcal{A})$  by induction on  $C$  as a  $\Sigma_Q$ -tree, moving from leaves to the root. For every segment  $\mathfrak{s}$  in  $C$ , if the transition  $q_1, \dots, q_k \Rightarrow^{\mathfrak{s}} q$  is used in  $r$  then we apply (16) with the substitution of  $h(z)$  for any node  $z$  in  $\mathfrak{s}$ . Also, if  $r(C) = q$ , for some final state  $q$  of  $\mathfrak{A}$ , then we apply (15) with the substitution  $h(x_{\mathfrak{s}_0})$ , where  $x_{\mathfrak{s}_0}$  is the  $x$ -node of the root segment  $\mathfrak{s}_0$  in  $C$ . It follows that  $\Pi_{\mathfrak{A}}, \mathcal{A} \models G_{\mathfrak{A}}$ .

( $\Rightarrow$ ) By induction on a derivation of  $G_{\mathfrak{A}}$ , we construct a  $\Sigma_Q$ -tree  $\mathcal{B}$ , an accepting run  $r$  of  $\mathfrak{A}$  on  $\mathcal{B}$ , and a homomorphism  $f: \mathcal{B} \rightarrow \mathcal{A}$ . To begin with, there is an object  $x^a$  for which (15) was triggered for some  $q \in Q_f$ . Then  $P_q(x^a)$  was deduced by an application of (16) for some  $\mathfrak{s}$ . If this  $\mathfrak{s}$  is 0-ary, then  $\mathfrak{s}$  is a  $\Sigma_Q$ -tree (of depth 0), the function  $r$  labelling  $\mathfrak{s}$  with  $q$  is an accepting run on  $\mathfrak{s}$ , and the substitution  $f_0$  used in (16) is a homomorphism from  $\mathfrak{s}$  to  $\mathcal{A}$ . If  $\mathfrak{s}$  is  $k$ -ary, for some  $k > 0$ , then there are  $y_{i_1}^a, \dots, y_{i_k}^a$  for which (16) was triggered. For each  $j = 1, \dots, k$ , consider the rule

$$P_{q_j}(x) \leftarrow \mathfrak{s}^j, P_{q_1^j}(y_{i_1}), \dots, P_{q_k^j}(y_{i_k})$$

by which  $P_{q_j}(y_{i_j}^a)$  was deduced. Take the ABox  $\mathcal{B}$  built up by glueing the  $x$  node of each segment  $\mathfrak{s}^j$  to the  $y_{i_j}$  node of  $\mathfrak{s}$ , extend  $r$  by labelling each  $\mathfrak{s}^j$  with  $q_j$ , and extend  $f_0$  to a  $\mathcal{B} \rightarrow \mathcal{A}$  homomorphism by taking the substitutions used in the rules. Now, if every  $\mathfrak{s}^j$  is 0-ary, then  $\mathcal{B}$  is a  $\Sigma_Q$ -tree and we are done. Otherwise, repeat the above procedure for the ‘arguments’ of each  $\mathfrak{s}^j$  of arity  $> 0$ . As the derivation of  $G_{\mathfrak{A}}$  is finite, sooner or later the procedure stops, as required.

As  $\mathcal{B} \in L(\mathfrak{A}) \subseteq L_Q$ , by Theorem 9 there exists a homomorphism  $h: C \rightarrow \mathcal{B}$ , for some cactus  $C \in \mathfrak{R}_Q$ . Then the composition of  $h$  and  $f$  is a homomorphism from  $C$  to  $\mathcal{A}$ , and so we  $\text{cov}_A, \mathcal{A} \models \mathbf{q}$  by Theorem 9, as required.  $\square$

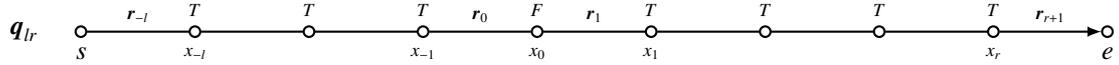
We do not know whether this sufficient condition of linear-datalog-rewritability of OMQs with a 1-CQ is also a necessary one. As follows from [28, 53], it is the case for ditree 1-CQs with root labelled by  $F$ ; see Example 11.

## 6. $AC^0/NL/P/\text{coNP}$ -Tetrachotomy of OMQs with a Path CQ

In this section, we focus on the OMQs  $(\text{cov}_A, \mathbf{q})$  with a twinless path CQ  $\mathbf{q}$ . So from now on, solitary  $F$ -nodes ( $T$ -nodes) in  $\mathbf{q}$  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 type.

We begin by dividing twinless path CQs 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  $\mathbf{q}$  by  $s$  and the last (leaf) node by  $e$ . We write  $x \leq y$  to say that there is a path from  $x$  to  $y$  in  $\mathbf{q}$ , and  $x < y$  whenever  $x \leq y$  and  $x \neq y$ . If  $x \leq y$ , then  $[x, y]$  comprises those atoms in  $\mathbf{q}$  whose variables are in the interval  $\{z \mid x \leq z \leq y\}$ ; 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  $x_{-l} < \dots < x_{-1} < x_0 < x_1 < \dots < x_r$  be all the  $F$ - or  $T$ -nodes in  $\mathbf{q}$ , with  $x_0$  being the only  $F$ -node and  $l + r \geq 1$ . We denote this 1-CQ by  $\mathbf{q}_{lr}$ . Let  $\mathbf{r}_i = (x_{i-1}, x_i)$ , where  $x_{-l-1} = s$  and  $x_{r+1} = e$ .



We write  $\mathbf{r}_i \rightsquigarrow \mathbf{r}_j$  if there is a homomorphism  $h: \mathbf{r}_i \rightarrow \mathbf{r}_j$  with  $h(x_{i-1}) = x_{j-1}$  and  $h(x_i) = x_j$ . We call  $\mathbf{q}_{lr}$  *right-periodic* if  $l = 0$  and  $\mathbf{r}_i \rightsquigarrow \mathbf{r}_1$  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  $\mathbf{r}_{-i} \rightsquigarrow \mathbf{r}_0$  for all  $i = 1, \dots, l$ . A 1-CQ  $\mathbf{q}$  is *periodic* if it is either right- or left-periodic, and *non-periodic* otherwise. (Similar notions can be defined for 1-CQs with a single  $T$ -node and at least one  $F$ -node.)

**Theorem 17 (tetrachotomy).** *For any  $\mathbf{Q} = (\text{cov}_A, \mathbf{q})$  with a twinless path CQ  $\mathbf{q}$ , the following hold:*

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

Item (1) is shown in Theorem 4 (i). The other two upper bounds follow from Corollary 10 and Theorem 19. The lower bounds follow from Theorems 18, 20 and 21.

We begin with the following criterion:

**Theorem 18.** *If  $q$  is a twinless path 1-CQ, then  $(\text{cov}_A, q)$  and  $(\text{cov}_A^\perp, q)$  are 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  $q$  with  $x < y$  (the other case is symmetric) and without any  $F$ - or  $T$ -nodes between them. Given a digraph  $G = (V, E)$  with nodes  $s, t \in V$ , we construct an ABox  $\mathcal{A}_G$  as follows. Replace each edge  $e = (u, v) \in E$  by a fresh copy  $q^e$  of  $q$  such that node  $x$  in  $q^e$  is renamed to  $u$  with  $T(u)$  being replaced by  $A(u)$ , and node  $y$  is renamed to  $v$  with  $F(v)$  being replaced by  $A(v)$ . Then  $\mathcal{A}_G$  comprises all such  $q^e$ , for  $e \in E$ , as well as  $T(s)$  and  $F(t)$ . We show that  $s \rightarrow_G t$  iff  $\text{cov}_A, \mathcal{A}_G \models q$ .

( $\Rightarrow$ ) Suppose there is a path  $s = v_0, \dots, v_n = t$  in  $G$  with  $e_i = (v_i, v_{i+1}) \in E$ , for  $i < n$ . Then, for any model  $\mathcal{I}$  of  $\text{cov}_A$  and  $\mathcal{A}_G$ , there is some  $i < n$  such that  $v_i \in T^{\mathcal{I}}$  and  $v_{i+1} \in F^{\mathcal{I}}$ . Thus, the identity map from  $q$  to its copy  $q^{e_i}$  is a  $q \rightarrow \mathcal{I}$  homomorphism, and so  $\mathcal{I} \models q$ .

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

By Corollary 10, all OMQs  $Q$  with a 1-CQ  $q$  are datalog-rewritable and lie in P. Our next task is to show that every such OMQ with a twinless path  $q$  is either linear-datalog-rewritable, and so NL-complete, or P-hard.

**Theorem 19.** *If  $q$  is a periodic twinless path 1-CQ, then  $Q = (\text{cov}_A, q)$  is linear-datalog-rewritable, and so lies in NL.*

*Proof.* It is not hard to either construct an explicit linear datalog-rewriting of  $Q$  or show that every cactus in  $\mathfrak{R}_Q^{\min}$  has branching number at most 1 and use Theorem 16. Here, we sketch a proof of the latter.

We only consider  $q_{0,r}$ . Suppose  $C \in \mathfrak{R}_Q^{\min}$ . Note first that if, in a segment  $s$  of  $C$ , some  $T$ -node  $x$  has been pruned (that is, its label  $T$  removed), then all the  $T$ -nodes  $y$  with  $x < y$  in  $s$  (if any) can also be pruned, contrary to the minimality of  $C$ . (To see this, consider any model  $\mathcal{I}$  of  $\mathcal{T}$  and  $C$ , in which all the  $A$ -nodes in the submodel  $\mathcal{I}_x$  generated by  $x$  are in  $T^{\mathcal{I}}$ . Let  $h$  be a homomorphism from  $q_{0,r}$  to  $\mathcal{I}$ . Let  $C'$  result from  $C$  by pruning all the  $T$ -nodes  $y$  with  $x < y$  and let  $\mathcal{I}'$  be the restriction of  $\mathcal{I}$  to  $C'$ . Then  $h$  is also a homomorphism from  $q_{0,r}$  to  $\mathcal{I}'$ .)

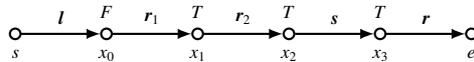
Consider any branch  $s_0, \dots, s_n$  of segments in  $C$  with the maximal number of nodes between the root of  $s_0$  and the leaf of  $s_n$ . By this choice,  $s_{n-1}$  cannot have any  $A$ -nodes after (w.r.t.  $<$ ) the root of  $s_n$ . We claim that  $s_n$  contains all of its  $T$ -nodes. Indeed, suppose at least one of them has been pruned. Then it is readily seen that the root of  $s_n$  in  $s_{n-1}$  can also be pruned (consider the models  $\mathcal{I}$  of  $\mathcal{T}$  and  $C$  in which this root is in  $F^{\mathcal{I}}$ ), contrary to the minimality of  $C$ . Now, let  $\mathcal{A}$  be the subgraph of  $C$  comprising the nodes on this branch (with all of their labels). It is not hard to check that  $\mathcal{T}, \mathcal{A} \models q_{0,r}$ . It follows that all the  $T$ -nodes that are not on the branch should have been pruned.  $\square$

We next show that the OMQs with 1-CQs not covered by Theorem 19 are all P-hard.

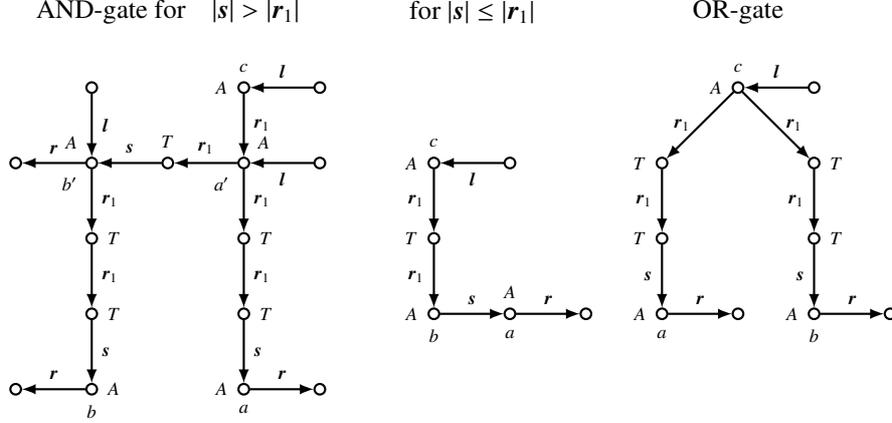
**Theorem 20.** *Let  $q = q_{l,r}$  be a twinless path 1-CQ such that one of the following conditions holds: (i)  $l, r \geq 1$ , or (ii)  $l = 0$  and  $q_{0,r}$  is not right-periodic, or (iii)  $r = 0$  and  $q_{l,0}$  is not left-periodic. Then  $(\text{cov}_A, q_{l,r})$  is P-hard.*

*Proof.* Each of the cases (i)–(iii) is proved by an FO-reduction of the monotone circuit evaluation problem, which is known to be P-complete. We remind the reader that a *monotone Boolean circuit* is a directed acyclic graph  $C$  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  $\alpha$  of  $F$  and  $T$  to the input gates of  $C$ , we compute the value of each gate in  $C$  under  $\alpha$  as usual in Boolean logic. The *output*  $C(\alpha)$  of  $C$  on  $\alpha$  is the truth-value of the output gate.

(ii) Let  $l = [s, x_0]$ , let  $n > 1$  be minimal with  $r_n \not\rightarrow r_1$ ,  $s = r_n$ , and let  $r = (x_n, e]$ . 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,  $q_{0,r}$  may look as follows:



We distinguish between two cases:  $|s| > |r_1|$  and  $|s| \leq |r_1|$ . Depending on the case, we use the following two gadgets for AND-gates; the gadget for OR-gates is the same in both cases:



Given a monotone circuit  $C$  and an assignment  $\alpha$ , we construct an ABox  $\mathcal{A}_{C,\alpha}$  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  $\text{cov}_A, \mathcal{A}_{C,\alpha} \models q_{0r}$  iff  $C(\alpha) = T$ .

( $\Leftarrow$ ) is proved by induction on the number of non-input gates in  $C$ . The basis is obvious. For the induction step, suppose the output gate  $g$  in  $C$  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  $\text{cov}_A$  and  $\mathcal{A}_{C,\alpha}$ . 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  $q_{0r} \rightarrow \mathcal{I}$  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  $C^-$  of  $C$  whose output gate is  $g_i$ . Then  $\mathcal{A}_{C^-, \alpha}$  is the sub-ABox of  $\mathcal{A}_{C,\alpha}$  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  $\mathcal{A}_{C^-, \alpha}$  (and so  $c \in F^{\mathcal{I}^-}$ ), then by IH there is a  $q_{0r} \rightarrow \mathcal{I}^-$  homomorphism, and so  $\mathcal{I} \models q_{0r}$  as well. The case when the output gate  $g$  in  $C$  is an OR-gate is similar.

( $\Rightarrow$ ) Suppose  $C(\alpha) = F$ . To show  $\text{cov}_A, \mathcal{A}_{C,\alpha} \not\models q_{0r}$ , we define a model  $\mathcal{I}$  of  $\text{cov}_A$  and  $\mathcal{A}_{C,\alpha}$  inductively by labelling the  $A$ -nodes in the gadget for each non-input gate  $g$  of  $C$  by  $F^{\mathcal{I}}$  or  $T^{\mathcal{I}}$  as follows: node  $c$  is labelled by the truth-value of  $g$  under  $\alpha$ , while node  $i$  (and node  $i'$  if applicable), for  $i = a, b$ , is labelled by the truth-value of  $g_i$  under  $\alpha$ , where  $g_a$  and  $g_b$  are the inputs of  $g$ . Suppose, on the contrary, that there is a homomorphism  $h: q_{0r} \rightarrow \mathcal{I}$  and consider possible locations of  $h(x_0) \in F^{\mathcal{I}}$ . Suppose first that  $|s| > |r_1|$  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  $|s| > |r_1|$ , 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  $b \in F^{\mathcal{I}}$ .

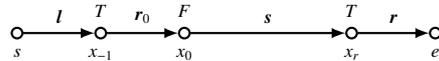
Case  $a, a', c \in F^{\mathcal{I}}, b, b' \in T^{\mathcal{I}}$ . If  $h(x_0) = a'$ , then  $h(x_1)$  is the central  $T$ -node. But then, since  $|s| > |r_1|$ , the node  $h(x_2)$  must be strictly between  $b'$  and the central  $T$ -node, which is impossible because there are no  $T$ -nodes there.

