Journal of Artificial Intelligence Research 76 (2023) 645-703 Submitted 07/2022; published 03/2023

Deciding FO-rewritability of Regular Languages and Ontology-mediated Queries in Linear Temporal Logic

Agi Kurucz AGI.KURUCZ@KCL.AC.UK
King's College London, UK

Vladislav Ryzhikov VLAD@DCS.BBK.AC.UK
Birkbeck, University of London, UK

Yury Savateev Y.SAVATEEV@SOTON.AC.UK
University of Southampton, UK

Michael Zakharyaschev MICHAEL@DCS.BBK.AC.UK
Birkbeck, University of London, UK

Abstract

Our concern is the problem of determining the data complexity of answering an ontology-mediated query (OMQ) formulated in linear temporal logic *LTL* over $(\mathbb{Z}, <)$ and deciding whether it is rewritable to an $\mathsf{FO}(<)$-query, possibly with some extra predicates. First, we observe that, in line with the circuit complexity and FO-definability of regular languages, OMQ answering in AC^0, ACC^0 and NC^1 coincides with $\mathsf{FO}(<, \equiv)$-rewritability using unary predicates $x \equiv 0 \,(\mathrm{mod}\ n)$, $\mathsf{FO}(<, \mathsf{MOD})$-rewritability, and $\mathsf{FO}(\mathsf{RPR})$-rewritability using relational primitive recursion, respectively. We prove that, similarly to known PSPACE-completeness of recognising $\mathsf{FO}(<)$-definability of regular languages, deciding $\mathsf{FO}(<, \equiv)$- and $\mathsf{FO}(<, \mathsf{MOD})$-definability is also PSPACE-complete (unless $\mathrm{ACC}^0 = \mathrm{NC}^1$). We then use this result to show that deciding $\mathsf{FO}(<)$-, $\mathsf{FO}(<, \equiv)$- and $\mathsf{FO}(<, \mathsf{MOD})$-rewritability of *LTL* OMQs is EXPSPACE-complete, and that these problems become PSPACE-complete for OMQs with a linear Horn ontology and an atomic query, and also a positive query in the cases of $\mathsf{FO}(<)$- and $\mathsf{FO}(<, \equiv)$-rewritability. Further, we consider $\mathsf{FO}(<)$-rewritability of OMQs with a binary-clause ontology and identify OMQ classes, for which deciding it is PSPACE-, Π^p_2- and coNP-complete.

1. Introduction

1.1 Motivation

The problem we consider in this article originates in the area of *ontology-based data access* (*OBDA*) to temporal data. The aim of the OBDA paradigm (Poggi et al., 2008; Xiao et al., 2018) and OBDA systems such as Mastro (`https://www.obdasystems.com`) or Ontop (`https://ontopic.biz`) is to facilitate management and integration of possibly incomplete and heterogeneous data by providing the user with a view of the data through the lens of a description logic (DL) ontology. As a result, the user can think of the data as a virtual knowledge graph (Xiao et al., 2019), $\mathcal{A}$, whose labels—unary and binary predicates supplied by an ontology, $\mathcal{O}$—are the only thing to know when formulating queries, $\varkappa$. Ontology-mediated queries (OMQs) $\boldsymbol{q} = (\mathcal{O}, \varkappa)$ are supposed to be answered over $\mathcal{A}$ under the open-world semantics (taking account of all models of $\mathcal{O}$ and $\mathcal{A}$), which can be prohibitively complex. So the key to practical OBDA is ensuring first-order rewritability of $\boldsymbol{q}$ (aka *boundedness* in the datalog literature (Abiteboul et al., 1995)), which reduces open-world reasoning to evaluating an FO-formula over $\mathcal{A}$. The W3C standard ontology language *OWL 2 QL* for OBDA is based on the *DL-Lite* family of DL (Calvanese et al., 2007; Artale et al., 2009), which uniformly guarantees FO-rewritability of all *OWL 2 QL* OMQs with a conjunctive query. Other ontology languages with this feature include various dialects of tgds (e.g., Baget et al., 2011; Calì et al., 2012; Civili & Rosati, 2012). However, this *uniform* approach to ensuring FO-rewritability inevitably imposes severe syntactical restrictions on ontology languages, making them rather inexpressive.

Theory and practice of OBDA have revived the interest in the *non-uniform* approach, where the problem is to decide whether a given OMQ, formulated in some expressive language, is FO-rewritable. This problem was thoroughly investigated in the 1980–90s for datalog queries (e.g., Vardi, 1988; Ullman & Gelder, 1988; Cosmadakis et al., 1988; Afrati & Papadimitriou, 1993; Marcinkowski, 1996). The data complexity and rewritability of OMQs in various DLs and disjunctive datalog have become an active research area in the past decade (Bienvenu et al., 2014; Kaminski et al., 2016; Lutz & Sabellek, 2017; Feier et al., 2019; Gerasimova et al., 2020) lying at the crossroads of logic, database theory, knowledge representation in AI, circuit and descriptive complexity, and constraint satisfaction problems.

There have been numerous attempts to extend ontology and query languages with constructors that are capable of representing events over *temporal data*; consult Lutz et al. (2008), Artale et al. (2017) for surveys and Gutiérrez-Basulto and Jung (2017), Borgwardt et al. (2019), Walega et al. (2020b, 2020a), Artale et al. (2022) for more recent developments. However, so far the focus has only been on the uniform complexity of reasoning with arbitrary ontologies and queries in a given language rather than on determining the data complexity and FO-rewritability of individual temporal OMQs. On the other hand, standard temporal logics are interpreted over linearly-ordered structures, and so the non-uniform analysis of OMQs in DLs and datalog mentioned above is not applicable to them.