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

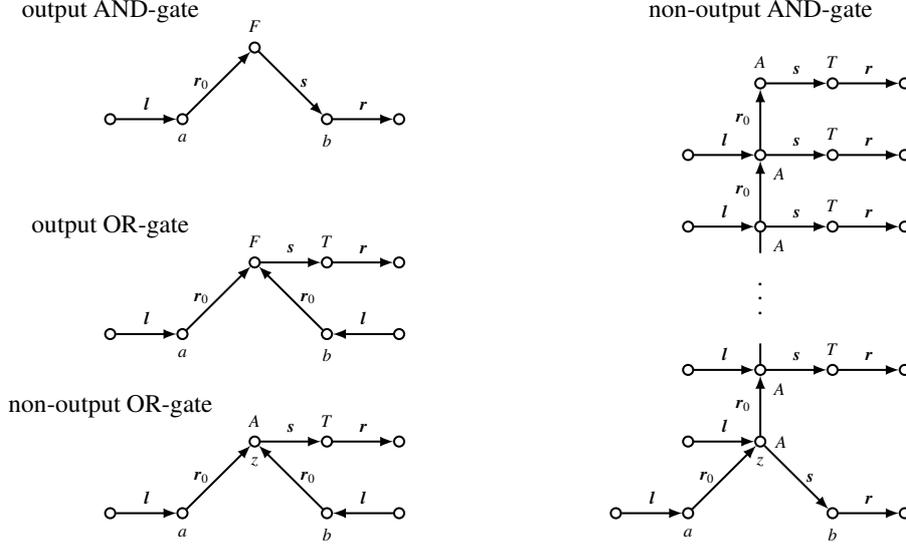
Suppose next that  $|s| \leq |r_1|$  and  $h(x_0) = c$  is in some AND-gadget. Then  $h(x_2) = b$ , provided that  $b \in T^{\mathcal{I}}$  (otherwise such  $h$  is impossible), which means that  $a \in F^{\mathcal{I}}$ , and so  $h(x_3)$  is located in some other gadget. However, this is impossible because of the following. In every gadget, the 'edges' leaving node  $c$  are labelled by  $r_1$ . So if  $|s| < |r_1|$  then  $h(x_3)$  must be strictly between the  $c$  node of the gadget and the end-node of an  $r_1$ -edge, but there are no  $T$ -nodes there. If  $|s| = |r_1|$  then  $s \rightsquigarrow r_1$ , contrary to  $s \not\rightsquigarrow r_1$ .

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

(iii) In our reduction, we require four intervals of  $q_{lr}$ :  $l = [s, x_{-1}]$ ,  $r_0 = (x_{-1}, x_0)$ ,  $s = (x_0, x_r)$ ,  $r = (x_r, e]$ . Note that  $r_0$  has no  $T$ -nodes.



We use the following gadgets for the gates, where the number of  $A$ -nodes in the gadget for a non-output AND-gate exceeds  $|\mathbf{q}_{lr}|$ :



Given a monotone circuit  $\mathcal{C}$  and an assignment  $\alpha$ , we construct an ABox  $\mathcal{A}_{\mathcal{C},\alpha}$  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 the topmost  $A$ -node 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  $i$  in the gadget for  $g$  with  $\alpha(g_i)$ . We claim that  $\text{cov}_A, \mathcal{A}_{\mathcal{C},\alpha} \models \mathbf{q}_{0r}$  iff  $\mathcal{C}(\alpha) = T$ .

( $\Leftarrow$ ) is proved by induction on the number of non-input gates in  $\mathcal{C}$ . The basis (when  $\mathcal{C}$  has one non-input gate) is obvious. For the induction step, suppose the output gate  $g$  in  $\mathcal{C}$  is an OR-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  $\text{cov}_A$  and  $\mathcal{A}_{\mathcal{C},\alpha}$ . If at least one of  $a$  or  $b$  in the gadget for  $g$  is in  $T^{\mathcal{I}}$ , then clearly  $\mathcal{I} \models \mathbf{q}_{lr}$ . It remains to consider the case when  $a$  and  $b$  are both in  $F^{\mathcal{I}}$ . Let  $i$  be such that  $g_i$  is a non-input gate. There are two cases. (i) If node  $z$  in the gadget for  $g_i$  is in  $F^{\mathcal{I}}$ , consider the subcircuit  $\mathcal{C}^-$  of  $\mathcal{C}$  whose output gate is  $g_i$ . Then  $\mathcal{A}_{\mathcal{C}^-,\alpha}$  is the sub-ABox of  $\mathcal{A}_{\mathcal{C},\alpha}$  with  $z$  as its topmost node, and  $A(z)$  replaced by  $F(z)$ . Now, if  $\mathcal{I}^-$  is the restriction of  $\mathcal{I}$  to  $\mathcal{A}_{\mathcal{C}^-,\alpha}$  (and so  $z \in F^{\mathcal{I}^-}$ ), then by IH there is a  $\mathbf{q}_{lr} \rightarrow \mathcal{I}^-$  homomorphism, and so  $\mathcal{I} \models \mathbf{q}_{lr}$  as well. (ii) If  $z \in T^{\mathcal{I}}$  then  $g_i$  is an AND-gate and, as the topmost  $A$ -node in the gadget for  $g_i$  is in  $F^{\mathcal{I}}$ , there is an  $A$ -node in the gadget for  $g_i$  that is in  $T^{\mathcal{I}}$  while the next  $A$ -node above it is in  $F^{\mathcal{I}}$ . So we have a  $\mathbf{q}_{lr} \rightarrow \mathcal{I}$  homomorphism.

The case when the output gate of  $\mathcal{C}$  is an AND-gate is similar.

( $\Rightarrow$ ) Suppose  $\mathcal{C}(\alpha) = F$ . To show  $\text{cov}_A, \mathcal{A}_{\mathcal{C},\alpha} \not\models \mathbf{q}_{lr}$ , we define a model  $\mathcal{I}$  of  $\text{cov}_A$  and  $\mathcal{A}_{\mathcal{C},\alpha}$  by putting the  $A$ -nodes of the gadget for any gate  $g$  in  $\mathcal{C}$  to  $F^{\mathcal{I}}$  (or  $T^{\mathcal{I}}$ ) if the truth-value of  $g$  under  $\alpha$  is  $F$  (or, respectively,  $T$ ). Suppose, on the contrary, that there is a homomorphism  $h: \mathbf{q}_{lr} \rightarrow \mathcal{I}$ . We track the possible locations of  $h(x_0) \in F^{\mathcal{I}}$ :

- If the output gate is an AND-gate, then  $h(x_0)$  cannot be its  $F$ -node, because then  $h(x_{-1}) = a$  and  $h(x_r) = b$ , and so at least one of them would be in  $F^{\mathcal{I}}$ , which is a contradiction.
- If the output gate is an OR-gate, then  $h(x_0)$  cannot be its  $F$ -node, because then either  $h(x_{-1}) = a$  or  $h(x_{-1}) = b$ , and so  $h(x_{-1})$  would be in  $F^{\mathcal{I}}$ , a contradiction.
- So suppose  $h(x_0)$  is an  $A$ -node in a gadget for a non-input and non-output gate  $g$ . If  $g$  is an OR-gate, then either  $h(x_{-1}) = a$  or  $h(x_{-1}) = b$  in the gadget for  $g$ , and so  $h(x_{-1})$  would be in  $F^{\mathcal{I}}$ , a contradiction. So suppose  $g$  is an AND-gate, and consider the gadget for  $g$ . Then  $h(x_0)$  cannot be any  $A$ -node located above  $z$ , because otherwise  $h(x_{-1})$  would be the previous  $A$ -node, and so in  $F^{\mathcal{I}}$ , a contradiction. Finally, if  $h(x_0) = z$  then, as the vertical line comprised of the  $r_0$  is longer than  $\mathbf{q}_{lr}$  and contains no  $T$ -nodes,  $h(x_1) \in T^{\mathcal{I}}$  must also be in the gadget for  $g$ , and it must be in one of the horizontal  $s$ . But this is impossible because  $r_0$  is non-empty, and so the distance between  $z = h(x_0)$  and  $h(x_1)$  would be greater than  $\delta(x_0, x_1)$ .

Thus, we cannot have a homomorphism  $h: \mathbf{q}_{lr} \rightarrow \mathcal{I}$ . □

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

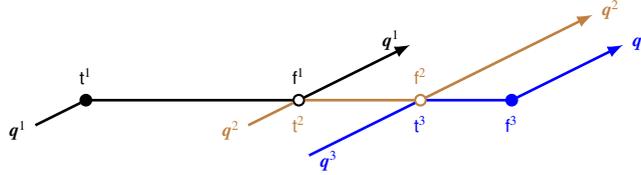
**Theorem 21.** *If  $q$  is a twinless path 2-CQ, then  $(\text{cov}_A, q)$  is coNP-hard.*

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

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 an ABox  $\mathcal{A}$ , we call a map  $h: q \rightarrow \mathcal{I}$  a *subhomomorphism* if the following conditions hold:

- $h(x) \in T^{\mathcal{I}}$ , for every  $T$ -node  $x$  in  $q$ , and  $h(x) \in F^{\mathcal{I}}$ , for every  $F$ -node  $x$  in  $q$ ;
- for any nodes  $x, y$  in  $q$ , if  $R(x, y)$  is in  $q$  for some  $R$ , then  $S(h(x), h(y))$  is in  $\mathcal{A}$  for some  $S$ .

Second, we define some ABoxes that are ‘built up’ from copies of  $q$  in a particular way. If  $x \leq y$ , we let  $\delta(x, y)$  denote the distance between  $x$  and  $y$  in  $q$ , that is, the number of edges in the path from  $x$  to  $y$ , and set  $|q| = \delta(s, e)$ . Given any path CQ  $q'$ , we write  $<_{q'}$  and  $\leq_{q'}$  for the ordering of nodes in  $q'$ , and  $\delta_{q'}$  for the distance in  $q'$ . We omit the subscripts when  $q' = q$ . Now let  $q^1, \dots, q^n$ ,  $n \geq 2$ , be disjoint copies of  $q$ . For any  $j$  and node  $x$  in  $q$ , we let  $x^j$  denote the copy of  $x$  in  $q^j$ , and let  $\iota^j: q^j \rightarrow q$  be the identity map. We assume that  $q$  contains a  $T$ -node  $<$ -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  $q^j$  such that  $t^j <_{q^j} f^j$ ; 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  $q$ )*. An example of a 3-chain is given in the picture below, where glue-contacts are depicted by  $\circ$ , while contacts that are not glue-contacts by  $\bullet$ .



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 21.2, 21.3 and 21.5:

**Lemma 21.1.** *Suppose  $\mathcal{H}$  is an n-chain, for some  $n \geq 2$  and a twinless path CQ  $q$ .*

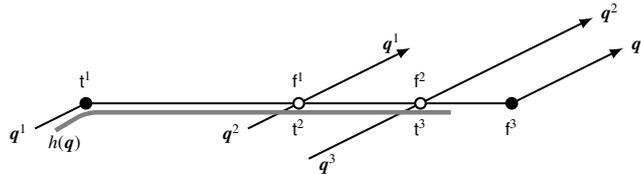
(i) *If  $h: q \rightarrow \text{ind}(\mathcal{H})$  is a function with  $s^1 <_{q^1} h(s) <_{q^1} f^1$ , and  $\mathcal{I}$  a model of  $\text{cov}_A$  and  $\mathcal{H}$  whose glue-contacts are all in  $F^{\mathcal{I}}$ , then  $h$  is not a  $q \rightarrow \mathcal{I}$  subhomomorphism.*

(ii) *If  $h: q \rightarrow \text{ind}(\mathcal{H})$  is a function with  $t^n <_{q^n} h(e) <_{q^n} e^n$ , and  $\mathcal{I}$  a model of  $\text{cov}_A$  and  $\mathcal{H}$  whose glue-contacts are all in  $T^{\mathcal{I}}$ , then  $h$  is not a  $q \rightarrow \mathcal{I}$  subhomomorphism.*

*Proof.* (i) Suppose on the contrary that  $h: q \rightarrow \mathcal{I}$  is a subhomomorphism. We consider the image  $h(q)$  of  $q$  in  $\mathcal{I}$  as a path CQ, so  $\leq_{h(q)}$  and  $\delta_{h(q)}$  are well-defined. Observe that, by our assumption on  $h$ ,

$$\text{for every } \ell, 1 \leq \ell < n, \text{ if } x \text{ is in } q^\ell \cap h(q) \text{ then } x \leq_{q^\ell} f^\ell; \quad (17)$$

$$\text{for every } \ell, 1 < \ell \leq n, \text{ if } x \text{ is in } q^\ell \cap h(q) \text{ then } t^\ell \leq_{q^\ell} x. \quad (18)$$



We define a function  $g_{\leftarrow} : \text{ind}(\mathcal{H}) \rightarrow \text{ind}(\mathcal{H})$  by taking  $g_{\leftarrow}(x) = h(t^\ell(x))$  whenever  $x$  is a node in  $\mathbf{q}^\ell$ , where we consider each glue-contact  $\mathbf{c} = \mathbf{f}^i = \mathbf{t}^{i+1}$ , for  $1 \leq i < n$ , as a node in  $\mathbf{q}^{i+1}$ , that is,  $g_{\leftarrow}(\mathbf{c}) = g_{\leftarrow}(\mathbf{t}^{i+1}) = h(\mathbf{t}^{i+1}(\mathbf{t}^{i+1}))$ . Throughout, we use the following obvious ‘shift’ property of  $g_{\leftarrow}$ : for every  $\ell$ ,  $1 \leq \ell \leq n$ ,

$$\begin{aligned} \text{if } y, z \text{ are both in the same copy } \mathbf{q}^\ell, y, z \neq \mathbf{t}^\ell \text{ whenever } \ell < n, \text{ and } y \preceq_{\mathbf{q}^\ell} z, \\ \text{then } g_{\leftarrow}(y) \preceq_{h(\mathbf{q})} g_{\leftarrow}(z) \text{ and } \delta_{\mathbf{q}^\ell}(y, z) = \delta_{h(\mathbf{q})}(g_{\leftarrow}(y), g_{\leftarrow}(z)). \end{aligned} \quad (19)$$

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  $g_{\leftarrow}^N(x) = x$ . We will ‘shift this fixpoint-cycle to the left.’ More precisely, we claim that

$$\text{there is a glue-contact } \mathbf{c} \text{ with } g_{\leftarrow}^N(\mathbf{c}) = \mathbf{c}. \quad (20)$$

Indeed, let  $y_0 = x, y_1 = g_{\leftarrow}(x), y_2 = g_{\leftarrow}^2(x), \dots, y_{N-1} = g_{\leftarrow}^{N-1}(x)$ . Then  $g_{\leftarrow}^N(y_j) = y_j$  for every  $j < N$ , and so if one of the  $y_j$  is a glue-contact, we are done with (20). So suppose otherwise. We cannot have that every  $y_j$  is in  $\mathbf{q}^1$ , as otherwise, by (19) and our assumption on  $h$ , for every  $j \leq N$ ,

$$\delta_{\mathbf{q}^1}(s^1, y_j) = \delta_{h(\mathbf{q})}(g_{\leftarrow}(s^1), g_{\leftarrow}(y_j)) = \delta_{h(\mathbf{q})}(h(s), y_{j+1}) = \delta_{\mathbf{q}^1}(h(s), y_{j+1}) < \delta_{\mathbf{q}^1}(s^1, y_{j+1}) \quad (21)$$

(here  $+$  is modulo  $N$ ). Therefore,

$$\text{there exists } j < N \text{ such that } y_j \text{ is in } \mathbf{q}^{\ell_j}, \text{ for some } \ell_j > 1, \quad (22)$$

and so, by (18), there is a glue-contact  $\mathbf{t}^{\ell_j}$  with  $\mathbf{t}^{\ell_j} \prec_{\mathbf{q}^{\ell_j}} y_j$ . (When  $\ell_j = 1$ , such a glue-contact does not exist.) For  $j < N$  with  $\ell_j > 1$ , we set  $d_j = \delta_{\mathbf{q}^{\ell_j}}(\mathbf{t}^{\ell_j}, y_j)$ . Let  $K < N$  be such that

$$d_K = \min\{d_j \mid j < N \text{ and } \ell_j > 1\}$$

(which is well-defined by (22)), and set  $\mathbf{c} = \mathbf{t}^{\ell_K} = \mathbf{t}^{\ell_K}$ . By (17), we have  $y_j \prec_{\mathbf{q}^{\ell_j}} \mathbf{t}^{\ell_j}$  whenever  $1 < \ell_j < n$ . Thus, by the definition of  $K$  and (19), for every  $j \leq N$ ,  $g_{\leftarrow}^j(\mathbf{c})$  belongs to the same copy  $\mathbf{q}^{\ell_{K+j}}$  as  $y_{K+j}$ ,  $g_{\leftarrow}^j(\mathbf{c}) \preceq_{\mathbf{q}^{\ell_{K+j}}} y_{K+j}$ , and

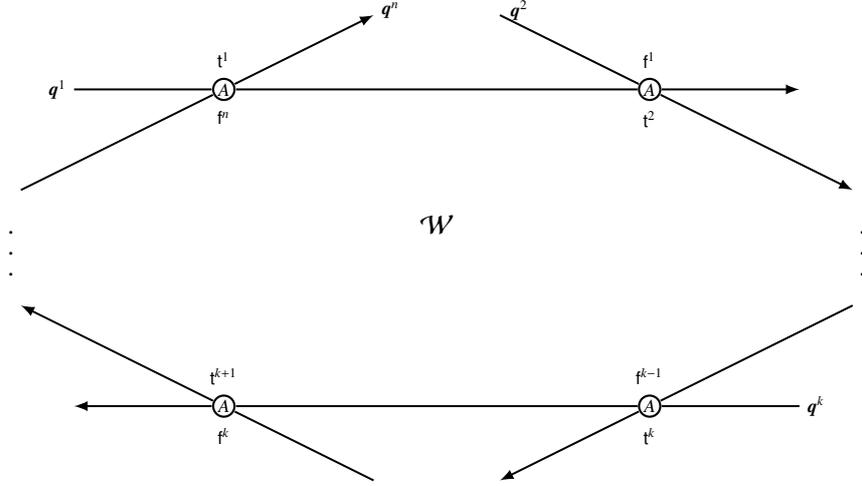
$$d_K = \delta_{\mathbf{q}^{\ell_{K+j}}}(g_{\leftarrow}^j(\mathbf{c}), y_{K+j}).$$

It follows, in particular, that  $g_{\leftarrow}^N(\mathbf{c})$  belongs to the same copy  $\mathbf{q}^{\ell_K}$  as  $y_K$ , and  $\delta_{\mathbf{q}^{\ell_K}}(g_{\leftarrow}^N(\mathbf{c}), y_K) = \delta_{\mathbf{q}^{\ell_K}}(\mathbf{c}, y_K)$ . Therefore,  $g_{\leftarrow}^N(\mathbf{c}) = \mathbf{c}$ , as required in (20).