In this paper, we take a first step towards understanding the problem of non-uniform FO-rewritability of OMQs over temporal data by focusing on the temporal dimension and considering OMQs given in linear temporal logic *LTL* interpreted over $(\mathbb{Z}, <)$. In fact, already this basic 'one-dimensional' temporal OBDA formalism provides enough expressive power in those real-world situations where the interaction among individuals in the object domain is not important and can be disregarded in data modelling. (This interaction is usually captured by binary relations (roles) in DLs, giving the models a 'two-dimensional' character.) We illustrate this claim and the language of *LTL* OMQs by an example.

Example 1. A typical scenario for the use of OBDA technologies is where a non-IT-expert user, say a turbine engineer, analyses the behaviour of a complex system, turbines in our example, based on various sensor measurements stored in a relational database. To be more specific, imagine that turbines, t, are equipped with sensors, s, to measure such parameters as the rotor speed, the temperature of the blades, vibration, active power, etc. The relational database in a remote diagnostic centre might store a binary predicate $location(s, t)$ saying that sensor s is located in turbine t and a ternary predicate $measurement(s, v, n)$ giving the numerical value v of the reading of s at time instant n. The timestamps of sensor readings are synchronised with a central clock, and so can be regarded as integers.

When defining events of interest like 'active power trip' or 'purging is over', engineers usually operate with facts such as 'the active power of turbine t measured by s is above 1.5MW at moment n', which can be obtained as database views of the form $\mathit{ActivePower}^{t,s}_{\geq 1.5}(n)$. We regard these unary predicates as atomic concepts that can be true or false at different moments of time. Omitting t and s to unclutter notation, we can then assume that our virtual database $\mathcal{A}$ consists of facts like

$$\mathit{Run}(6),\ \mathit{ActivePower}_{\geq 1.5}(7),\ \mathit{Malfunction}(7),\ \mathit{Disabled}(10), \tag{1}$$

based on which we analyse the behaviour of the turbines. As some sensors might occasionally fail to send their measurements, we cannot assume the data to be complete. Thus, in our example data above, the sensor detecting if the turbine is running (by measuring the electric current) failed to send a signal at time instant 7. However, the power sensor attached to the turbine recorded $\geq$ 1.5MW at 7, which should imply that the turbine was running at 7. This piece of domain knowledge can be encoded by the ontology axiom

$$\Box_F \Box_P (\mathit{ActivePower}_{\geq 1.5} \to \mathit{Run}) \tag{2}$$

with the *LTL*-operators $\Box_F$ (always in the future) and $\Box_P$ (always in the past). Other *LTL* axioms in our example ontology $\mathcal{O}$ (designed by a domain expert) could look like

$$\Box_F \Box_P (\mathit{Pause} \land \mathit{Run} \to \bot), \tag{3}$$
$$\Box_F \Box_P (\mathit{Malfunction} \to \bigcirc_F \mathit{Pause}), \tag{4}$$
$$\Box_F \Box_P (\mathit{Malfunction} \to \Diamond_F \mathit{Diagnostics}), \tag{5}$$
$$\Box_F \Box_P (\mathit{Disabled} \to \neg \Diamond_F \mathit{Diagnostics}). \tag{6}$$

The first of them says that a turbine cannot be paused and running at the same time; the second and third say that immediately after ($\bigcirc_F$) a malfunction, the turbine is paused and will eventually ($\Diamond_F$) be diagnosed; the fourth axiom asserts that a disabled turbine will never undergo diagnostics in the future.

Now, if we are interested in continuous runs lasting at least two time units that end up in a non-run state, we (engineers) could write and execute the following simple query $\varkappa(x)$ with the previous-time operator $\bigcirc_P$, assuming that $\varkappa(x)$ is mediated by the ontology $\mathcal{O}$:

$$\varkappa(x) \ = \ \neg \mathit{Run} \land \bigcirc_P \mathit{Run} \land \bigcirc_P \bigcirc_P \mathit{Run}.$$

Intuitively, we are looking for those timestamps x in the active domain of the database at which this temporal formula is a logical consequence of $\mathcal{O}$ and the data. It is not hard to see that the only certain answer to the OMQ $(\mathcal{O}, \varkappa(x))$ over $\mathcal{A}$ given by (1) is the time instant 8 because we can derive $\neg \mathit{Run}(x)$ if $\mathit{Pause}(x)$ or $\mathit{Malfunction}(x-1)$ is in $\mathcal{A}$, or $\mathcal{O}$ and $\mathcal{A}$ are inconsistent; and we know for certain that $\mathit{Run}(x)$ iff $\mathcal{A}$ contains $\mathit{Run}(x)$ or $\mathit{ActivePower}_{\geq 1.5}(x)$, or again $\mathcal{O}$ and $\mathcal{A}$ are inconsistent. These conditions can be expressed by the $\mathsf{FO}(<)$-query $\boldsymbol{Q}(x) = \varphi(x) \lor \mathit{Incons}$, to be evaluated over $\mathcal{A}$, where

$$\varphi(x) = (\mathit{Pause}(x) \lor \mathit{Malfunction}(x-1)) \land (\mathit{Run}(x-1) \lor \mathit{ActivePower}_{\geq 1.5}(x-1)) \land \\ (\mathit{Run}(x-2) \lor \mathit{ActivePower}_{\geq 1.5}(x-2)).$$

and *Incons* is a disjunction of a few sentences such as

$$\exists x\,(\mathit{Malfunction}(x) \wedge \mathit{ActivePower}_{\geq 1.5}(x+1)),$$
$$\exists x, y\,((y \geq x) \wedge \mathit{Disabled}(x) \wedge \mathit{Malfunction}(y)), \quad \dots$$

that describe all of the cases when $\mathcal{O}$ is inconsistent with $\mathcal{A}$ (which are left to the reader). The aim of a temporal OBDA system is to construct such an $\mathsf{FO}(<)$-rewriting $\boldsymbol{Q}(x)$ of the OMQ $(\mathcal{O}, \varkappa(x))$ automatically, and evaluate it over the original relational data using a conventional database management system. The OMQ $(\mathcal{O}, \varkappa'(x))$ with

$$\varkappa'(x) = \varkappa(x) \ \wedge \ (\mathit{Diagnostics} \vee \bigcirc_{\!F} \mathit{Diagnostics} \vee \bigcirc_{\!F}\bigcirc_{\!F} \mathit{Diagnostics})$$

also returns 8 over $\mathcal{A}$ because (5) and (6) imply that diagnostics took place some time in the interval $[8, 10]$. We obtain an $\mathsf{FO}(<)$-rewriting of $(\mathcal{O}, \varkappa'(x))$ by adding to $\boldsymbol{Q}(x)$ the conjunct

$$\exists y\,[(x \leq y \leq x+2) \wedge (\mathit{Diagnostics}(y) \vee (\mathit{Disabled}(y) \wedge \exists z\,((y-3 \leq z < y) \wedge \mathit{Malfunction}(z))))].$$

1.2 Problems and Related Work

The initial problem we are interested in can be formulated in complexity-theoretic terms: given an *LTL* OMQ $\boldsymbol{q}$, determine the data complexity of answering $\boldsymbol{q}$ over any data instance $\mathcal{A}$ in a given signature Ξ. For simplicity's sake, let us assume that $\boldsymbol{q}$ is Boolean (with a yes/no certain answer). It is also convenient to think of each $\mathcal{A}$ as a word whose symbol at position ℓ is the set of all atoms in $\mathcal{A}$ with timestamp ℓ. Then the data instances $\mathcal{A}$ over which the answer to $\boldsymbol{q}$ is yes form a language, $\boldsymbol{L}(\boldsymbol{q})$, over the alphabet 2^{Ξ}. In fact, using the automata-theoretic view of *LTL* (Vardi & Wolper, 1986), one can show (see Proposition 5 below) that the language $\boldsymbol{L}(\boldsymbol{q})$ is regular, and so can be decided in NC^1 (Barrington & Thérien, 1988; Barrington, 1989).

This observation naturally leads to the task of recognising the complexity of the word problem for a given regular language. The circuit and descriptive complexity of regular languages was investigated by Barrington (1989), Barrington et al. (1992), Straubing (1994) who established an $\mathrm{AC}^0/\mathrm{ACC}^0/\mathrm{NC}^1$ trichotomy, gave algebraic characterisations of languages in these classes (implying that the trichotomy is decidable) and also in terms of extensions of FO. Namely, the regular languages $\boldsymbol{L}$ in AC^0 are definable by $\mathsf{FO}(<, \equiv)$-sentences with unary predicates $x \equiv 0 \,(\mathrm{mod}\ n)$; those in ACC^0 are definable by $\mathsf{FO}(<, \mathsf{MOD})$-sentences with quantifiers $\exists^n x\, \psi(x)$ checking whether the number of positions satisfying ψ is divisible by n; and all regular languages $\boldsymbol{L}$ are definable in $\mathsf{FO}(\mathsf{RPR})$ with relational primitive recursion (Compton & Laflamme, 1990). $\mathsf{FO}(<)$-definable regular languages, which are decidable in AC^0, were proven to be the same as star-free languages (McNaughton & Papert, 1971), and their algebraic characterisation as languages with aperiodic syntactic monoids was obtained by Schützenberger (1965). The problem of deciding whether the language of a given DFA $\mathfrak{A}$ is $\mathsf{FO}(<)$-definable is known to be PSpace-complete (Stern, 1985; Cho & Huynh, 1991; Bernátsky, 1997)[1]. However, the precise complexity of deciding whether a given regular language is in AC^0 and $\mathsf{FO}(<, \equiv)$-definable, or in ACC^0 and $\mathsf{FO}(<, \mathsf{MOD})$-definable, or NC^1-complete and is not $\mathsf{FO}(<, \mathsf{MOD})$-definable (unless $\mathrm{ACC}^0 = \mathrm{NC}^1$) has remained open. It will be the first major problem we address in this article.

1. This is also a special case of general results on finite monoids (Beaudry et al., 1992; Fleischer & Kufleitner, 2018).