It remains to show that (20) leads to a contradiction. Indeed,  $\mathbf{c} \in F^{\mathcal{I}}$  by our assumption, and so  $\mathbf{c}$  cannot be in  $T^{\mathcal{I}}$  by the minimality of  $\mathcal{I}$ . On the other hand, we show by induction on  $j \geq 1$  that  $g_{\leftarrow}^j(\mathbf{c}) \in T^{\mathcal{I}}$ , and so  $\mathbf{c} = g_{\leftarrow}^N(\mathbf{c}) \in T^{\mathcal{I}}$ . If  $j = 1$  then  $g_{\leftarrow}(\mathbf{c}) = h(t^\ell(\mathbf{t}^\ell))$  for some  $\ell$ , and so  $g_{\leftarrow}(\mathbf{c}) \in T^{\mathcal{I}}$  as  $t^\ell(\mathbf{t}^\ell)$  is a  $T$ -node in  $\mathbf{q}$  and  $h$  is a subhomomorphism. If  $j > 1$  then  $g_{\leftarrow}^{j-1}(\mathbf{c}) \in T^{\mathcal{I}}$  by the IH, and so  $g_{\leftarrow}^{j-1}(\mathbf{c})$  is not a glue-contact and  $t^\ell(g_{\leftarrow}^{j-1}(\mathbf{c}))$  must be a  $T$ -node in  $\mathbf{q}$  for some  $\ell$ . Thus,  $g_{\leftarrow}^j(\mathbf{c}) = h(t^\ell(g_{\leftarrow}^{j-1}(\mathbf{c})))$  is in  $T^{\mathcal{I}}$ .

(ii) The proof is similar. Now we define a function  $g_{\rightarrow} : \text{ind}(\mathcal{H}) \rightarrow \text{ind}(\mathcal{H})$  by taking again  $g_{\rightarrow}(x) = h(t^\ell(x))$  whenever  $x$  is a node in  $\mathbf{q}^\ell$ , but now we consider each contact  $\mathbf{c} = \mathbf{f}^i = \mathbf{t}^{i+1}$  as a node in  $\mathbf{q}^i$ , that is,  $g_{\rightarrow}(\mathbf{c}) = g_{\rightarrow}(\mathbf{f}^i) = h(t^i(\mathbf{f}^i))$ . Then, in the proof of (20) for  $g_{\rightarrow}$ , the assumption on  $h$  and the ‘shift’ property for  $g_{\rightarrow}$  (analogous to (19)) imply that there exists  $j < N$  such that  $y_j$  is in  $\mathbf{q}^{\ell_j}$ , for some  $\ell_j < n$ , and so we can define the contacts  $\mathbf{f}^{\ell_j}$  such that  $y_j \preceq_{\mathbf{q}^{\ell_j}} \mathbf{f}^{\ell_j}$  for each  $j$  with  $\ell_j < n$  (that is, we ‘shift the fixpoint-cycle to the right’).  $\square$

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



**Lemma 21.2.** *Suppose  $\mathcal{W}$  is an  $n$ -cogwheel for some  $n \geq |q|$ . For any model  $\mathcal{I}$  of  $\text{cov}_A$  and  $\mathcal{W}$ , we have  $\mathcal{I} \not\models q$  iff the contacts in  $\mathcal{I}$  are either all in  $T^{\mathcal{I}}$  or all in  $F^{\mathcal{I}}$ .*

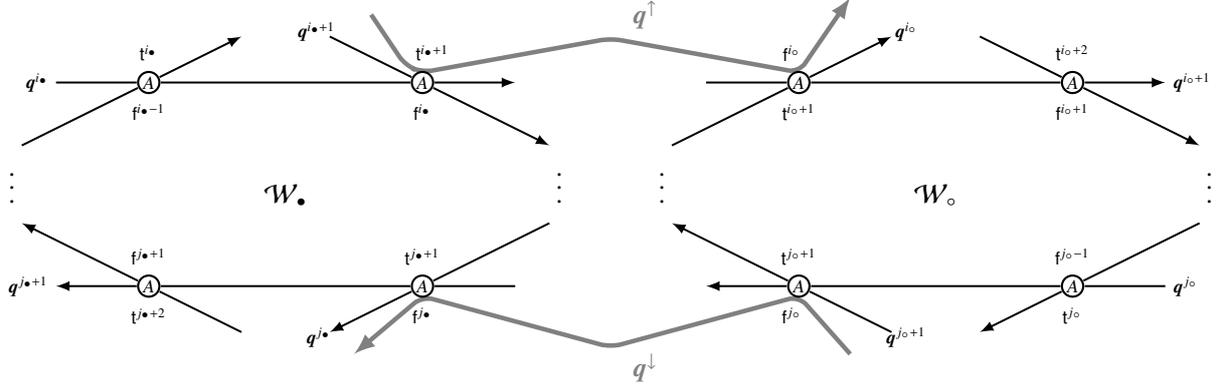
*Proof.* ( $\Rightarrow$ ) Suppose the contact  $f^{i-1} = t^i$  is in  $T^{\mathcal{I}}$ . Since  $\mathcal{I} \not\models q$ , the ‘clockwise next’ contact  $f^i = t^{i+1}$  is also in  $T^{\mathcal{I}}$ . It follows by induction that all of the contacts in  $\mathcal{W}$  are in  $T^{\mathcal{I}}$ . If the contact  $f^{i-1} = t^i$  is in  $F^{\mathcal{I}}$ , then the ‘anti-clockwise next’ contact  $f^{i-2} = t^{i-1}$  is also in  $F^{\mathcal{I}}$ , from which it follows by induction that all of the contacts are in  $F^{\mathcal{I}}$ .

( $\Leftarrow$ ) Suppose otherwise. Suppose first that there exists a model  $\mathcal{I}$  of  $\text{cov}_A$  and  $\mathcal{W}$  such that all contacts in  $\mathcal{I}$  are in  $F^{\mathcal{I}}$  and  $\mathcal{I} \models q$ , and so there is a homomorphism  $h: q \rightarrow \mathcal{I}$ . As  $n \geq |q|$ , we may consider the image  $h(q)$  of  $q$  in  $\mathcal{I}$  as a path  $CQ$ , so  $\leq_{h(q)}$  and  $\delta_{h(q)}$  are well-defined. Further,  $h(q)$  must intersect with at least two copies of  $q$  in  $\mathcal{W}$  (otherwise  $h(t^j(t^j)) = t^j$  were in  $F^{\mathcal{I}}$ , for some  $T$ -node  $t^j(t^j)$  in  $q$ , contrary to the minimality of  $\mathcal{I}$ ). As  $t^{j+1}(t^{j+1}) < t^j(t^j)$  for every  $j$ , there is not enough room for  $h(q)$  intersecting only with cogs. So without loss of generality we can assume that the intersection of  $h(q)$  with each of the copies  $q^1, \dots, q^k$ , for some  $k, 2 \leq k < n$ , consists of not just contacts. (In fact, the intersections with  $q^2, \dots, q^{k-1}$  consist of non-cog-nodes only, while with  $q^1$  and  $q^k$  they might or might not contain cog-nodes.) In particular,  $h(s) <_{q^1} f^1$ . We also have  $s^1 <_{q^1} h(s)$  (otherwise  $h(t^1(t^1)) = t^1$  were in  $F^{\mathcal{I}}$ ). Now, consider the sub-ABox  $\mathcal{H}$  of  $\mathcal{W}$  consisting of the copies  $q^1, \dots, q^k$ . Then  $\mathcal{H}$  is a  $k$ -chain, and  $h$  is a  $q \rightarrow \text{ind}(\mathcal{H})$  function satisfying the assumption of Lemma 21.1 (i). Let  $\mathcal{I}^-$  be the restriction of  $\mathcal{I}$  to  $\mathcal{H}$ . Then, by Lemma 21.1 (i),  $h$  is not a  $q \rightarrow \mathcal{I}^-$  subhomomorphism, and so it is not a  $q \rightarrow \mathcal{I}$  homomorphism either, which is a contradiction.

The case of  $\mathcal{I}$  with contacts in  $T^{\mathcal{I}}$  is similar, where we use Lemma 21.1 (ii).  $\square$

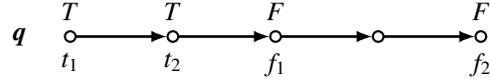
Next, for each variable  $p$  occurring in the 3CNF  $\psi$ , we take a fresh pair of cogwheels and make sure that they always encode the opposite truth-values of the literals  $p$  and  $\neg p$ . To achieve this, we connect the two cogwheels in each pair with further two copies of  $q$  in a special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

Let  $\mathcal{W}_\bullet$  and  $\mathcal{W}_\circ$  be two disjoint  $n$ -cogwheels for some  $n > 4|q| + 2$ , and let  $q^\uparrow, q^\downarrow$  be two more fresh and disjoint copies of  $q$ . For  $j = \uparrow, \downarrow$  and a node  $x$  in  $q$ , we denote by  $x^j$  the copy of  $x$  in  $q^j$ . We pick two contacts  $c_\bullet^\uparrow = f^\bullet = t^{\bullet+1}$  and  $c_\bullet^\downarrow = f^{\bullet} = t^{\bullet+1}$  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  $> 2|q|$ . Similarly, we pick two contacts  $c_\circ^\uparrow = f^\circ = t^{\circ+1}$  and  $c_\circ^\downarrow = f^{\circ} = t^{\circ+1}$  in  $\mathcal{W}_\circ$  such that the contact-distance between them in  $\mathcal{W}_\circ$  is  $> 2|q|$ . Then we glue together the contact  $c_\bullet^\uparrow$  in  $\mathcal{W}_\bullet$  with  $f_1^\uparrow$ , and also the contact  $c_\circ^\uparrow$  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  $c_\bullet^\downarrow$  in  $\mathcal{W}_\bullet$  with  $t_1^\downarrow$ , and also the contact  $c_\circ^\downarrow$  in  $\mathcal{W}_\circ$  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  $c_\bullet^\uparrow = f^\bullet = t^{\bullet+1} = f_1^\uparrow$  and  $c_\circ^\uparrow = f^\circ = t^{\circ+1} = f_2^\uparrow$   $F$ -connections and the contacts  $c_\bullet^\downarrow = f^\bullet = t^{\bullet+1} = t_1^\downarrow$  and  $c_\circ^\downarrow = f^\circ = t^{\circ+1} = t_2^\downarrow$   $T$ -connections in  $\mathcal{B}$ .

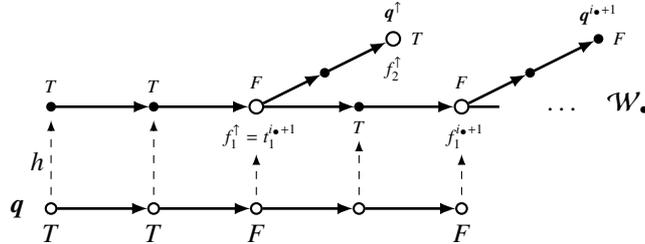


Throughout, for any  $k$ , we let  $t_k$  ( $f_k$ ) denote the  $k$ th  $T$ -node ( $F$ -node) in  $q$ . In particular,  $t_{last-1}$  ( $f_{last-1}$ ) denotes the last but one  $T$ -node ( $F$ -node) in  $q$ , and  $t_{last}$  ( $f_{last}$ ) the last  $T$ -node ( $F$ -node). We assume that  $t_1 < f_1$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

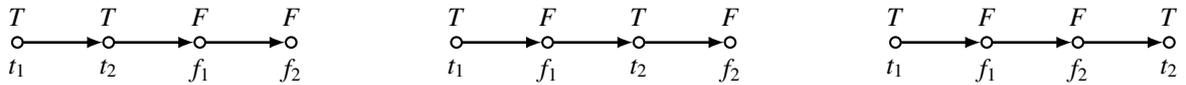
We want to achieve that, for any model  $\mathcal{I}$  of  $\text{cov}_A$  and  $\mathcal{B}$ , we have  $\mathcal{I} \not\models q$  iff the contacts in  $\mathcal{W}_\bullet$  are all in  $T^{\mathcal{I}}$  while the contacts in  $\mathcal{W}_\circ$  are all in  $F^{\mathcal{I}}$ , or the other way round. Using Lemma 21.2 and the fact that the  $F$ -connections are  $F$ -nodes in  $q^\uparrow$  while the  $T$ -connections are  $T$ -nodes in  $q^\downarrow$ , 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 ‘ $\pm|q|$ -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  $h(q)$  of a potential homomorphism  $h: q \rightarrow \mathcal{I}$  are excluded. Our choices depend on the particular 2-CQ. For example, consider the 2-CQ



If we choose  $t^{i\bullet+1} = t_1^{i\bullet+1}$  and  $f^{i\bullet+1} = f_1^{i\bullet+1}$ , 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 do have the following  $h: q \rightarrow \mathcal{I}$  homomorphism:



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



For each particular 2-CQ  $q$ , 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 21.3 below, we describe a system of choices that works for *every* 2-CQ. The different potential locations of a homomorphic image place 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, and so 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  $q$  to the initial cog of a copy  $q^j$ , then we must have that  $x \leq v^j(h(x))$ , otherwise there is not enough room for the whole  $h(q)$  in the cog.

**Lemma 21.3.** Suppose  $\mathcal{B}$  is an  $n$ -bike such that the following hold for its  $F$ -connections:

- if  $t_2 < f_1$  and  $\delta(f_1, f_2) \geq \delta(t_1, f_1)$ , then  $\mathfrak{t}^{i_1+k} = t_2^{i_1+k}$  and  $\mathfrak{f}^{i_1+k} = f_2^{i_1+k}$ , for all  $k, 1 \leq k \leq |q|$ ; otherwise,  $\mathfrak{t}^{i_1+k} = t_1^{i_1+k}$  and  $\mathfrak{f}^{i_1+k} = f_1^{i_1+k}$ , for all  $k, 1 \leq k \leq |q|$ ;
- $\mathfrak{f}^{i_1} = f_2^{i_1}$  and  $\mathfrak{f}^{i_1-k} = f_1^{i_1-k}$ , for all  $k, 0 < k \leq |q|$ ;
- $\mathfrak{t}^{i_1-k} = t_1^{i_1-k}$  and  $\mathfrak{t}^{i_1-k} = t_1^{i_1-k}$ , for all  $k, k \leq |q|$ ;
- $\mathfrak{t}^{i_1+1} = t_1^{i_1+1}$  and  $\mathfrak{f}^{i_1+1} = f_1^{i_1+1}$ ;
- $\mathfrak{t}^{i_1} = f_2^{i_1}$  and  $\mathfrak{f}^{i_1-k} = f_1^{i_1-k}$ , for all  $k, 0 < k \leq |q|$ ;

and the following hold for its  $T$ -connections:

- $\mathfrak{t}^{j_1+1} = t_1^{j_1+1}$ ,  $\mathfrak{t}^{j_1-k} = t_1^{j_1-k}$  and  $\mathfrak{f}^{j_1-k} = f_1^{j_1-k}$ , for all  $k \leq |q|$ ;
- if  $t_2 < f_1$  and  $\delta(f_1, f_2) \geq \delta(t_1, f_1)$ , then  $\mathfrak{t}^{j_1+k} = t_2^{j_1+k}$  and  $\mathfrak{f}^{j_1+k} = f_2^{j_1+k}$ , for all  $k, 1 \leq k \leq |q|$ ; otherwise,  $\mathfrak{t}^{j_1+k} = t_1^{j_1+k}$  and  $\mathfrak{f}^{j_1+k} = f_1^{j_1+k}$ , for all  $k, 1 \leq k \leq |q|$ .

Then, for any model  $\mathcal{I}$  of  $\text{cov}_A$  and  $\mathcal{B}$ , we have  $\mathcal{I} \not\cong q$  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  $> 2|q|$ .

*Proof.* The implication  $(\Rightarrow)$  clearly holds for any  $n$ -bike  $\mathcal{B}$  by the  $(\Rightarrow)$  direction of Lemma 21.2.

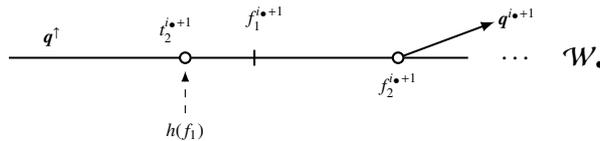
To show  $(\Leftarrow)$ , suppose  $\mathcal{B}$  is as above, and  $\mathcal{I}$  is a model of  $\text{cov}_A$  and  $\mathcal{B}$  such that all contacts in  $\mathcal{W}_\bullet$  are in  $F^{\mathcal{I}}$  and all contacts in  $\mathcal{W}_\circ$  are in  $T^{\mathcal{I}}$  or the other way round. The proof is via excluding all possible locations in  $\mathcal{B}$  for the image  $h(q)$  of a potential subhomomorphism  $h: q \rightarrow \mathcal{I}$ . As  $n > 4|q| > |q|$ , we may consider the image  $h(q)$  of  $q$  in  $\mathcal{I}$  as a path CQ. By Lemma 21.2,  $h(q)$  must intersect with at least one of  $q^\uparrow$  and  $q^\downarrow$ . As the  $F$ -connections are of distance  $> 2|q|$  from the  $T$ -connections,  $h(q)$  cannot intersect with both  $q^\uparrow$  and  $q^\downarrow$  at the same time. There are several cases, and we show that all of them lead to a contradiction. First, we deal with the case when  $h(q) \cap q^\uparrow \neq \emptyset$ . We track the location of  $h(f_1)$ . We have the following four cases (1) $^\uparrow$ –(4) $^\uparrow$ :

(1) $^\uparrow$   $h(f_1) = \mathfrak{c}_\bullet^\uparrow = f_1^\uparrow$  and  $h(f_2)$  is in  $q^\uparrow$ . Then  $h(f_2) = \mathfrak{c}_\circ^\uparrow = f_2^\uparrow$ , and so either  $h(f_1)$  or  $h(f_2)$  is in  $T^{\mathcal{I}}$ .

(2) $^\uparrow$   $h(f_1)$  is in  $q^{i_1+k}$  in  $\mathcal{W}_\bullet$  for some  $1 \leq k \leq |q|$ , and  $h(f_2)$  is also in  $\mathcal{W}_\bullet$ . We cannot have  $h(q) \cap q^\uparrow = \{\mathfrak{c}_\bullet^\uparrow\}$  by Lemma 21.2 for  $\mathcal{W}_\bullet$ . So  $h(q)$  must begin in  $q^\uparrow$ . As  $\mathfrak{c}_\bullet^\uparrow = f_1^\uparrow$ , we must have  $\mathfrak{c}_\bullet^\uparrow \leq_{h(q)} h(f_1)$  (otherwise there is not enough room for  $h(q)$  in  $q^\uparrow$ ). If  $\mathfrak{c}_\bullet^\uparrow <_{h(q)} h(f_1)$ , then let  $\ell$  be such that  $h(e)$  is in  $q^{i_1+\ell}$  and  $\mathfrak{t}^{i_1+\ell} <_{q^{i_1+\ell}} h(e)$ . Then  $h(e) <_{q^{i_1+\ell}} e^{i_1+\ell}$  must hold, as otherwise both  $h(t^{i_1+\ell}(\mathfrak{t}^{i_1+\ell}))$  and  $h(t^{i_1+\ell}(\mathfrak{f}^{i_1+\ell}))$  would be contacts in  $\mathcal{W}_\bullet$ . Therefore, the sub-ABox  $\mathcal{H}$  of  $\mathcal{B}$  consisting of  $q^\uparrow$  and  $q^{i_1+1}, \dots, q^{i_1+\ell}$  is a  $\ell + 1$ -chain and  $h$  satisfies the conditions of Lemma 21.1, both in (i) and (ii), with respect to the restriction  $\mathcal{I}^-$  of  $\mathcal{I}$  to  $\mathcal{H}$ . Thus, by Lemma 21.1,  $h$  is not a  $q \rightarrow \mathcal{I}^-$  subhomomorphism, and so it is not a  $q \rightarrow \mathcal{I}$  subhomomorphism either.

So suppose that  $h(f_1) = \mathfrak{c}_\bullet^\uparrow = \mathfrak{f}^{i_1} = \mathfrak{t}^{i_1+1}$ , and so all contacts in  $\mathcal{W}_\bullet$  are in  $F^{\mathcal{I}}$ . We consider the two cases:

- $t_2 < f_1$  and  $\delta(f_1, f_2) \geq \delta(t_1, f_1)$ . Then  $\mathfrak{t}^{i_1+1} = t_2^{i_1+1}$  and  $\mathfrak{f}^{i_1+1} = f_2^{i_1+1}$ , and so  $h(f_1) = t_2^{i_1+1}$ .



We track the location of  $h(f_2)$ . On the one hand,

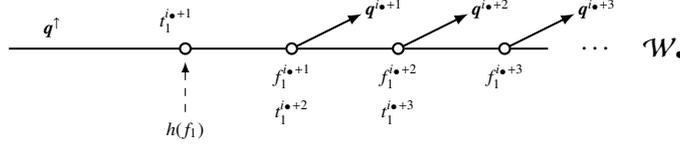
$$\delta_{h(q)}(t_2^{i_1+1}, h(f_2)) = \delta_{h(q)}(h(f_1), h(f_2)) = \delta(f_1, f_2) < \delta(t_2, f_2) = \delta_{q^{i_1+1}}(t_2^{i_1+1}, f_2^{i_1+1}).$$

On the other,

$$\delta_{h(q)}(t_2^{i_*+1}, h(f_2)) = \delta_{h(q)}(h(f_1), h(f_2)) = \delta(f_1, f_2) \geq \delta(t_1, f_1) > \delta(t_2, f_1) = \delta_{q^{i_*+1}}(t_2^{i_*+1}, f_1^{i_*+1}).$$

Therefore,  $h(f_2)$  is a node between  $f_1^{i_*+1}$  and  $f_2^{i_*+1}$ . But there is no such an  $F$ -node in  $q^{i_*+1}$ .

- Either  $f_1 < t_2$  or  $\delta(f_1, f_2) < \delta(t_1, f_1)$ . Then  $t^{i_*+k} = t_1^{i_*+k}$  and  $f^{i_*+k} = f_1^{i_*+k}$ , for all  $k$ ,  $1 \leq k \leq |q|$ , and so  $h(f_1) = t_1^{i_*+1}$ .



If  $\delta(f_1, f_2) < \delta(t_1, f_1)$ , then  $h(f_2)$  is between  $t_1^{i_*+1}$  and  $f_1^{i_*+1}$ , but there is no such an  $F$ -node in  $q^{i_*+1}$ .

So suppose that  $f_1 < t_2$ . Then  $h(t_2)$  must be in the final cog of  $q^{i_*+\ell}$ , for some  $\ell$  with  $1 \leq \ell \leq |q|$  (as all contacts in  $W_\bullet$  are in  $F^I$ , and there are no  $T$ -nodes between  $t_1^{i_*+k}$  and  $f_1^{i_*+k}$  for any  $k$ ). Thus,

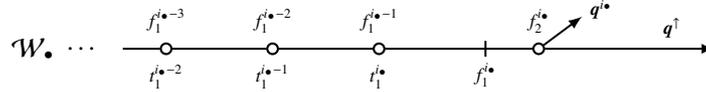
$$\delta(f_1, t_2) = \delta_{h(q)}(h(f_1), h(t_2)) = \delta_{h(q)}(t_1^{i_*+1}, h(t_2)) > \delta_{q^{i_*+\ell}}(f_1^{i_*+\ell}, h(t_2)).$$

On the other hand, as  $t_1 < f_1 < t_2$ , there is no  $T$ -node between  $f_1^{i_*+\ell}$  and  $t_2^{i_*+\ell}$ , and so

$$\delta(f_1, t_2) = \delta_{q^{i_*+\ell}}(f_1^{i_*+\ell}, t_2^{i_*+\ell}) \leq \delta_{q^{i_*+\ell}}(f_1^{i_*+\ell}, h(t_2)),$$

which is a contradiction.

- (3)<sup>\uparrow</sup>  $h(f_1)$  is in  $q^{i_*-k}$  in  $W_\bullet$ , for some  $k \leq |q|$ , and  $h(f_1) \neq c_\bullet^\uparrow$ . As  $t^{i_*} = f_2^{i_*}$ ,  $h(f_1)$  cannot be in the final cog of  $q^{i_*}$  (otherwise there is no room for  $h(q)$  in  $q^{i_*}$ ). As  $t^{i_*} = f_2^{i_*}$  and  $t^{i_*-k} = f_1^{i_*-k}$ , for  $1 \leq k \leq |q|$ ,  $h(f_1) = f_1^{i_*-\ell}$  must hold for some  $\ell \leq |q|$  (otherwise  $h(q) \subseteq W_\bullet$ , which we cannot have by Lemma 21.2). So  $h(t_1) = t^{i_*-\ell} = t_1^{i_*-\ell}$  is a contact in  $W_\bullet$ . But then either  $h(f_2)$  (when  $\ell = 0$ ) or  $h(f_1)$  (when  $\ell > 0$ ) is also a contact in  $W_\bullet$ , a contradiction.



- (4)<sup>\uparrow</sup>  $h(f_1)$  is in  $q^\uparrow \cup W_\circ$ , and  $h(f_1) \neq c_\bullet^\uparrow$ . We must have

$$c_\bullet^\uparrow = f_1^\uparrow \prec_{h(q)} h(f_1), \tag{23}$$

for otherwise there is not enough room for  $h(q)$  in  $q^\uparrow$ . We first track the location of  $h(f_2)$ . By (23), we have  $c_\circ^\uparrow = f_2^\uparrow \prec_{h(q)} h(f_2)$ , and so  $h(f_2)$  cannot be in  $q^\uparrow$  (otherwise there is not enough room for  $h(q)$  in  $q^\uparrow$ ). As  $t^{i_*} = f_2^{i_*}$ ,  $h(f_2)$  cannot be in the final cog of  $q^{i_*}$  (otherwise there is not enough room for  $h(q)$  in  $q^{i_*}$ ). Thus,

$$h(f_2) \text{ is in } q^{i_*+k} \text{ for some } k \text{ with } 1 \leq k \leq |q|. \tag{24}$$

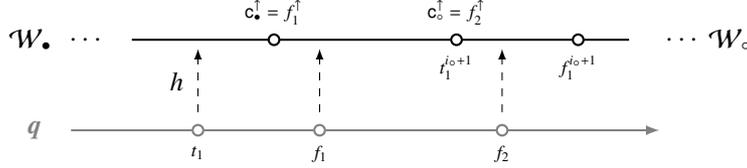
Next, we track the location of  $h(t_1)$ . There are three cases:

- $h(t_1)$  is in  $W_\circ$ . Then, as the part of  $q$  preceding  $t_1$  is *empty* (containing no  $T$ - or  $F$ -nodes), there exists a subhomomorphism from  $q$  to  $W_\circ$ , contrary to Lemma 21.2 for  $W_\circ$ .
- $h(t_1)$  is in  $q^\uparrow$ . Then  $t_1^\uparrow \prec_{q^\uparrow} h(t_1)$  by (23). As the part of  $q$  preceding  $t_1$  is *empty*, there exists a subhomomorphism  $h' : q \rightarrow \mathcal{I}$  such that  $h'(s)$  is in  $q^\uparrow$ ,  $s^\uparrow \prec_{q^\uparrow} h'(s)$  and  $h'(q) \subseteq q^\uparrow \cup W_\circ$ . Let  $\ell \leq |q|$  be such that  $h'(e)$  is in  $q^{i_*+\ell}$  and  $t^{i_*+\ell} \prec_{q^{i_*+\ell}} h'(e)$ . By (24),  $\ell \geq 1$ . Also, we have  $h'(e) \prec_{q^{i_*+\ell}} e^{i_*+\ell}$ , as otherwise both  $h(t^{i_*+\ell}(t^{i_*+\ell}))$  and  $h(t^{i_*+\ell}(t^{i_*+\ell}))$  would be contacts in  $W_\circ$ . Therefore, the sub-ABox  $\mathcal{H}$  of  $\mathcal{B}$  consisting of  $q^\uparrow$  and  $q^{i_*+1}, \dots, q^{i_*+\ell}$  is a  $\ell + 1$ -chain and  $h'$  satisfies the conditions of Lemma 21.1, both in (i) and (ii), with respect to the restriction  $\mathcal{I}^-$  of  $\mathcal{I}$  to  $\mathcal{H}$ . Thus, by Lemma 21.1,  $h'$  is not a  $q \rightarrow \mathcal{I}^-$  subhomomorphism, and so it is not a  $q \rightarrow \mathcal{I}$  subhomomorphism either.

- $h(t_1)$  is in  $\mathcal{W}_\bullet \setminus \{\mathbf{c}_\bullet^\uparrow\}$ . Then  $h(t_1) \prec_{h(q)} \mathbf{c}_\bullet^\uparrow$ . We use this fact and (24) to compute the distance from  $\mathbf{c}_\bullet^\uparrow = t_1^{i_0+1} = f_2^\uparrow$  to  $h(f_2)$ :

$$\delta_{h(q)}(\mathbf{c}_\bullet^\uparrow, h(f_2)) = \delta_{h(q)}(\mathbf{c}_\bullet^\uparrow, h(f_1)) < \delta_{h(q)}(h(t_1), h(f_1)) = \delta(t_1, f_1) = \delta_{q^{i_0+1}}(t_1^{i_0+1}, f_1^{i_0+1}),$$

and so  $h(f_2)$  is in  $q^{i_0+1}$  between  $t_1^{i_0+1}$  and  $f_1^{i_0+1}$ . But there is no  $F$ -node there.



Next, we deal with the case when  $h(q) \cap q^\downarrow \neq \emptyset$ . (The proof is similar but not totally symmetrical to the  $q^\uparrow$  cases because of our assumption that  $t_1 < f_1$ .) We track the location of  $h(t_1)$ . Four cases (1) $^\downarrow$ –(4) $^\downarrow$  are possible:

- (1) $^\downarrow$   $h(t_1) = \mathbf{c}_\bullet^\downarrow = t_1^\downarrow$  and  $h(t_2)$  is in  $q^\downarrow$ . Then  $h(t_2) = \mathbf{c}_\bullet^\downarrow = t_2^\downarrow$ , and so either  $h(t_1)$  or  $h(t_2)$  is in  $F^I$ .
- (2) $^\downarrow$   $h(t_1)$  is in  $q^{j_0+k}$  in  $\mathcal{W}_\circ$  for some  $1 \leq k \leq |q|$ , and  $h(t_2)$  is also in  $\mathcal{W}_\circ$ . We cannot have  $h(q) \cap q^\downarrow = \{\mathbf{c}_\bullet^\downarrow\}$  by Lemma 21.2 for  $\mathcal{W}_\circ$ . So  $h(q)$  must begin in  $q^\downarrow$ . As  $\mathbf{c}_\bullet^\downarrow = t_1^\downarrow$ , we must have that  $\mathbf{c}_\bullet^\downarrow \leq_{h(q)} h(t_1)$  (otherwise there is not enough room for  $h(q)$  in  $q^\downarrow$ ). Then, as the part of  $q$  preceding  $t_1$  is empty, there exists a subhomomorphism from  $q$  to  $\mathcal{W}_\circ$ , contradicting Lemma 21.2 for  $\mathcal{W}_\circ$ .
- (3) $^\downarrow$   $h(t_1)$  is in  $q^{j_0-k}$  in  $\mathcal{W}_\circ$  for some  $k \leq |q|$ , and  $h(t_1) \neq \mathbf{c}_\bullet^\downarrow$ . As  $f^{j_0} = f_1^{j_0}$  and  $t_1 < f_1$ ,  $h(t_1)$  cannot be in the final cog of  $q^{j_0}$  (otherwise there is not enough room for  $h(q)$  in  $q^{j_0}$ ). So  $h(t_1) \prec_{h(q)} \mathbf{c}_\bullet^\downarrow$  and  $h(q)$  starts in  $\mathcal{W}_\circ$ . As we cannot have  $h(q) \cap q^\downarrow = \{\mathbf{c}_\bullet^\downarrow\}$  by Lemma 21.2 for  $\mathcal{W}_\circ$ ,  $h(q)$  must continue in  $q^\downarrow$  (and end either in  $q^\downarrow$  or in  $\mathcal{W}_\bullet$ ). As  $t^{j_0-k} = t_1^{j_0-k}$  and  $f^{j_0-k} = f_1^{j_0-k}$  for  $k \leq |q|$ ,  $h(t_1)$  cannot be a contact in  $\mathcal{W}_\circ$  different from  $\mathbf{c}_\bullet^\downarrow$ , otherwise  $h(f_1)$  would also be a contact in  $\mathcal{W}_\circ$ ;  $h(t_1)$  cannot be in the initial cog of  $q^{j_0-k}$  for any  $k \leq |q|$ , otherwise there would not be enough room for  $h(q)$  in that cog. Thus,  $h(t_1) = t^{j_0-m}$  for some  $m \leq |q|$  and  $T$ -node  $t$  in  $q$  with  $t_1 < t < f_1$ , and so  $t_2 < f_1$ . We track the location of  $h(f_1)$ . If  $h(f_1)$  is in  $\mathcal{W}_\circ$ , then  $h(f_1)$  cannot be in the final cog of  $q^{j_0}$ , as  $h(q)$  continues in  $q^\downarrow$ . So  $m \geq 1$  must hold and

$$\delta(t_1, t) = \delta_{q^{j_0-m}}(t_1^{j_0-m}, t^{j_0-m}) = \delta_{h(q)}(t_1^{j_0-m}, h(t_1)) = \delta_{h(q)}(t_1^{j_0-m+1}, h(f_1)).$$

Thus,  $h(f_1)$  is an  $F$ -node between  $t_1^{j_0-m+1}$  and  $f_1^{j_0-m+1}$ . But there is no such  $F$ -node, so  $h(f_1)$  cannot be in  $\mathcal{W}_\circ$ .