The characterisation of regular languages in terms of FO-definability allows us to reformulate the initial problem in terms of FO-rewritability that reduces OMQ answering (under the open world assumption) to model checking various types of FO-formulas: given an *LTL* OMQ $\boldsymbol{q}$, how complex is it to decide whether $\boldsymbol{q}$ is $\mathsf{FO}(<)$-, $\mathsf{FO}(<, \mathsf{MOD})$- or $\mathsf{FO}(<, \mathsf{MOD})$-rewritable (that is, $\boldsymbol{L}(\boldsymbol{q})$ is $\mathsf{FO}(<)$-, $\mathsf{FO}(<, \equiv)$- or $\mathsf{FO}(<, \mathsf{MOD})$-definable)? Note that, by Kamp's Theorem (Kamp, 1968; Rabinovich, 2014), $\mathsf{FO}(<)$-rewritability reduces answering *LTL* OMQs to model checking *LTL*-formulas. $\mathsf{FO}(\mathsf{RPR})$-rewritability of all *LTL* OMQs was established by Artale et al. (2021) who also provided uniform rewritability results for various classes of *LTL* OMQs (to be defined below); see Table 2.

1.3 Our Contribution

The first main result of this paper consists of the following parts. Let $\mathcal{L}$ be one of the languages $\mathsf{FO}(<)$, $\mathsf{FO}(<, \equiv)$ or $\mathsf{FO}(<, \mathsf{MOD})$. First, using the algebraic characterisation results of Barrington (1989), Barrington et al. (1992), Straubing (1994), we obtain criteria for the $\mathcal{L}$-definability of the language $\boldsymbol{L}(\mathfrak{A})$ of any given DFA $\mathfrak{A}$ in terms of a limited part of the transition monoid of $\mathfrak{A}$ (Theorem 6). Then, using our criteria and generalising the construction of Cho and Huynh (1991), we show that deciding $\mathcal{L}$-definability of $\boldsymbol{L}(\mathfrak{A})$ for any minimal DFA $\mathfrak{A}$ is PSpace-hard (Theorem 8). Finally, we apply our criteria to give a PSpace-algorithm deciding $\mathcal{L}$-definability of $\boldsymbol{L}(\mathfrak{A})$ for not only any DFA but also any 2NFA $\mathfrak{A}$ (Theorem 15).

To investigate $\mathcal{L}$-rewritability of *LTL* OMQs $\boldsymbol{q} = (\mathcal{O}, \varkappa)$, we follow the classification of Artale et al. (2021), according to which the axioms of every *LTL* ontology $\mathcal{O}$ are given in the clausal form

$$\Box_P \Box_F \big(C_1 \land \dots \land C_k \ \to \ C_{k+1} \lor \dots \lor C_{k+m}\big), \tag{7}$$

where the C_i are atoms, possibly prefixed by the temporal operators $\bigcirc_F$, $\bigcirc_P$, $\Box_F$, $\Box_P$. Given any $\boldsymbol{o} \in \{\Box, \bigcirc, \Box\bigcirc\}$ and $\boldsymbol{c} \in \{\mathit{bool}, \mathit{horn}, \mathit{krom}, \mathit{core}\}$, we denote by $\mathit{LTL}_{\boldsymbol{c}}^{\boldsymbol{o}}$ the fragment of *LTL* with clauses (7), in which the C_i can only use the (future and past) operators indicated in $\boldsymbol{o}$, and $m \le 1$ if $\boldsymbol{c} = \mathit{horn}$; $k + m \le 2$ if $\boldsymbol{c} = \mathit{krom}$; $k + m \le 2$ and $m \le 1$ if $\boldsymbol{c} = \mathit{core}$; and arbitrary k, m if $\boldsymbol{c} = \mathit{bool}$. If $\boldsymbol{o}$ is omitted, the C_i are atomic. An $\mathit{LTL}_{\mathit{horn}}^{\boldsymbol{o}}$-ontology $\mathcal{O}$ is linear if, in each of its axioms (7), at most one C_i, for $1 \le i \le k$, can occur on the right-hand side of an axiom in $\mathcal{O}$ (is an IDB predicate in datalog parlance). We distinguish between arbitrary $\mathit{LTL}_{\boldsymbol{c}}^{\boldsymbol{o}}$ OMQs $\boldsymbol{q} = (\mathcal{O}, \varkappa)$, where $\mathcal{O}$ is any $\mathit{LTL}_{\boldsymbol{c}}^{\boldsymbol{o}}$ ontology and $\varkappa$ any *LTL*-formula with $\bigcirc$-, $\Box$- and $\Diamond$-operators; positive OMQs (OMPQs), where $\varkappa$ is $\to, \neg$-free; existential OMPQs (OMPEQs) with $\Box$-free $\varkappa$; and atomic OMQs (OMAQs) with atomic $\varkappa$.

The second main result of this article is the tight complexity bounds on deciding $\mathcal{L}$-rewritability (and so data complexity) of *LTL* OMQs from the classes defined above, which are summarised in Table 1. The ExpSpace upper bound in the first stripe is shown using our $\mathcal{L}$-definability criteria and exponential-size NFAs for *LTL* akin to those of Vardi (2007); in the proof of the matching lower bound, an exponential-size automaton is encoded in a polynomial-size ontology. If the ontology in an $\mathit{LTL}_{\mathit{horn}}^{\bigcirc}$ OMAQ is linear, we show that its language (yes-data instances) can be captured by a 2NFA with polynomially-many states, which allows us to reduce the complexity of deciding $\mathcal{L}$-rewritability to PSpace. However, for linear $\mathit{LTL}_{\mathit{horn}}^{\bigcirc}$ OMPQs (with more expressive queries $\varkappa$), the existence of polynomial-state 2NFAs remains open; instead, we show how the structure of the canonical models

class of OMQs	$\mathsf{FO}(<)$	$\mathsf{FO}(<, \equiv)$	$\mathsf{FO}(<, \mathsf{MOD})$
$LTL^{\bigcirc}_{horn}$ OMAQs $LTL^{\Box\bigcirc}_{bool}$ OMQs	ExpSpace [Th. 16]		
LTL_{krom} OMPEQs	ExpSpace [Th. 19]		
linear $LTL^{\bigcirc}_{horn}$ OMAQ	PSpace [Th. 22]		
linear $LTL^{\bigcirc}_{horn}$ OMPQs	PSpace [Th. 25]	PSpace [Th. 27]	?
$LTL^{\bigcirc}_{krom}$ OMAQs	coNP [Th. 28]	all in AC^0 (Artale et al., 2021)	–
$LTL^{\bigcirc}_{core}$ OMPEQs	Π^p_2 [Th. 30]		
$LTL^{\bigcirc}_{core}$ OMPQs	PSpace [Th. 35]		

Table 1: Complexity of deciding FO-rewritability of *LTL* OMQs.

for $LTL^{\bigcirc}_{horn}$-ontologies can be utilised to yield a PSpace algorithm. In the third stripe of the table, we deal with binary-clause ontologies. The coNP-completeness of deciding FO-rewritability of $LTL^{\bigcirc}_{krom}$ OMAQs is established using unary NFAs and results of Stockmeyer and Meyer (1973). The Π^p_2-completeness for $LTL^{\bigcirc}_{core}$ OMPEQs (without $\vee$ in ontologies but with $\wedge$, $\vee$, $\Diamond$ in queries) and the PSpace-completeness for $LTL^{\bigcirc}_{core}$ OMPQs (admitting $\Box$ in queries, too) can be explained by the fact that the combined complexity of answering such OMPEQs and OMPQs is NP-hard rather than tractable as in the previous case.

It might be of interest to compare the results in Table 1 with the complexity of deciding FO-rewritability (boundedness) of datalog queries and OMQs with a DL ontology and a conjunctive (CQ) or atomic query, which is:

- undecidable for linear datalog queries with binary predicates and for ternary linear datalog queries with a single recursive rule (Hillebrand et al., 1995; Marcinkowski, 1999);
- 2NExpTime-complete for monadic disjunctive datalog queries and OMQs with an $\mathcal{ALC}$ ontology and a CQ (Bourhis & Lutz, 2016; Feier et al., 2019);
- 2ExpTime-complete for monadic datalog queries (Cosmadakis et al., 1988; Benedikt et al., 2015), even with a single recursive rule (Kikot et al., 2021);
- NExpTime-complete for OMQs with an ontology in any DL between $\mathcal{ALC}$ and $\mathcal{SHIU}$ and an atomic query (Bienvenu et al., 2014);
- ExpTime-complete for OMQs with an $\mathcal{EL}$ ontology (Lutz & Sabellek, 2017, 2019);
- PSpace-complete for linear monadic programs (Cosmadakis et al., 1988; van der Meyden, 2000);
- NP-complete for linear monadic single rule programs (Vardi, 1988).

1.4 Structure

The article is organised in the following way. In the next section, we introduce and illustrate by multiple examples *LTL* OMQs and their semantics. We also briefly remind the reader of the basic algebraic and automata-theoretic notions that will be used later on in this article and show that FO-rewritability of *LTL* OMQs is equivalent to FO-definability of certain regular languages. In Section 3, we obtain algebraic characterisations of FO-definability, which are used in Sections 4 and 5 to show that deciding each type of FO-definability of regular languages is PSpace-complete. In Sections 6-8, we prove the complexity bounds from Table 1 and then conclude in Section 9. Some of the technical results and constructions are given in the appendices to the article.

2. Preliminaries

2.1 Temporal Ontology-mediated Queries

In our setting, the alphabet of linear temporal logic *LTL* comprises a set of *atomic concepts* (or simply *atoms*) A_i, $i < \omega$. *Basic temporal concepts*, C, are defined by the grammar

$$C \quad ::= \quad A_i \quad | \quad \Box_F C \quad | \quad \Box_P C \quad | \quad \bigcirc_F C \quad | \quad \bigcirc_P C$$

with the *temporal operators* $\Box_F/\Box_P$ (always in the future/past) and $\bigcirc_F/\bigcirc_P$ (at the next/previous moment). A *temporal ontology*, $\mathcal{O}$, is a finite set of *axioms* of the form[2]

$$C_1 \land \dots \land C_k \ \to \ C_{k+1} \lor \dots \lor C_{k+m}, \tag{8}$$

where $k, m \geq 0$, the C_i are basic temporal concepts, the empty $\land$ is $\top$, and the empty $\lor$ is $\bot$. Following the *DL-Lite* convention (Artale et al., 2009, 2015), we classify ontologies by the shape of their axioms and the temporal operators that can occur in them. Suppose $\boldsymbol{c} \in \{\mathit{horn}, \mathit{krom}, \mathit{core}, \mathit{bool}\}$ and $\boldsymbol{o} \in \{\Box, \bigcirc, \Box\bigcirc\}$. The axioms of an $\mathit{LTL}^{\boldsymbol{o}}_{\boldsymbol{c}}$*-ontology* may only contain occurrences of the (future and past) temporal operators in $\boldsymbol{o}$ and satisfy the following restrictions on k and m in (8) indicated by $\boldsymbol{c}$: *horn* requires $m \leq 1$, *krom* requires $k + m \leq 2$, *core* both $k + m \leq 2$ and $m \leq 1$, while *bool* imposes no restrictions. To illustrate, axioms (2) and (3) from Example 1 are allowed in all of these fragments, (4) is in $\mathit{LTL}^{\bigcirc}_{\mathit{core}}$, (6) can be expressed in $\mathit{LTL}^{\Box}_{\mathit{core}}$ and (5) can be expressed in $\mathit{LTL}^{\Box}_{\mathit{krom}}$ as explained in Remark 3 below.