If  $h(f_1)$  is in  $q^\downarrow \setminus \{\mathbf{c}_\bullet^\downarrow\}$ , then we cannot have  $h(f_1) = f_1^\downarrow$ , as otherwise we had  $h(t_1) = t_1^\downarrow = \mathbf{c}_\bullet^\downarrow$ . As  $t_2 < f_1$  and  $\mathbf{c}_\bullet^\downarrow = t_2^\downarrow$ , we cannot have  $f_1^\downarrow \prec_{q^\downarrow} h(f_1)$ , otherwise there is not enough room for  $h(q)$  in  $q^\downarrow$ . Therefore,  $h(f_1) \prec_{q^\downarrow} f_1^\downarrow$ , and so  $h(f_1) = \mathbf{c}_\bullet^\downarrow$  must hold. So in any case,  $h(f_1)$  must be in  $\mathcal{W}_\bullet$ . As  $f_1^{j_0} \leq_{q^{j_0}} f^{j_0}$ ,  $h(f_1)$  cannot be in the final cog of  $q^{j_0}$  (otherwise there is not enough room for  $h(q)$  in  $q^{j_0}$ ). So  $h(f_1)$  is in  $q^{j_0+k}$  for some  $k$  with  $1 \leq k \leq |q|$ . We consider the two cases:

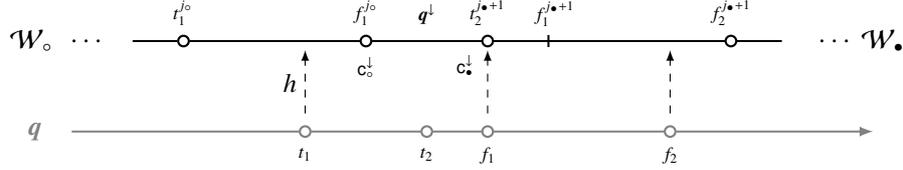
- $\delta(f_1, f_2) \geq \delta(t_1, f_1)$ , and so  $t^{j_0+k} = t_2^{j_0+k}$  and  $f^{j_0+k} = f_2^{j_0+k}$ , for all  $1 \leq k \leq |q|$ .  $h(f_1)$  cannot be in the final cog of any  $q^{j_0+k}$  (otherwise there is not enough room for  $h(q)$  in  $q^{j_0+k}$ ). If  $h(f_1) = f_1^{j_0+k}$  for some  $1 \leq k \leq |q|$ , then  $h(t_2)$  and  $h(f_2)$  are both contacts in  $\mathcal{W}_\bullet$ . So suppose that  $h(f_1) = t_2^{j_0+\ell}$  is a contact in  $\mathcal{W}_\bullet$  for some  $1 \leq \ell \leq |q|$ . We track the location of  $h(f_2)$ . On the one hand,

$$\delta_{h(q)}(t_2^{j_0+\ell}, h(f_2)) = \delta_{h(q)}(h(f_1), h(f_2)) = \delta(f_1, f_2) < \delta(t_2, f_2) = \delta_{q^{j_0+\ell}}(t_2^{j_0+\ell}, f_2^{j_0+\ell}).$$

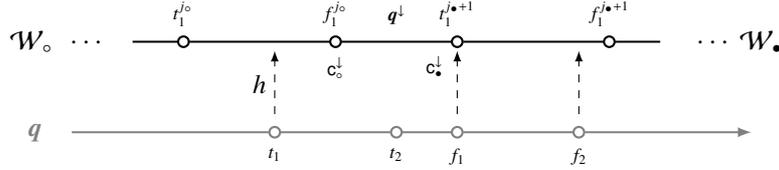
On the other,

$$\delta_{h(q)}(t_2^{j_0+\ell}, h(f_2)) = \delta_{h(q)}(h(f_1), h(f_2)) = \delta(f_1, f_2) \geq \delta(t_1, f_1) > \delta(t_2, f_1) = \delta_{q^{j_0+\ell}}(t_2^{j_0+\ell}, f_1^{j_0+\ell}),$$

and so  $h(f_2)$  is a node between  $f_1^{j_0+\ell}$  and  $f_2^{j_0+\ell}$ . But there is no such  $F$ -node in  $q^{j_0+\ell}$ .



- $\delta(f_1, f_2) < \delta(t_1, f_1)$ , and so  $t^{j_0+k} = t_1^{j_0+k}$  and  $f^{j_0+k} = f_1^{j_0+k}$ , for all  $1 \leq k \leq |q|$ .  $h(f_1)$  cannot be in the final cog of any  $q^{j_0+k}$  (otherwise there is not enough room for  $h(q)$  in  $q^{j_0+k}$ ). So  $h(f_1) = t_1^{j_0+\ell}$  is a contact in  $W_\bullet$  for some  $1 \leq \ell \leq |q|$ , and so  $h(f_2)$  is between  $t_1^{j_0+\ell}$  and  $f_1^{j_0+\ell}$ . But there is no such  $F$ -node in  $q^{j_0+\ell}$ .



- (4) $^\perp$   $h(t_1)$  is in  $q^\perp \cup W_\bullet$ , and  $h(t_1) \neq c_\bullet^\perp$ . We must have  $c_\bullet^\perp = t_1^\perp <_{h(q)} h(t_1)$  (otherwise there is not enough room for  $h(q)$  in  $q^\perp$ ). If  $h(t_1)$  is in  $q^\perp$ , then  $c_\bullet^\perp = t_2^\perp \leq_{q^\perp} h(t_1)$  follows. We cannot have  $c_\bullet^\perp = t_2^\perp <_{q^\perp} h(t_1)$  (otherwise there is not enough room for  $h(q)$  in  $q^\perp$ ), and so  $h(t_1) = c_\bullet^\perp$  must hold. So in any case,  $h(t_1)$  is in  $W_\bullet$ . Then, as the part of  $q$  preceding  $t_1$  is empty, there exists a subhomomorphism from  $q$  to  $W_\bullet$ , contrary to Lemma 21.2 for  $W_\bullet$ .

This completes the proof of Lemma 21.3.  $\square$

Let  $\psi$  be a 3CNF with  $n_\psi$  clauses of the form  $\ell_1 \vee \ell_2 \vee \ell_3$ , where each  $\ell_i$  is a literal, and let  $n > (n_\psi + 2) \cdot |q|$ . For each propositional variable  $p$  in  $\psi$ , we take a fresh  $n$ -bike  $\mathcal{B}^p$  having  $n$ -cogwheels  $W_\bullet^p$ ,  $W_\circ^p$  and satisfying the conditions in Lemma 21.3. We pick three nodes  $v_1, v_2$  and  $v_3$  in  $q$  such that each  $v_a$  is a  $T$ -node or an  $F$ -node, and  $v_1 < v_2 < v_3$ . Then, for every clause  $c = (\ell_1^c \vee \ell_2^c \vee \ell_3^c)$  in  $\psi$ , we proceed as follows. We take a fresh copy  $q^c$  of  $q$ , consider the copies  $v_1^c, v_2^c$  and  $v_3^c$  of the chosen nodes in  $q^c$ , and replace their  $F$ - or  $T$ -labels with  $A$ . Then, for  $a = 1, 2, 3$ , we glue  $v_a^c$  to a fresh (unused as  $T$ - or  $F$ -connections) contact

- (p1) in  $W_\bullet^p$  iff either  $\ell_a^c = p$  and  $v_a$  is an  $F$ -node in  $q$ , or  $\ell_a^c = \neg p$  and  $v_a$  is a  $T$ -node in  $q$ ;

- (p2) in  $W_\circ^p$  iff either  $\ell_a^c = p$  and  $v_a$  is a  $T$ -node in  $q$ , or  $\ell_a^c = \neg p$  and  $v_a$  is a  $F$ -node in  $q$ .

We call the chosen contacts in the three  $n$ -cogwheels the *wheel-contacts for  $c$* . For example, if  $q$  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  $> 2|q|$ . We treat the resulting labelled graph as an ABox and denote it by  $\mathcal{A}_\psi$ . Clearly, the size of  $\mathcal{A}_\psi$  is polynomial in the sizes of  $q$  and  $\psi$ .

The following lemma is a consequence of the definition of  $\mathcal{A}_\psi$ , and the ‘easy’ ( $\Rightarrow$ ) direction of Lemma 21.3.

**Lemma 21.4.** *If  $\text{cov}_A, \mathcal{A}_\psi \not\models q$ , then  $\psi$  is satisfiable.*

*Proof.* Suppose  $\mathcal{I}$  is a model of  $\text{cov}_A$  and  $\mathcal{A}_\psi$  such that  $\mathcal{I} \not\models q$ . As for each variable in  $\psi$ , the  $n$ -bike  $\mathcal{B}^p$  satisfies the conditions in Lemma 21.3, we have that all contacts in the  $n$ -cogwheel  $W_\bullet^p$  are in  $F^{\mathcal{I}}$  and all contacts in  $W_\circ^p$  are in  $T^{\mathcal{I}}$  or the other way round. As  $\mathcal{I} \not\models q$ , for every clause  $c = (\ell_1^c \vee \ell_2^c \vee \ell_3^c)$  in  $\psi$ , there is  $a \in \{1, 2, 3\}$  such that either  $v_a$  is a  $T$ -node in  $q$  but  $v_a^c \in F^{\mathcal{I}}$ , or  $v_a$  is an  $F$ -node in  $q$  but  $v_a^c \in T^{\mathcal{I}}$ . Define an assignment  $a$  by setting  $a(\ell_a^c) = T$  for each clause  $c$  in  $\psi$  (and arbitrary otherwise). We claim that  $a$  is well-defined in the sense that we never set both  $a(p) = T$

and  $\alpha(\neg p) = T$ . Indeed, suppose otherwise. Suppose also that the former is because of  $\ell_{a_1}^{c_1}$  in a clause  $c_1$  and the latter because of  $\ell_{a_2}^{c_2}$  in a clause  $c_2$ .

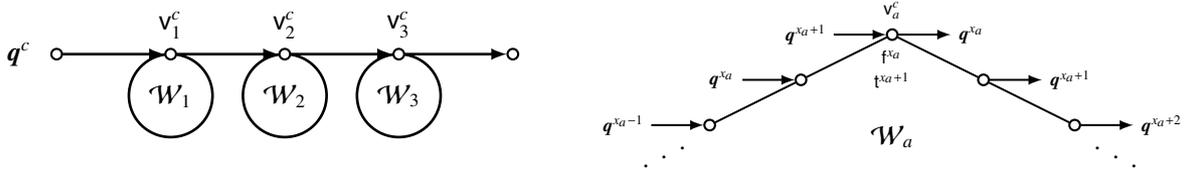
*Case 1:*  $v_{a_1}$  is a  $T$ -node in  $\mathbf{q}$  but  $v_{a_1}^{c_1} \in F^I$ . As  $\alpha(p) = T$  implies that  $\ell_{a_1}^{c_1} = p$ , by (p2) of the construction  $v_{a_1}^{c_1}$  is a contact in the  $n$ -cogwheel  $\mathcal{W}_o^p$ . So all contacts in  $\mathcal{W}_o^p$  are in  $F^I$ . On the other hand,  $\alpha(\neg p) = T$  implies that  $\ell_{a_2}^{c_2} = \neg p$ . If  $v_{a_2}$  is a  $T$ -node in  $\mathbf{q}$  but  $v_{a_2}^{c_2} \in F^I$ , then  $v_{a_2}^{c_2}$  is a contact in  $\mathcal{W}_o^p$  by (p1), and so all contacts in  $\mathcal{W}_o^p$  are also in  $F^I$ , a contradiction. And if  $v_{a_2}$  is an  $F$ -node in  $\mathbf{q}$  but  $v_{a_2}^{c_2} \in T^I$ , then  $v_{a_2}^{c_2}$  is a contact in  $\mathcal{W}_o^p$  by (p2), and so all contacts in  $\mathcal{W}_o^p$  are in  $T^I$ , a contradiction again.

*Case 2:*  $v_{a_1}$  is an  $F$ -node in  $\mathbf{q}$  but  $v_{a_1}^{c_1} \in T^I$ . This case is similar and left to the reader.

Thus, the assignment  $\alpha$  is well-defined and makes true at least one literal in every clause in  $\psi$ .  $\square$

It remains to find some conditions on  $\mathcal{A}_\psi$  that would guarantee that the converse of Lemma 21.4 also holds. So suppose that  $\psi$  is satisfiable under an assignment  $\alpha$ . We define a model  $\mathcal{I}_\alpha$  of  $\text{cov}_A$  and  $\mathcal{A}_\psi$  as follows: for every variable  $p$  in  $\psi$ , if  $\alpha(p) = T$  then we put all contacts of the  $n$ -cogwheel  $\mathcal{W}_o^p$  to  $T^{\mathcal{I}_\alpha}$  and all contacts of the  $n$ -cogwheel  $\mathcal{W}_o^{\neg p}$  to  $F^{\mathcal{I}_\alpha}$ ; if  $\alpha(p) = F$ , we put all contacts of  $\mathcal{W}_o^p$  to  $F^{\mathcal{I}_\alpha}$  and all contacts of  $\mathcal{W}_o^{\neg p}$  to  $T^{\mathcal{I}_\alpha}$ . We aim to find some conditions on  $\mathcal{A}_\psi$  that would imply  $\mathcal{I}_\alpha \not\models \mathbf{q}$ .

To formulate these conditions, we introduce some new notation for the three wheel-contacts, uniformly for any given clause  $c$  in  $\psi$  (that is, depending not on  $c$ , but only on  $a = 1, 2, 3$  and  $\mathbf{q}$ ). For each  $a = 1, 2, 3$ , we let  $\mathcal{W}_a$  denote the  $n$ -cogwheel the node  $v_a^c$  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  $\mathbf{q}^{x_a}$  and the  $T$ -node  $t^{x_a+1}$  of some copy  $\mathbf{q}^{x_a+1}$  ( $\pm$  is modulo  $n$ ).



For each  $a = 1, 2, 3$ , we need to choose the contacts  $v_a$ ,  $f^{x_a \pm k}$  and  $t^{x_a \pm k}$ , for  $k \leq |\mathbf{q}|$ , in such a way that  $\mathcal{I}_\alpha \not\models \mathbf{q}$  (and so the converse of Lemma 21.4 holds). There might be different ways of choosing these contacts so that all potential  $\mathbf{q} \rightarrow \mathcal{I}_\alpha$  homomorphisms are excluded. Our algorithm below selects contacts that are suitable for  $\psi$  and  $\mathbf{q}$  uniformly, depending only on the particular 2-CQ  $\mathbf{q}$ , but not on the satisfying assignment  $\alpha$ . 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 21.3. We do not have any information about the particular labelings of  $v_1^c$ ,  $v_2^c$  and  $v_3^c$  in  $\mathcal{I}_\alpha$  other than the fact that the cogwheel attached to each of them represents a truth-value:

$$\text{for each } a = 1, 2, 3, \text{ the contacts in } \mathcal{W}_a \text{ are either all in } T^{\mathcal{I}_\alpha} \text{ or all in } F^{\mathcal{I}_\alpha}. \quad (25)$$

Throughout, we use the following notation:  $t_\square$  denotes the last  $T$ -node preceding  $f_1$ ,  $t_\diamond$  denotes the last  $T$ -node preceding  $f_2$ , and  $t_\#$  denotes the last  $T$ -node preceding  $f_{last}$ . (These all are well-defined, as  $t_1 < f_1$  by our assumption.) We call  $\mathcal{A}_\psi$  a  $\psi$ -gadget if the following conditions hold, for all clauses  $c$  in  $\psi$ , all  $k \leq |\mathbf{q}|$ , and all  $\ell$  with  $1 \leq \ell \leq |\mathbf{q}|$ :

- $v_1 = t_1$ ,  $t^{x_1+1} = t_1^{x_1+1}$ ,  $t^{x_1-k} = t_\square^{x_1-k}$ ;
- if  $t_{last} < f_1$  then  $f^{x_1-k} = f_2^{x_1-k}$ ,  $v_2 = t_{last}$ ,  $t^{x_2+\ell} = t_{last-1}^{x_2+\ell}$ ,  $f^{x_2+\ell} = f_1^{x_2+\ell}$ ,  $t^{x_2-k} = t_1^{x_2-k}$ ,  

$$f^{x_2-k} = \begin{cases} f_1^{x_2-k}, & \text{if } \delta(t_1, t_2) = \dots = \delta(t_{last-1}, t_{last}) = \delta(t_{last}, f_2), \\ f_2^{x_2-k}, & \text{otherwise;} \end{cases}$$
- if  $f_1 < t_{last}$  then  $f^{x_1-k} = f_1^{x_1-k}$ ,  $v_2 = f_1$ ,  $t^{x_2+\ell} = t_\square^{x_2+\ell}$ ,  $f^{x_2+\ell} = f_1^{x_2+\ell}$ ,  $f^{x_2-k} = f_2^{x_2-k}$ , and we consider two cases:
  - (i) if  $f_1 < t_\diamond$  and there exist some  $T$ -node  $t < t_\diamond$  and  $k_t \geq 1$  with  $\delta(t_\square, f_1) = \delta(t, t_\diamond) + k_t \cdot \delta(t_\diamond, f_2)$ , then let  $t_\star$  be such a  $t$  with the smallest  $k_t$ ,  $t^{x_2-(k_t-1)} = t_\star^{x_2-(k_t-1)}$ , and  $t^{x_2-k} = t_\diamond^{x_2-k}$  for  $k \neq k_t - 1$ ;
  - (ii) otherwise,  $t^{x_2-k} = t_\diamond^{x_2-k}$ ;
- $v_3 = f_{last}$ ,  $t^{x_3-k} = t_\#^{x_3-k}$ ,  $f^{x_3-k} = f_{last}^{x_3-k}$ ,  $t^{x_3+\ell} = t_\#^{x_3+\ell}$ ,  $f^{x_3+\ell} = f_{last}^{x_3+\ell}$ .