A basic concept is called an *IDB* (intensional database) *concept* in an ontology $\mathcal{O}$ if its atom occurs on the right-hand side of some axiom in $\mathcal{O}$. The set of IDB atomic concepts in $\mathcal{O}$ is denoted by $\mathit{idb}(\mathcal{O})$. An $\mathit{LTL}^{\boldsymbol{o}}_{\mathit{horn}}$-ontology is called *linear* if each of its axioms $C_1 \land \dots \land C_k \to D$, where D is either a basic temporal concept C or $\bot$, contains *at most one* IDB concept C_i, for $1 \leq i \leq k$.

A *data instance*—or an *ABox* in description logic parlance—is a finite set $\mathcal{A}$ of atoms $A_i(\ell)$, for some *timestamps* $\ell \in \mathbb{Z}$, together with a finite interval $\mathsf{tem}(\mathcal{A}) = [m, n] \subseteq \mathbb{Z}$, the *active domain* of $\mathcal{A}$, such that $m \leq \ell \leq n$, for all $A_i(\ell) \in \mathcal{A}$. If $\mathcal{A} = \emptyset$, then $\mathsf{tem}(\mathcal{A})$ may also be $\emptyset$. Otherwise, we assume without loss of generality that $m = 0$. If $\mathsf{tem}(\mathcal{A})$ is not specified explicitly, it is assumed to be either empty or $[0, n]$, where n is the maximal timestamp in $\mathcal{A}$. By a *signature*, Ξ, we mean any finite set of atomic concepts. An ABox $\mathcal{A}$ is a Ξ-*ABox* if $A_i(\ell) \in \mathcal{A}$ implies $A_i \in \Xi$.

We query ABoxes by means of *temporal concepts*, $\varkappa$, which are *LTL*-formulas built from the atoms A_i, Booleans $\land$, $\lor$, $\neg$, temporal operators $\bigcirc_F$, $\Box_F$, $\Diamond_F$ (eventually) and their

2. From now on, to improve readability we make the prefix $\Box_P \Box_F$ in axioms implicit (which is taken into account in their semantics).

past-time counterparts $\bigcirc_P$, $\Box_P$, $\Diamond_P$ (previously). If $\varkappa$ does not contain $\neg$, we call it *positive*; if $\varkappa$ does not contain $\Box_P$ and $\Box_F$ either, we call it *positive existential*.

A *temporal interpretation* is a structure of the form $\mathcal{I} = (\mathbb{Z}, A_0^{\mathcal{I}}, A_1^{\mathcal{I}}, \dots)$ with $A_i^{\mathcal{I}} \subseteq \mathbb{Z}$, for every $i < \omega$. The *extension* $\varkappa^{\mathcal{I}}$ of a temporal concept $\varkappa$ in $\mathcal{I}$ is defined inductively as usual in *LTL* under the 'strict semantics' (Gabbay, Kurucz, Wolter, & Zakharyaschev, 2003; Demri, Goranko, & Lange, 2016):

$$(\bigcirc_F \varkappa)^{\mathcal{I}} = \big\{\, n \in \mathbb{Z} \mid n+1 \in \varkappa^{\mathcal{I}} \,\big\},$$
$$(\Box_F \varkappa)^{\mathcal{I}} = \big\{\, n \in \mathbb{Z} \mid k \in \varkappa^{\mathcal{I}} \text{ for all } k > n \,\big\},$$
$$(\Diamond_F \varkappa)^{\mathcal{I}} = \big\{\, n \in \mathbb{Z} \mid \text{there is } k > n \text{ with } k \in \varkappa^{\mathcal{I}} \,\big\},$$

and symmetrically for the past-time operators. We regard $\mathcal{I}, n \models \varkappa$ as synonymous to $n \in \varkappa^{\mathcal{I}}$. An axiom (7) is *true* in an interpretation $\mathcal{I}$ if $C_1^{\mathcal{I}} \cap \dots \cap C_k^{\mathcal{I}} \subseteq C_{k+1}^{\mathcal{I}} \cup \dots \cup C_{k+m}^{\mathcal{I}}$. An interpretation $\mathcal{I}$ is a *model* of $\mathcal{O}$ if all axioms of $\mathcal{O}$ are true in $\mathcal{I}$; it is a *model* of $\mathcal{A}$ if $A_i(\ell) \in \mathcal{A}$ implies $\ell \in A_i^{\mathcal{I}}$.