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

**Lemma 21.5.** *If  $\mathcal{A}_\psi$  is a  $\psi$ -gadget, then  $\mathcal{I}_a \not\models \mathbf{q}$ .*

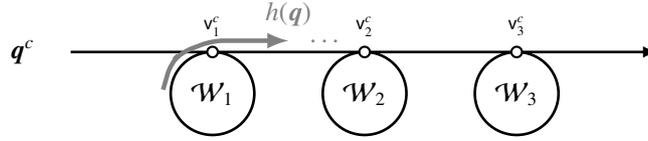
*Proof.* The proof is via excluding all possible locations in  $\mathcal{A}_\psi$  for the image  $h(\mathbf{q})$  of a potential subhomomorphism  $h: \mathbf{q} \rightarrow \mathcal{I}_a$ . We begin with the following observation: for any clause  $c$  in  $\psi$ ,

$$\text{there is no subhomomorphism } h: \mathbf{q} \rightarrow \mathcal{I}_a \text{ such that } h(v_a) = v_a^c \text{ for all } a = 1, 2, 3. \quad (26)$$

(In particular, there is no subhomomorphism from  $\mathbf{q}$  onto  $\mathbf{q}^c$  in  $\mathcal{I}_a$ .) Indeed, suppose on the contrary that there is such a subhomomorphism  $h$  for some  $c$ . Suppose  $a(\ell_a^c) = T$  for some  $a$ . If  $\ell_a^c = p$ , then either  $v_a$  is an  $F$ -node in  $\mathbf{q}$  but  $v_a^c \in T^{\mathcal{I}_a}$  as it is in  $\mathcal{W}_a^p$ , or  $v_a$  is a  $T$ -node in  $\mathbf{q}$  but  $v_a^c \in F^{\mathcal{I}_a}$  as it is in  $\mathcal{W}_a^p$ , both are impossible when  $h(v_a) = v_a^c$ . The case of  $\ell_a^c = \neg p$  is dually symmetric. It follows that  $a(\ell_a^c) \neq T$  for any  $a = 1, 2, 3$ , contrary to a satisfying  $\psi$ .

Because of Lemma 21.3,  $h(\mathbf{q})$  must intersect with some copy  $\mathbf{q}^c$  for some clause  $c$ . By (26),  $h(\mathbf{q})$  must properly intersect with at least one of the  $n$ -cogwheels glued to  $\mathbf{q}^c$  in the sense that  $h(\mathbf{q}) \cap \mathcal{W}_a \not\subseteq \{c_a\}$  for some  $a = 1, 2, 3$ . By Lemma 21.2, we may assume that  $h(\mathbf{q}) \not\subseteq \mathcal{W}_a$  for any  $a = 1, 2, 3$ . Also by Lemma 21.2, we may assume that if  $h(\mathbf{q})$  properly intersects with  $\mathcal{W}_a$ , then every node in  $h(\mathbf{q}) \cap \mathcal{W}_a$  is in  $\mathbf{q}^{x_a \pm k}$  for some  $k \leq |\mathbf{q}|$ . As the wheel-contacts for different clauses in each  $n$ -cogwheel are far from each other, there is a unique  $c$  with  $h(\mathbf{q})$  properly intersecting with one or two of the  $n$ -cogwheels  $\mathcal{W}_1, \mathcal{W}_2$  and  $\mathcal{W}_3$  glued to  $\mathbf{q}^c$  (it cannot properly intersect with all three). It is not hard to check that, by (26), all options for such a  $h(\mathbf{q})$  are covered by the six cases (1)–(6) below, and we need to show that none of them is possible.

- (1)  $h(\mathbf{q})$  starts in  $\mathcal{W}_1$  and  $h(v_1) \prec_{h(\mathbf{q})} v_1^c$ .

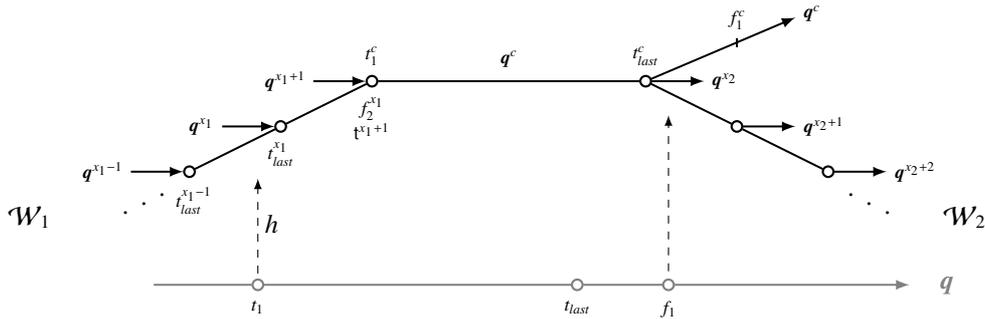


As  $v_1 = t_1$ , we have

$$h(t_1) \prec_{h(\mathbf{q})} t_1^c \quad (27)$$

and  $h(t_1)$  is in  $\mathcal{W}_1$ . We track the location of  $h(f_1)$ . We consider the two further subcases  $t_{last} < f_1$  and  $f_1 < t_{last}$ .

*Case  $t_{last} < f_1$ .* Then  $v_2 = t_{last}$ . As there is no  $F$ -node between  $t_1^c$  and  $t_{last}^c$  in  $\mathbf{q}^c$ , either  $t_{last}^c \leq_{h(\mathbf{q})} h(f_1)$  or  $h(f_1)$  is in  $\mathcal{W}_1$ . Suppose first that  $t_{last}^c \leq_{h(\mathbf{q})} h(f_1)$ .



As  $h(t_{last}) \prec_{h(\mathbf{q})} t_{last}^c$  by (27), we have

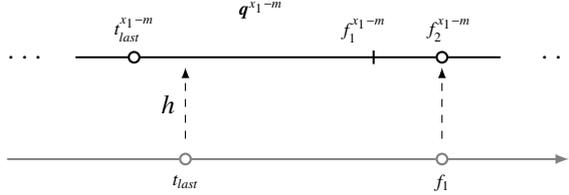
$$\delta(t_{last}, f_1) = \delta_{h(\mathbf{q})}(h(t_{last}), h(f_1)) > \delta_{h(\mathbf{q})}(h(t_{last}), t_{last}^c) = \delta_{h(\mathbf{q})}(h(t_1), t_1^c). \quad (28)$$

On the other hand, we analyse the location of  $h(t_1)$  in  $\mathcal{W}_1$ . As  $t_1^{x_1+1} = t_1^{x_1+1}$ ,  $h(t_1)$  cannot be in the initial cog of  $\mathbf{q}^{x_1+1}$  (otherwise there is not enough room for  $h(\mathbf{q})$  in  $\mathbf{q}^{x_1+1}$ ). As  $t_1^{x_1} = t_{last}^{x_1} = t_{last}^{x_1}$ ,  $f_1^{x_1} = f_2^{x_1}$ , and there is no  $T$ -node between  $t_{last}$  and  $f_2$ , we have

$$\delta_{h(\mathbf{q})}(h(t_1), t_1^c) = \delta_{h(\mathbf{q})}(h(t_1), f_2^{x_1}) \geq \delta_{q^1}(t_{last}^{x_1}, f_2^{x_1}) = \delta(t_{last}, f_2) > \delta(t_{last}, f_1),$$

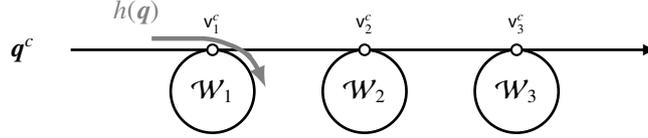
contrary to (28).

Next, suppose that  $h(f_1)$  is in  $\mathcal{W}_1$ , and so  $h(f_1) \leq_{h(q)} t_1^c$ . As  $t^{x_1+1} = t_1^{x_1+1}$ ,  $h(f_1)$  cannot be in the initial cog of  $q^{x_1+1}$  (otherwise there is no room for  $h(q)$  in  $q^{x_1+1}$ ). As  $t^{x_1-k} = t_{\square}^{x_1-k} = t_{last}^{x_1-k}$  and  $f^{x_1-k} = f_2^{x_1-k}$  for  $k \leq |q|$ , it follows that either  $h(f_1) = f_1^{x_1-m}$  or  $h(f_1) = f_2^{x_1-m}$  for some  $m \leq |q|$ . If  $h(f_1) = f_1^{x_1-m}$ , then both  $h(t_{last})$  and  $h(f_2)$  are contacts in  $\mathcal{W}_1$ , contradicting (25). And if  $h(f_1) = f_2^{x_1-m}$ , then  $h(t_{last})$  is a  $T$ -node between  $t_{last}^{x_1-m}$  and  $f_2^{x_1-m}$ . But there is no such  $T$ -node.



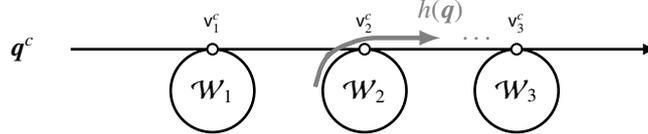
Case  $f_1 < t_{last}$ . As  $v_2 = f_1$ , it follows from (27) that  $h(f_1) <_{h(q)} f_1^c = v_2^c$ , and so  $h(f_1) \leq_{h(q)} t_1^c$  and  $h(f_1)$  is in  $\mathcal{W}_1$  (as there is no  $F$ -node between  $t_1^c$  and  $f_1^c$ ). As  $t^{x_1+1} = t_1^{x_1+1}$ ,  $h(f_1)$  cannot be in the initial cog of  $q^{x_1+1}$  (otherwise there is no room for  $h(q)$  in  $q^{x_1+1}$ ). As  $f^{x_1-k} = f_1^{x_1-k}$  for  $k \leq |q|$ ,  $h(f_1)$  must be a contact in  $\mathcal{W}_1$ . But then  $h(t_{\square})$  is a contact in  $\mathcal{W}_1$  too, as  $t^{x_1-k} = t_{\square}^{x_1-k}$  for  $k \leq |q|$ , contradicting (25).

(2)  $h(q)$  ends in  $\mathcal{W}_1$ .



Then  $v_1^c \leq_{h(q)} h(v_1)$ , as otherwise there is no room for  $h(q)$  in  $q^c$ . As  $v_1 = t_1$ , we have that  $t_1^c \leq_{h(q)} h(t_1)$  and  $h(t_1)$  is in  $\mathcal{W}_1$ . As the part of  $q$  preceding  $t_1$  is empty, there exists a subhomomorphism from  $q$  to the restriction of  $\mathcal{I}_a$  to  $\mathcal{W}_1$ , contrary to Lemma 21.2.

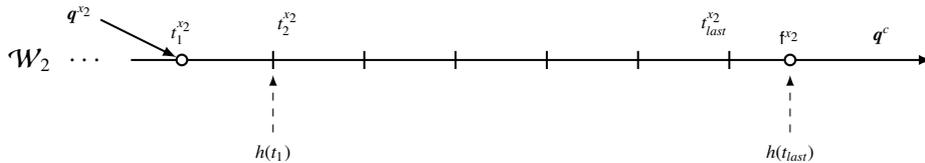
(3)  $h(q)$  starts in  $\mathcal{W}_2$  and  $h(v_2) \leq_{h(q)} v_2^c$ .



We consider the two further subcases  $t_{last} < f_1$  and  $f_1 < t_{last}$ .

Case  $t_{last} < f_1$ . Then  $v_2 = t_{last}$ , and so  $h(t_{last}) \leq_{h(q)} t_{last}^c$  and  $h(t_{last})$  is in  $\mathcal{W}_2$ . As  $t^{x_2+1} = t_{last-1}^{x_2+1}$ ,  $h(t_{last})$  cannot be in the initial cog of  $q^{x_2+1}$  (otherwise there is no room for  $h(q)$  in  $q^{x_2+1}$ ). If  $h(t_{last}) <_{h(q)} v_2^c = t_{last}^c$ , then  $h(f_{last}) <_{h(q)} v_3^c = f_{last}^c$ . As the part of  $q$  following  $f_{last}$  is empty, we may assume that  $h(q) \subseteq \mathcal{W}_2 \cup q^c$ , and so  $h(e) <_{q^c} e^c$ . Let  $k \leq |q|$  be such that  $h(s)$  is in  $q^{x_2-k}$  and  $h(s) <_{q^{x_2-k}} f^{x_2-k}$ . Then  $s^{x_2-k} <_{q^{x_2-k}} h(s)$  must hold, as otherwise both  $h(t^{x_2-k}(t^{x_2-k}))$  and  $h(t^{x_2-k}(f^{x_2-k}))$  would be contacts in  $\mathcal{W}_2$ , contradicting (25). Therefore, the sub-ABox  $\mathcal{H}$  of  $\mathcal{A}_{\psi}$  consisting of  $q^{x_2-k}, \dots, q^{x_2}$  and  $q^c$  is a  $k+2$ -chain and  $h$  satisfies the conditions of Lemma 21.1, both in (i) and (ii), with respect to the restriction  $\mathcal{I}_a^-$  of  $\mathcal{I}_a$  to  $\mathcal{H}$ . Thus, by Lemma 21.1,  $h$  is not a  $q \rightarrow \mathcal{I}_a^-$  subhomomorphism, and so it is not a  $q \rightarrow \mathcal{I}_a$  subhomomorphism either.

So suppose that  $h(t_{last}) = v_2^c = f^{x_2}$ . As  $t^{x_2} = t_1^{x_2}$ , it follows that  $h(t_1) = t_2^{x_2}, \dots, h(t_{last-1}) = t_{last}^{x_2}$  and  $\delta_{q^{x_2}}(t_1^{x_2}, t_2^{x_2}) = \dots = \delta_{q^{x_2}}(t_{last-1}^{x_2}, t_{last}^{x_2}) = \delta_{q^{x_2}}(t_{last}^{x_2}, f^{x_2})$  must hold, which cannot happen with either choice for  $f^{x_2}$ .



Case  $f_1 < t_{last}$ . Then  $v_2 = f_1$ , and so  $h(f_1) \leq_{h(q)} f_1^c$  and  $h(f_1)$  is in  $\mathcal{W}_2$ . We track the location of  $h(f_1)$ . As  $t^{x_2+1} = t_{\square}^{x_2+1}$ ,  $h(f_1)$  cannot be in the initial cog of  $q^{x_2+1}$  (otherwise there is no room for  $h(q)$  in  $q^{x_2+1}$ ). So

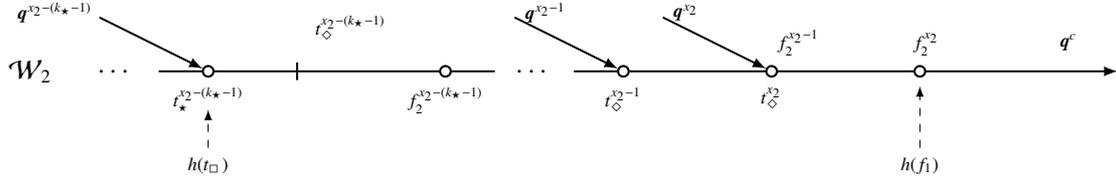
$$h(f_1) \text{ is in } q^{x_2-k} \text{ for some } k \leq |q|. \quad (29)$$

We have  $f^{x_2-k} = f_2^{x_2-k}$  for  $k \leq |q|$ . For  $t^{x_2-k}$  when  $k \leq |q|$ , we have the two cases (i) and (ii).

(i) We have  $f_1 < t_{\diamond}$  and  $\delta(t_{\square}, f_1) = \delta(t_{\star}, t_{\diamond}) + k_{t_{\star}} \cdot \delta(t_{\diamond}, f_2)$  for the  $T$ -node  $t_{\star} < t_{\diamond}$  with minimal such  $k_{t_{\star}}$ . Also,  $t^{x_2-(k_{t_{\star}}-1)} = t_{\star}^{x_2-(k_{t_{\star}}-1)}$ , and  $t^{x_2-k} = t_{\diamond}^{x_2-k}$  for all  $k \leq |q|$  with  $k \neq k_{t_{\star}} - 1$ . Suppose first that  $h(f_1) = v_2^c$ . We track the location of  $h(t_{\square})$ . By (29),  $h(t_{\square})$  is in  $q^{x_2-k}$  for some  $k \leq |q|$ . As

$$\delta_{h(q)}(h(t_{\square}), v_2^c) = \delta_{h(q)}(h(t_{\square}), h(f_1)) = \delta(t_{\square}, f_1), \quad (30)$$

either  $h(t_{\square}) = t_{\star}^{x_2-(k_{t_{\star}}-1)} = t^{x_2-(k_{t_{\star}}-1)}$  is a contact in  $\mathcal{W}_2$  contradicting (25), or  $h(t_{\square})$  is in  $q^{x_2-k}$  for some  $k < k_{t_{\star}} - 1$ . In the latter case,  $h(t_{\square})$  cannot be a contact in  $\mathcal{W}_2$  (as  $h(f_1)$  is), and it cannot be a non-cog node (as there is no  $T$ -node between  $t_{\diamond}^{x_2-k}$  and  $f_2^{x_2-k}$ ). So suppose  $h(t_{\square})$  is in the initial cog of  $q^{x_2-k}$ . Then  $h(t_{\square}) = t^{x_2-k}$  for some  $T$ -node  $t < t_{\diamond}$ , and  $\delta(t_{\square}, f_1) = \delta(t, t_{\diamond}) + (k+1) \cdot \delta(t_{\diamond}, f_2)$  by (30). As  $k+1 < k_{t_{\star}}$ , this contradicts the minimality of  $k_{\star}$  with this property.



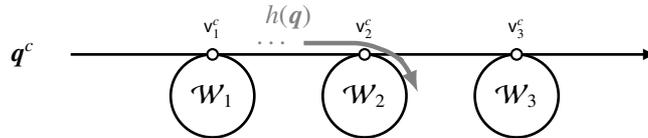
Next, suppose  $h(f_1) <_{h(q)} v_2^c$ . Then  $h(f_2) <_{h(q)} f_2^c$ , and so  $h(f_2) \leq_{h(q)} v_2^c$ . As  $t^{x_2+1} = t_{\square}^{x_2+1}$ ,  $h(f_2)$  cannot be in the initial cog of  $q^{x_2+1}$  (otherwise there is not enough room for  $h(q)$  in  $q^{x_2+1}$ ). If  $h(f_2)$  is a contact in  $\mathcal{W}_2$ , then either  $h(t_{\star})$  or  $h(t_{\diamond})$  is also a contact in  $\mathcal{W}_2$ , contradicting (25). As  $f^{x_2-k} = f_2^{x_2-k}$  for all  $k \leq |q|$ , it follows that  $h(f_2) = f_1^{x_2-\ell}$  for some  $\ell \leq |q|$ . We claim that  $f_1^{x_2-\ell}$  is in the 'initial' cog of  $q^{x_2-\ell}$  (and so there is not enough room for  $h(q)$  in  $q^{x_2-\ell}$ ). Indeed, as  $f_1 < t_{\diamond}$ , this clearly holds for  $\ell \neq k_{t_{\star}} - 1$ . So let  $\ell = k_{t_{\star}} - 1$  and suppose on the contrary that  $t_{\star} < f_1$ . Then  $t_{\star} \leq t_{\square}$  (as  $t_{\square}$  is the last  $T$ -node preceding  $f_1$ ), and so  $\delta(t_{\star}, f_1) \geq \delta(t_{\square}, f_1) > \delta(t_{\star}, t_{\diamond})$  (as  $k_{t_{\star}} \geq 1$ ), contradicting  $f_1 < t_{\diamond}$ .

(ii) We either have  $t_{\diamond} < f_1$  or there are no  $T$ -node  $t < t_{\diamond}$  and  $k_t \geq 1$  with  $\delta(t_{\square}, f_1) = \delta(t, t_{\diamond}) + k_t \cdot \delta(t_{\diamond}, f_2)$ . Also,  $t^{x_2-k} = t_{\diamond}^{x_2-k}$  and  $f^{x_2-k} = f_2^{x_2-k}$ , for all  $k \leq |q|$ .

If  $t_{\diamond} < f_1$  then we track the location of  $h(t_{\diamond})$ . By (29),  $h(t_{\diamond})$  is in  $q^{x_2-k}$  for some  $k \leq |q|$ . But it cannot be in the initial cog (otherwise there is not enough room for  $h(q)$  in  $q^{x_2-k}$ ), it cannot be a contact (otherwise  $h(f_2)$  is a contact too, contradicting (25)), and it cannot be between  $t_{\diamond}^{x_2-k}$  and  $f_2^{x_2-k}$  (as there is no  $T$ -node there).

If  $f_1 < t_{\diamond}$  but there are no  $T$ -node  $t < t_{\diamond}$  and  $k_t \geq 1$  such that  $\delta(t_{\square}, f_1) = \delta(t, t_{\diamond}) + k_t \cdot \delta(t_{\diamond}, f_2)$ , then suppose first that  $h(f_1) = v_2^c$ . We track the location of  $h(t_{\square})$ . By (29),  $h(t_{\square})$  is in  $q^{x_2-k}$  for some  $k \leq |q|$ . It cannot be a contact by (25), it cannot be in an initial cog by our assumption, and it cannot be between  $t_{\diamond}^{x_2-k}$  and  $f_2^{x_2-k}$  (as there is no  $T$ -node there). Finally, suppose that  $h(f_1) <_{h(q)} v_2^c$ . Then  $h(f_2) <_{h(q)} f_2^c$ , and so  $h(f_2) \leq_{h(q)} v_2^c$ . As  $t^{x_2+1} = t_{\square}^{x_2+1}$ ,  $h(f_2)$  cannot be in the initial cog of  $q^{x_2+1}$  (otherwise there is not enough room for  $h(q)$  in  $q^{x_2+1}$ ). If  $h(f_2)$  is a contact in  $\mathcal{W}_2$ , then  $h(t_{\diamond})$  is also a contact in  $\mathcal{W}_2$ , contradicting (25). So  $h(f_2) = f_1^{x_2-\ell}$  must hold for some  $\ell \leq |q|$ . As  $f_1 < t_{\diamond}$ ,  $f_1^{x_2-\ell}$  is in the initial cog of  $q^{x_2-\ell}$ , so there is not enough room for  $h(q)$  in  $q^{x_2-\ell}$ .

(4)  $h(q)$  ends in  $\mathcal{W}_2$  and  $v_2^c \leq_{h(q)} h(v_2)$ .



We again consider the two further subcases  $t_{last} < f_1$  and  $f_1 < t_{last}$ .

*Case  $t_{last} < f_1$ .* Then  $v_2 = t_{last}$ , and so  $t_{last}^c \leq_{h(q)} h(t_{last})$  and  $h(t_{last})$  is in  $\mathcal{W}_2$ . As  $t_{last}^{x_2} <_{q^{x_2}} f^{x_2}$ ,  $h(t_{last})$  cannot be in the final cog of  $q^{x_2}$  (otherwise there is not enough room for  $h(q)$  in  $q^{x_2}$ ). As  $t^{x_2+\ell} = t_{last-1}^{x_2+\ell}$  and  $f^{x_2+\ell} = f_1^{x_2+\ell}$  for  $1 \leq \ell \leq |q|$ , if  $h(t_{last}) = t_{last-1}^{x_2+k}$  is a contact in  $\mathcal{W}_2$  for some  $1 \leq k \leq |q|$ , then  $h(f_1)$  is in  $q^{x_2+k}$  preceding  $f_1^{x_2+k}$ . But there is no such  $F$ -node. And if  $h(t_{last}) = f_1^{x_2+k}$  for some  $1 \leq k \leq |q|$ , then both  $h(t_{last-1})$  and  $h(f_1)$  are contacts in  $\mathcal{W}_2$ , contradicting (25).

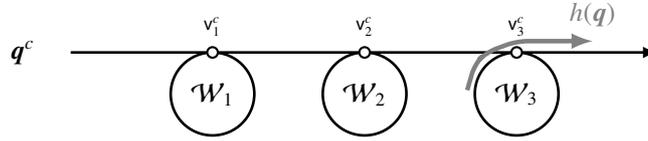


*Case  $f_1 < t_{last}$ .* Then  $v_2 = f_1$ , and so  $f_1^c \leq_{h(q)} h(f_1)$  and  $h(f_1)$  is in  $\mathcal{W}_2$ . As  $f^{x_2} = f_2^{x_2}$  and  $f^{x_2+\ell} = f_1^{x_2+\ell}$  for  $1 \leq \ell \leq |q|$ ,  $h(f_1)$  cannot be in the final cog of  $q^{x_2+\ell}$  for any  $\ell \leq |q|$  (otherwise there is not enough room for  $h(q)$  in  $q^{x_2+\ell}$ ). As  $t^{x_2+\ell} = t_{\square}^{x_2+\ell}$  for  $1 \leq \ell \leq |q|$ , if  $h(f_1)$  is a contact in  $\mathcal{W}_2$  different from  $v_2^c$ , then  $h(t_{\square})$  is a contact in  $\mathcal{W}_2$  as well, contradicting (25). As there is no  $F$ -node between  $t_{\square}^{x_2+\ell}$  and  $f_1^{x_2+\ell}$  for any  $\ell$ , it follows that  $h(f_1) = v_2^c$ . We track the location of  $h(t_{\Delta})$  for the first  $T$ -node  $t_{\Delta}$  succeeding  $f_1$  in  $q$ . As  $f^{x_2} = f_2^{x_2}$ ,  $h(t_{\Delta})$  cannot be in the final cog of  $q^{x_2}$  (otherwise there is not enough room for  $h(q)$  in  $q^{x_2}$ ). Further,  $h(t_{\Delta})$  cannot be a contact in  $\mathcal{W}_2$  by (25), and so it must be in the final cog of  $q^{x_2+k}$  for some  $1 \leq k \leq |q|$  (as there is no  $T$ -node between  $t_{\square}^{x_2+\ell}$  and  $f_1^{x_2+\ell}$  for any  $\ell$ ). But then

$$\delta_{q^{x_2+k}}(f_1^{x_2+k}, t_{\Delta}^{x_2+k}) = \delta(f_1, t_{\Delta}) = \delta_{h(q)}(h(f_1), h(t_{\Delta})) > \delta_{q^{x_2+k}}(f_1^{x_2+k}, h(t_{\Delta})),$$

and so  $h(t_{\Delta})$  cannot be a  $T$ -node in  $q^{x_2+k}$ .

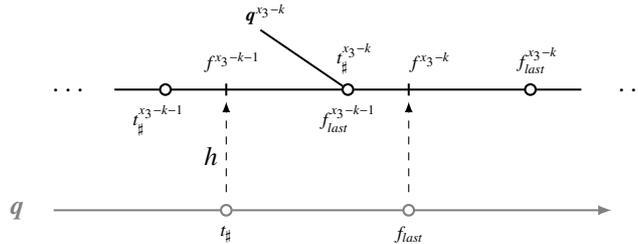
(5)  $h(q)$  starts in  $\mathcal{W}_3$ .



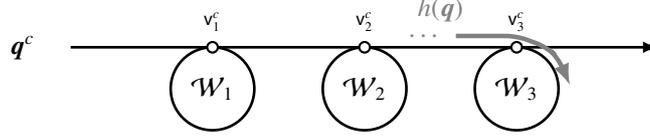
Then  $h(v_3) \leq_{h(q)} v_3^c$ , as otherwise there is no room for  $h(q)$  in  $q^c$ . As  $v_3 = f_{last}$ , we have  $h(f_{last}) \leq_{h(q)} f_{last}^c$  and  $h(f_{last})$  is in  $\mathcal{W}_3$ . As  $t^{x_3+1} = t_{\#}^{x_3+1}$  and  $t^{x_3-k} = t_{\#}^{x_3-k}$  for  $k \leq |q|$ ,  $h(f_{last})$  cannot be in the initial cog of either  $q^{x_3+1}$  or  $q^{x_3-k}$  for any  $k \leq |q|$  (otherwise there is not enough room for  $h(q)$  in that cog). As  $f^{x_3-k} = f_{last}^{x_3-k}$  for  $k \leq |q|$ , if  $h(f_{last})$  is a contact in  $\mathcal{W}_3$ , then  $h(t_{\#})$  is a contact in  $\mathcal{W}_3$  as well, contradicting (25). So suppose that  $h(f_{last}) = f_{last}^{x_3-k}$  for some  $k \leq |q|$  and some  $F$ -node  $f$  with  $t_{\#} < f < f_{last}$ . We track the location of  $h(t_{\#})$ . Clearly,  $h(t_{\#})$  cannot be in the initial cog of  $q^{x_3-k}$ , as otherwise there is not enough room for  $h(q)$  in  $q^{x_3-k}$ . Thus,  $h(t_{\#})$  is in  $q^{x_3-k-1}$  and

$$\delta_{h(q)}(h(t_{\#}), f_{last}^{x_3-k-1}) = \delta_{h(q)}(h(f_{last}), f_{last}^{x_3-k}) = \delta_{q^{x_3-k}}(f^{x_3-k}, f_{last}^{x_3-k}) = \delta(f, f_{last}).$$

Then  $h(t_{\#}) = f^{x_3-k-1}$ , and so it is not in  $T^{\mathcal{I}^a}$ .



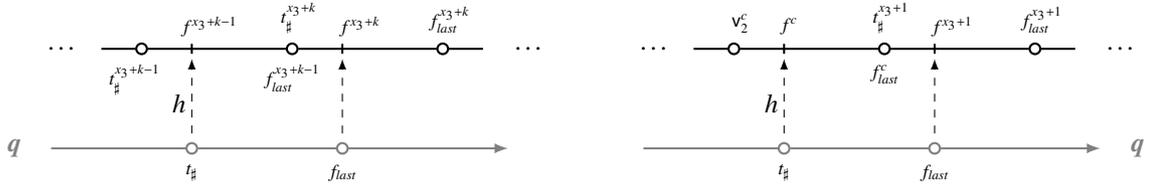
(6)  $h(q)$  ends in  $\mathcal{W}_3$  and  $v_3^c <_{h(q)} h(v_3)$ .



As  $v_3 = f_{last}$ , we have that  $f_{last}^c <_{h(q)} h(f_{last})$  and  $h(f_{last})$  is in  $\mathcal{W}_3$ . As  $f^{x_3} = f_{last}^{x_3}$  and  $f^{x_3+\ell} = f_{last}^{x_3+\ell}$  for  $1 \leq \ell \leq |q|$ ,  $h(f_{last})$  cannot be in the final cog of  $q^{x_3}$  or  $q^{x_3+\ell}$  for any  $\ell$  (otherwise there is not enough room for  $h(q)$  in that cog). As  $t^{x_3+\ell} = t_{\#}^{x_3+\ell}$  for  $1 \leq \ell \leq |q|$ ,  $h(f_{last})$  cannot be a contact in  $\mathcal{W}_3$ , otherwise  $h(t_{\#})$  is also a contact in  $\mathcal{W}_3$ , contradicting (25). So  $h(f_{last}) = f^{x_3+k}$  must hold for some  $1 \leq k \leq |q|$  and some  $F$ -node  $f$  with  $t_{\#} < f < f_{last}$ . We track the location of  $h(t_{\#})$ . Let  $i = c$  if  $k = 1$ , and  $i = x_3 + k - 1$  otherwise. Then

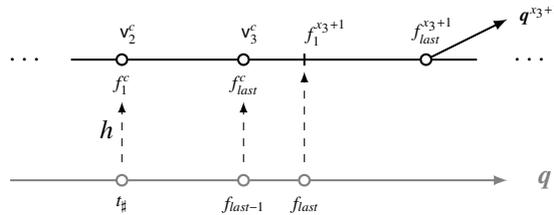
$$\delta_{h(q)}(h(t_{\#}), f_{last}^i) = \delta_{h(q)}(h(f_{last}), f_{last}^{x_3+k}) = \delta_{q^{x_3+k}}(f^{x_3+k}, f_{last}^{x_3+k}) = \delta(f, f_{last}).$$

If  $k > 1$  then  $h(t_{\#}) = f^{x_3+k-1}$ , and so it is not in  $T^{\mathcal{I}_a}$ . If  $k = 1$  then there are two cases: either  $v_2^c <_{q^c} f^c$  or  $f^c \leq_{q^c} v_2^c$ . If  $v_2^c <_{q^c} f^c$  then  $h(t_{\#}) = f^c$ , and so it is not in  $T^{\mathcal{I}_a}$ .



So suppose  $f_1^c \leq_{q^c} f^c \leq_{q^c} v_2^c$ . Then  $v_2 = f = f_1$  and  $f_1 < t_{last}$ , and so  $h(t_{\#}) = v_2^c = f^c = f_1^c$ . as  $t_{\#} < f = f_1$ , the number of  $F$ -nodes in  $q^c$  between  $h(t_{\#}) = f_1^c$  and  $v_3^c = f_{last}^c$  is smaller than the number of  $F$ -nodes between  $t_{\#}$  and  $f_{last}$  in  $q$ . Therefore, as there are no  $F$ -nodes between  $v_3^c$  and  $f_1^{x_3+1} = h(f_{last})$ , we must have

$$h(f_{last-1}) = v_3^c \text{ is a contact in } \mathcal{W}_3. \quad (31)$$



On the other hand, as  $t_{\#} < f = f_1$ , it follows that there is no  $T$ -node in  $q$  between  $f_1$  and  $f_{last}$ , and so  $f_{last} < t_{last}$ . We track the location of  $h(t_{\#})$  for the first  $T$ -node  $t_{\blacktriangle}$  succeeding  $f_{last}$  in  $q$ . It cannot be between  $t_{\#}^{x_3+\ell}$  and  $f_{last}^{x_3+\ell}$ , for some  $\ell$ , as there is no  $T$ -node there for any  $\ell$ . If  $h(t_{\#})$  is a contact in  $\mathcal{W}_3$ , then this contradicts (25) and (31). Finally, if  $h(t_{\#})$  is in the final cog of  $q^{x_3+\ell}$ , for some  $1 \leq \ell \leq |q|$ , then

$$\delta_{q^{x_3+1}}(f_{last}^{x_3+1}, h(t_{\#})) < \delta_{q^{x_3+1}}(f_1^{x_3+1}, h(t_{\#})) = \delta_{q^{x_3+1}}(h(f_{last}), h(t_{\#})) = \delta(f_{last}, t_{\blacktriangle}) = \delta_{q^{x_3+1}}(f_{last}^{x_3+1}, t_{\blacktriangle}^{x_3+1}).$$

But there is no  $T$ -node between  $f_{last}^{x_3+1}$  and  $t_{\blacktriangle}^{x_3+1}$ .

This completes the proof of Lemma 21.5.  $\square$

Finally, given a 3CNF  $\psi$ , take some  $\psi$ -gadget  $\mathcal{A}_{\psi}$ . By Lemmas 21.4 and 21.5, we have  $\text{cov}_{\mathcal{A}} \mathcal{A}_{\psi} \neq q$  iff  $\psi$  is satisfiable. This completes the proof of Theorem 21.