An $LTL_{\boldsymbol{c}}^{\boldsymbol{o}}$ *ontology-mediated query* (OMQ) is a pair of the form $\boldsymbol{q} = (\mathcal{O}, \varkappa)$, where $\mathcal{O}$ is an $LTL_{\boldsymbol{c}}^{\boldsymbol{o}}$ ontology and $\varkappa$ a temporal concept. If $\varkappa$ is positive, we call $\boldsymbol{q}$ a *positive OMQ* (OMPQ, for short), if $\varkappa$ is positive existential, we call $\boldsymbol{q}$ a *positive existential OMQ* (OMPEQ), and if $\varkappa$ is an atomic concept, we call $\boldsymbol{q}$ *atomic* (OMAQ). The set of atomic concepts occurring in $\boldsymbol{q}$ (in $\mathcal{O}$) is denoted by $\mathsf{sig}(\boldsymbol{q})$ (respectively, $\mathsf{sig}(\mathcal{O})$).

We can treat $\boldsymbol{q} = (\mathcal{O}, \varkappa)$ as a *Boolean* OMQ, which returns yes/no, or as a *specific* OMQ, which returns timestamps from the ABox in question assigned to the free variable, say x, in the standard FO-translation of $\varkappa$. In the latter case, we write $\boldsymbol{q}(x) = (\mathcal{O}, \varkappa(x))$. More precisely, the *certain answer* to a Boolean OMQ $\boldsymbol{q} = (\mathcal{O}, \varkappa)$ over an ABox $\mathcal{A}$ is yes if, for every model $\mathcal{I}$ of $\mathcal{O}$ and $\mathcal{A}$, there is $k \in \mathbb{Z}$ such that $k \in \varkappa^{\mathcal{I}}$, in which case we write $(\mathcal{O}, \mathcal{A}) \models \exists x\, \varkappa(x)$. If $(\mathcal{O}, \mathcal{A}) \not\models \exists x \varkappa(x)$, the certain answer to $\boldsymbol{q}$ over $\mathcal{A}$ is no. We write $(\mathcal{O}, \mathcal{A}) \models \varkappa(k)$, for $k \in \mathbb{Z}$, if $k \in \varkappa^{\mathcal{I}}$ in all models $\mathcal{I}$ of $\mathcal{O}$ and $\mathcal{A}$. A *certain answer* to a specific OMQ $\boldsymbol{q}(x) = (\mathcal{O}, \varkappa(x))$ over $\mathcal{A}$ is any $k \in \mathsf{tem}(\mathcal{A})$ with $(\mathcal{O}, \mathcal{A}) \models \varkappa(k)$. By the *answering* (or *evaluation*) *problem* for $\boldsymbol{q}$ or $\boldsymbol{q}(x)$ we understand the decision problem '$(\mathcal{O}, \mathcal{A}) \models^? \exists x \varkappa(x)$' or '$(\mathcal{O}, \mathcal{A}) \models^? \varkappa(k)$' with input $\mathcal{A}$ or, respectively, $\mathcal{A}$ and $k \in \mathsf{tem}(\mathcal{A})$, We say that $\boldsymbol{q}/\boldsymbol{q}(x)$ is in a complexity class $\mathcal{C}$ if the answering problem for $\boldsymbol{q}/\boldsymbol{q}(x)$ is in $\mathcal{C}$.

Example 2. (i) Suppose $\mathcal{O}_1 = \{A \to \Box_F B,\ \Box_F B \to C\}$ and $\boldsymbol{q}_1 = (\mathcal{O}_1, C \land D)$. The certain answer to $\boldsymbol{q}_1$ over $\mathcal{A}_1 = \{D(0), B(1), A(1)\}$ is yes, and no over $\mathcal{A}_2 = \{D(0), A(1)\}$. The only answer to $\boldsymbol{q}_1(x) = \big(\mathcal{O}_1, (C \land D)(x)\big)$ over $\mathcal{A}_1$ is 0.

(ii) Let $\mathcal{O}_2 = \{\, \bigcirc_P A \to B,\ \bigcirc_P B \to A,\ A \land B \to \bot \,\}$. The certain answer to $\boldsymbol{q}_2 = (\mathcal{O}_2, C)$ over $\mathcal{A}_1 = \{A(0)\}$ is no, and yes over $\mathcal{A}_2 = \{A(0), A(1)\}$. There are no certain answers to $\boldsymbol{q}_2(x) = (\mathcal{O}_1, C(x))$ over $\mathcal{A}_1$, while over $\mathcal{A}_2$ the answers are 0 and 1.

(iii) Consider next $\mathcal{O}_3 = \{\bigcirc_P B_k \land A_0 \to B_k,\ \bigcirc_P B_{1-k} \land A_1 \to B_k \mid k = 0, 1\}$. For any word $\boldsymbol{e} = e_1 \dots e_n \in \{0,1\}^n$, let

$$\mathcal{A}_{\boldsymbol{e}} = \{B_0(0)\} \cup \{A_{e_i}(i) \mid 0 < i \leq n\} \cup \{E(n)\}.$$

The certain answer to $\boldsymbol{q}_3 = (\mathcal{O}_3, B_0 \land E)$ over $\mathcal{A}_{\boldsymbol{e}}$ is yes iff the number of 1s in $\boldsymbol{e}$ is even.

(iv) Let $\mathcal{O}_4 = \{A \to \bigcirc_F B\}$ and $\boldsymbol{q}_4 = (\mathcal{O}_4, B)$. Then, the answer to $\boldsymbol{q}_4$ over $\mathcal{A} = \{A(0)\}$ is yes; however, there are no certain answers to $\boldsymbol{q}_4(x) = (\mathcal{O}_4, B(x))$ over $\mathcal{A}$.