## 7. Conclusion and Outlook

This article contributes to the growing area of research into the non-uniform complexity of OMQ answering (which has been given a strong impetus by the recent breakthrough results in non-uniform CSPs). Although there exist algorithms that are capable of recognising the data complexity and rewritability type of OMQs given in different DLs (such as  $\mathcal{EL}$ ,  $\mathcal{ALC}$  or  $\mathcal{SHIU}$ ), monadic datalog or disjunctive datalog, they are of so high complexity that complete syntactic and practical general classifications of OMQs seem hardly possible.

Here, we take a different, bottom-up route to understanding the data complexity and rewritability of non-Horn OMQs. Namely, we fix the basic ontology  $\text{cov}_A$  with a single axiom saying that class  $A$  is covered by the union of two classes  $F$  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 and/or necessary conditions for membership in the complexity classes  $\text{AC}^0, \text{L}, \text{NL}, \text{P}$ . The main result of the article is a remarkably transparent and tractable  $\text{AC}^0/\text{NL}/\text{P}/\text{coNP}$  tetrachotomy (requiring a pretty complex proof) for OMQs with the ontology  $\text{cov}_A^\perp$  (making  $T$  and  $F$  disjoint) and a path CQ.

These positive results can hopefully be generalised to more expressive non-Horn ontologies in Schema.org and (fragments of) the description logics  $DL\text{-Lite}_{krom}$  and  $DL\text{-Lite}_{bool}$  in the  $DL\text{-Lite}$  family [57], thereby extending the classical OBDA paradigm to queries mediated by non-Horn ontologies. Note that  $DL\text{-Lite}_{krom}$  only allows binary clauses as ontology axioms, and so only total covering  $\top \sqsubseteq F \sqcup T$  (that is,  $\forall x (F(x) \vee T(x))$ ). Compared to  $\text{cov}_A$ , OMQs with the ontology  $\text{cov}_\top$  are in general less complex (see Example 15), so it would be interesting and challenging to extend our tetrachotomy in Theorem 17 for path CQs to the case of total covering. Generalising it to OMQs with tree-shaped CQs is another promising direction. Yet another direction of extending our tetrachotomy is to consider path CQs *with*  $FT$ -twins. While it is easy to extend the proof of Theorem 18 to show NL-hardness of these, generalising other results might turn out to be more challenging.

For OMQs with a dag-shaped (let alone arbitrary) CQ, finding complete syntactic classifications could be much harder: as we show in Theorem 14, deciding FO-rewritability of such OMQs (even with a 1-CQ) turns out to be  $\text{PSPACE}$ -hard. We are working on pinpointing the exact complexity of classifying arbitrary OMQs with covering according to their data complexity and rewritability type. In particular, we believe that deciding FO-rewritability is actually  $2\text{EXPTIME}$ -complete for OMQs with a 1-CQ, and  $2\text{NEXPTIME}$ -complete for OMQs with arbitrary CQs, which matches the complexity of deciding boundedness of monadic datalog and, respectively, disjunctive datalog queries. The novel technique developed in the proof of Theorem 14 paves the way to identifying the exact complexity of deciding boundedness of monadic single rule (disjunctive) datalog programs, which seems to have been an open problem since the 1980s. We will address this problem in a forthcoming publication.

## Acknowledgements

The work of O. Gerasimova was supported by the Russian Academic Excellence Project 5-100. The work of V. Podolskii was partially supported by the RFBR grant 18-01-00822. The work of M. Zakharyashev was supported by the EPSRC U.K. grant EP/S032282.

## References

- [1] A. Schaerf, On the complexity of the instance checking problem in concept languages with existential quantification, *J. of Intelligent Information Systems* 2 (1993) 265–278.
- [2] F. Baader, D. Calvanese, D. L. McGuinness, D. Nardi, P. F. Patel-Schneider (Eds.), *The Description Logic Handbook*, 2 ed., Cambridge University Press, Cambridge, UK, 2007.
- [3] F. Baader, I. Horrocks, C. Lutz, U. Sattler, *An Introduction to Description Logic*, Cambridge University Press, 2017.
- [4] A. Poggi, D. Lembo, D. Calvanese, G. De Giacomo, M. Lenzerini, R. Rosati, Linking data to ontologies, *Journal on Data Semantics X* (2008) 133–173.
- [5] D. Calvanese, G. De Giacomo, D. Lembo, M. Lenzerini, R. Rosati, Tractable reasoning and efficient query answering in description logics: the  $DL\text{-Lite}$  family, *Journal of Automated Reasoning* 39 (2007) 385–429.
- [6] G. Xiao, D. Calvanese, R. Kontchakov, D. Lembo, A. Poggi, R. Rosati, M. Zakharyashev, Ontology-based data access: A survey, in: J. Lang (Ed.), *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI 2018, July 13-19, 2018, Stockholm, Sweden.*, ijcai.org, 2018, pp. 5511–5519.
- [7] G. Xiao, L. Ding, B. Cogrel, D. Calvanese, Virtual knowledge graphs: An overview of systems and use cases, *Data Intell.* 1 (2019) 201–223.

- [8] S. Abiteboul, R. Hull, V. Vianu, *Foundations of Databases*, Addison-Wesley, 1995.
- [9] T. Eiter, M. Ortiz, M. Šimkus, T. Tran, G. Xiao, Query rewriting for Horn-*SHIQ* plus rules, in: J. Hoffmann, B. Selman (Eds.), *Proceedings of the Twenty-Sixth AAAI Conference on Artificial Intelligence*, July 22-26, 2012, Toronto, Ontario, Canada, AAAI Press, 2012.
- [10] U. Hustadt, B. Motik, U. Sattler, Data complexity of reasoning in very expressive description logics, in: L. P. Kaelbling, A. Saffiotti (Eds.), *IJCAI-05, Proceedings of the Nineteenth International Joint Conference on Artificial Intelligence*, Edinburgh, Scotland, UK, July 30 - August 5, 2005, Professional Book Center, 2005, pp. 466–471.
- [11] R. Rosati, On conjunctive query answering in EL, in: D. Calvanese, E. Franconi, V. Haarslev, D. Lembo, B. Motik, A. Turhan, S. Tessaris (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, 2007.
- [12] D. Gabbay, A. Kurucz, F. Wolter, M. Zakharyashev, Many-Dimensional Modal Logics: Theory and Applications, volume 148 of *Studies in Logic and the Foundations of Mathematics*, Elsevier, 2003.
- [13] D. Carral, C. Feier, B. C. Grau, P. Hitzler, I. Horrocks, *EL*-ifying ontologies, in: S. Demri, D. Kapur, C. Weidenbach (Eds.), *Automated Reasoning - 7th International Joint Conference, IJCAR 2014, Held as Part of the Vienna Summer of Logic, VSL 2014, Vienna, Austria, July 19-22, 2014*. *Proceedings*, volume 8562 of *Lecture Notes in Computer Science*, Springer, 2014, pp. 464–479.
- [14] Y. Zhou, B. C. Grau, Y. Nenov, M. Kaminski, I. Horrocks, Pagoda: Pay-as-you-go ontology query answering using a datalog reasoner, *J. Artif. Intell. Res.* 54 (2015) 309–367.
- [15] E. Botoeva, D. Calvanese, V. Santarelli, D. F. Savo, A. Solimando, G. Xiao, Beyond OWL 2 QL in OBDA: rewritings and approximations, in: D. Schuurmans, M. P. Wellman (Eds.), *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*, February 12-17, 2016, Phoenix, Arizona, USA, AAAI Press, 2016, pp. 921–928.
- [16] A. Bötcher, C. Lutz, F. Wolter, Ontology approximation in Horn description logics, in: S. Kraus (Ed.), *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI 2019, Macao, China, August 10-16, 2019*, ijcai.org, 2019, pp. 1574–1580.
- [17] D. Hovland, R. Kontchakov, M. G. Skjæveland, A. Waaler, M. Zakharyashev, Ontology-based data access to Slegge, in: C. d’Amato, M. Fernández, V. A. M. Tamma, F. Lécué, P. Cudré-Mauroux, J. F. Sequeda, C. Lange, J. Heflin (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*, Springer, 2017, pp. 120–129.
- [18] E. Kharlamov, D. Hovland, M. G. Skjæveland, D. Bilidas, E. Jiménez-Ruiz, G. Xiao, A. Soylu, D. Lanti, M. Rezk, D. Zheleznyakov, M. Giese, H. Lie, Y. E. Ioannidis, Y. Kotidis, M. Koubarakis, A. Waaler, Ontology based data access in Statoil, *J. Web Sem.* 44 (2017) 3–36.
- [19] C. Lutz, F. Wolter, Non-uniform data complexity of query answering in description logics, in: G. Brewka, T. Eiter, S. A. McIlraith (Eds.), *Principles of Knowledge Representation and Reasoning: Proceedings of the Thirteenth International Conference, KR 2012, Rome, Italy, June 10-14, 2012*, AAAI Press, 2012.
- [20] M. Bienvenu, B. ten Cate, C. Lutz, F. Wolter, Ontology-based data access: A study through disjunctive datalog, CSP, and MMSNP, *ACM Transactions on Database Systems* 39 (2014) 33:1–44.
- [21] C. Lutz, I. Seylan, F. Wolter, The data complexity of ontology-mediated queries with closed predicates, *Log. Methods Comput. Sci.* 15 (2019).
- [22] A. Hernich, C. Lutz, F. Papacchini, F. Wolter, Dichotomies in ontology-mediated querying with the guarded fragment, *ACM Trans. Comput. Log.* 21 (2020) 20:1–20:47.
- [23] T. Feder, M. Y. Vardi, The computational structure of monotone monadic SNP and constraint satisfaction: A study through datalog and group theory, *SIAM J. Comput.* 28 (1998) 57–104.
- [24] A. A. Bulatov, A dichotomy theorem for nonuniform CSPs, in: C. Umans (Ed.), *58th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2017, Berkeley, CA, USA, October 15-17, 2017*, IEEE Computer Society, 2017, pp. 319–330.
- [25] D. Zhuk, A proof of CSP dichotomy conjecture, in: C. Umans (Ed.), *58th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2017, Berkeley, CA, USA, October 15-17, 2017*, IEEE Computer Society, 2017, pp. 331–342.
- [26] P. Bourhis, C. Lutz, Containment in monadic disjunctive datalog, MMSNP, and expressive description logics, in: C. Baral, J. P. Delgrande, F. Wolter (Eds.), *Principles of Knowledge Representation and Reasoning: Proceedings of the Fifteenth International Conference, KR 2016, Cape Town, South Africa, April 25-29, 2016*, AAAI Press, 2016, pp. 207–216.
- [27] C. Feier, A. Kuusisto, C. Lutz, Rewritability in monadic disjunctive datalog, MMSNP, and expressive description logics, *Logical Methods in Computer Science* 15 (2019).
- [28] C. Lutz, L. Sabellek, Ontology-mediated querying with the description logic EL: trichotomy and linear datalog rewritability, in: C. Sierra (Ed.), *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI 2017, Melbourne, Australia, August 19-25, 2017*, ijcai.org, 2017, pp. 1181–1187.
- [29] M. Y. Vardi, Decidability and undecidability results for boundedness of linear recursive queries, in: C. Edmondson-Yurkanan, M. Yannakakis (Eds.), *Proceedings of the Seventh ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems*, March 21-23, 1988, Austin, Texas, USA, ACM, 1988, pp. 341–351.
- [30] J. Marcinkowski, 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, IEEE Computer Society, 1996, pp. 13–24.
- [31] J. Marcinkowski, Achilles, turtle, and undecidable boundedness problems for small DATALOG programs, *SIAM J. Comput.* 29 (1999) 231–257.
- [32] S. S. Cosmadakis, H. Gaifman, P. C. Kanellakis, M. Y. Vardi, Decidable optimization problems for database logic programs (preliminary report), in: *STOC*, 1988, pp. 477–490.
- [33] M. Benedikt, B. ten Cate, T. Colcombet, M. Vanden Boom, 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*, IEEE Computer Society, 2015, pp. 293–304.
- [34] J. D. Ullman, A. V. Gelder, Parallel complexity of logical query programs, *Algorithmica* 3 (1988) 5–42.
- [35] R. Ramakrishnan, Y. Sagiv, J. D. Ullman, M. Y. Vardi, Proof-tree transformation theorems and their applications, in: *Proceedings of the eighth ACM SIGACT-SIGMOD-SIGART symposium on Principles of database systems*, ACM, 1989, pp. 172–181.
- [36] Y. P. Saraiya, Linearizing nonlinear recursions in polynomial time, in: A. Silberschatz (Ed.), *Proceedings of the Eighth ACM SIGACT-*

- SIGMOD-SIGART Symposium on Principles of Database Systems, March 29-31, 1989, Philadelphia, Pennsylvania, USA, ACM Press, 1989, pp. 182–189.
- [37] W. Zhang, C. T. Yu, D. Troy, Necessary and sufficient conditions to linearize double recursive programs in logic databases, *ACM Trans. Database Syst.* 15 (1990) 459–482.
- [38] F. N. Afrati, M. Gergatsoulis, F. Toni, Linearisability on datalog programs, *Theor. Comput. Sci.* 308 (2003) 199–226.
- [39] M. Kaminski, Y. Nenov, B. C. Grau, Datalog rewritability of disjunctive datalog programs and non-Horn ontologies, *Artif. Intell.* 236 (2016) 90–118.
- [40] O. Gerasimova, S. Kikot, M. Zakharyashev, Checking the data complexity of ontology-mediated queries: A case study with non-uniform CSPs and Polyanna, in: C. Lutz, U. Sattler, C. Tinelli, A. Turhan, F. Wolter (Eds.), *Description Logic, Theory Combination, and All That - Essays Dedicated to Franz Baader on the Occasion of His 60th Birthday*, volume 11560 of *Lecture Notes in Computer Science*, Springer, 2019, pp. 329–351.
- [41] R. Gault, P. Jeavons, Implementing a test for tractability, *Constraints* 9 (2004) 139–160.
- [42] R. Elmasri, S. B. Navathe, *Fundamentals of Database Systems*, 7th ed., Pearson, 2015.
- [43] P. F. Patel-Schneider, Analyzing schema.org, in: P. Mika, T. Tudorache, A. Bernstein, C. Welty, C. A. Knoblock, D. Vrandečić, P. T. Groth, N. F. Noy, K. Janowicz, C. A. Goble (Eds.), *The Semantic Web - ISWC 2014 - 13th International Semantic Web Conference*, Riva del Garda, Italy, October 19-23, 2014. Proceedings, Part I, volume 8796 of *Lecture Notes in Computer Science*, Springer, 2014, pp. 261–276.
- [44] A. Hernich, C. Lutz, A. Ozaki, F. Wolter, Schema.org as a description logic, in: Q. Yang, M. J. Wooldridge (Eds.), *Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence, IJCAI 2015*, Buenos Aires, Argentina, July 25-31, 2015, AAAI Press, 2015, pp. 3048–3054.
- [45] J. D. Ullman, *Principles of Database and Knowledge-Base Systems*, Volume II, Computer Science Press, 1989.
- [46] C.-L. Chang, R. C.-T. Lee, *Symbolic Logic and Mechanical Theorem Proving*, 1st ed., Academic Press, Inc., USA, 1973.
- [47] N. Immerman, *Descriptive Complexity*, Springer, 1999.
- [48] E. Dantsin, T. Eiter, G. Gottlob, A. Voronkov, Complexity and expressive power of logic programming, *ACM Computing Surveys* 33 (2001) 374–425.
- [49] L. J. Stockmeyer, The polynomial-time hierarchy, *Theor. Comput. Sci.* 3 (1976) 1–22.
- [50] L. Egri, B. Larose, P. Tesson, Symmetric datalog and constraint satisfaction problems in logspace, in: *Logic in Computer Science, 2007. LICS 2007. 22nd Annual IEEE Symposium on*, IEEE, 2007, pp. 193–202.
- [51] S. S. Cosmadakis, P. C. Kanellakis, Parallel evaluation of recursive rule queries, in: A. Silberschatz (Ed.), *Proceedings of the Fifth ACM SIGACT-SIGMOD Symposium on Principles of Database Systems*, March 24-26, 1986, Cambridge, Massachusetts, USA, ACM, 1986, pp. 280–293.
- [52] J. F. Naughton, Data independent recursion in deductive databases, in: A. Silberschatz (Ed.), *Proceedings of the Fifth ACM SIGACT-SIGMOD Symposium on Principles of Database Systems*, March 24-26, 1986, Cambridge, Massachusetts, USA, ACM, 1986, pp. 267–279.
- [53] C. Lutz, L. Sabellek, A complete classification of the complexity and rewritability of ontology-mediated queries based on the description logic EL, *CoRR* abs/1904.12533 (2019). URL: <http://arxiv.org/abs/1904.12533>. arXiv:1904.12533.
- [54] C. H. Papadimitriou, M. Yannakakis, A note on succinct representations of graphs, *Information and Control* 71 (1986) 181–185.
- [55] S. Arora, B. Barak, *Computational Complexity: A Modern Approach*, 1st ed., Cambridge University Press, New York, NY, USA, 2009.
- [56] H. Comon, M. Dauchet, R. Gilleron, C. Löding, F. Jacquemard, D. Lugiez, S. Tison, M. Tommasi, *Tree automata techniques and applications*, Available on: <http://www.grappa.univ-lille3.fr/tata>, 2007.
- [57] A. Artale, D. Calvanese, R. Kontchakov, M. Zakharyashev, The DL-Lite family and relations, *Journal of Artificial Intelligence Research (JAIR)* 36 (2009) 1–69.