(v) Finally, suppose $\mathcal{O}_5 = \{A \to B \lor \bigcirc_F B\}$. The certain answer to $\boldsymbol{q}_5 = (\mathcal{O}_5, B)$ over $\mathcal{A} = \{A(0), C(1)\}$ is yes; however, there are no certain answers to $\boldsymbol{q}_5(x)$ over $\mathcal{A}$. $\dashv$

Thus, as shown by Example 2 (iv) and (v), a Boolean OMAQ $\boldsymbol{q} = (\mathcal{O}, B)$ can have an answer yes over an ABox $\mathcal{A}$ even though the set of certain answers to the specific OMAQ $\boldsymbol{q}(x) = (\mathcal{O}, B(x))$ over $\mathcal{A}$ is empty. (Clearly, the existence of certain answers to $\boldsymbol{q}(x)$ over $\mathcal{A}$ implies that the answer to $\boldsymbol{q}$ over $\mathcal{A}$ is yes.) In (iv), the reason for the absence of certain answers to $\boldsymbol{q}_4(x)$ is that any $k \in \mathbb{Z}$ with $(\mathcal{O}, \mathcal{A}) \models B(k)$ is not in $\mathsf{tem}(\mathcal{A})$. In ($v$), the reason is that there is no $k \in \mathbb{Z}$ with $(\mathcal{O}, \mathcal{A}) \models B(k)$ even though every model $\mathcal{I}$ of $\mathcal{O}$ and $\mathcal{A}$ contains some $k \in \mathsf{tem}(\mathcal{A}) \subseteq \mathbb{Z}$ with $\mathcal{I}, k \models B$.

Two OMQs are called Ξ-*equivalent*, for a signature Ξ, if they return the same certain answers over any Ξ-ABox. Without loss of generality, we assume that, when answering an *LTL* OMQ $\boldsymbol{q}$ or $\boldsymbol{q}(x)$ over Ξ-ABoxes, we always have $\Xi \subseteq \mathsf{sig}(\boldsymbol{q})$. Indeed, if this is not the case, we can extend the ontology of $\boldsymbol{q}$ with $|\Xi|$-many dummy axioms of the form $A \to A$ and obtain a Ξ-equivalent OMQ.

Remark 3. If arbitrary *LTL*-formulas (possibly with the until or since operators) in the scope of $\Box_P \Box_F$ are used as axioms of an ontology $\mathcal{O}$, then one can construct an $LTL^{\Box\bigcirc}_{bool}$ ontology $\mathcal{O}'$ that is a *model-conservative extension* of $\mathcal{O}$ (e.g., Fisher et al., 2001; Artale et al., 2013). For example, let $\mathcal{O}'$ be the result of replacing axiom (5) in $\mathcal{O}$ from Example 1 by two axioms $\mathit{Malfunction} \land \Box_F X \to \bot$ and $\top \to X \lor \mathit{Diagnostics}$, for a fresh X. Then the OMQ $\boldsymbol{q} = (\mathcal{O}, \varkappa)$ is $\mathsf{sig}(\boldsymbol{q})$-equivalent to $\boldsymbol{q}' = (\mathcal{O}', \varkappa)$. Axiom (6) can be replaced with $\mathit{Diagnostics} \to \Box_P Y$ and $\mathit{Disabled} \land Y \to \bot$ with fresh Y.

Similarly, every $LTL^{\Box\bigcirc}_{horn}$ OMQ $\boldsymbol{q} = (\mathcal{O}, \varkappa)$ has the same certain answers over any $\mathsf{sig}(\boldsymbol{q})$-ABox as an $LTL^{\Box\bigcirc}_{horn}$ OMQ $\boldsymbol{q}' = (\mathcal{O}', \varkappa)$, in which $\mathcal{O}'$ contains axioms of the form $\boldsymbol{C} \to \bot$ or $\boldsymbol{C} \to B$ only, for some $\boldsymbol{C} = C_1 \land \cdots \land C_n$ and an atomic concept B. For example, the axiom $A \to \bigcirc_F \Box_F B$ can be replaced by $\bigcirc_P A \to X$, $\bigcirc_P X \to X$, and $\bigcirc_P X \to B$ with fresh X. Note also that if $\mathcal{O}$ is a linear $LTL^{\bigcirc}_{horn}$ ontology, then $\mathcal{O}'$ is also a linear $LTL^{\bigcirc}_{horn}$ ontology.

We now introduce the central notion of this article, which reduces answering OMQs to evaluating FO-formulas over structures representing ABoxes.

Let $\mathcal{L}$ be a class of FO-formulas that can be interpreted over finite linear orders. A Boolean OMQ $\boldsymbol{q}$ is $\mathcal{L}$-*rewritable over* Ξ-*ABoxes* if there is an $\mathcal{L}$-sentence $\boldsymbol{Q}$ such that, for any Ξ-ABox $\mathcal{A}$, the certain answer to $\boldsymbol{q}$ over $\mathcal{A}$ is yes iff $\mathfrak{S}_\mathcal{A} \models \boldsymbol{Q}$. Here, $\mathfrak{S}_\mathcal{A}$ is a structure[3] with domain $\mathsf{tem}(\mathcal{A})$ ordered by $<$, in which $\mathfrak{S}_\mathcal{A} \models A_i(\ell)$ iff $A_i(\ell) \in \mathcal{A}$. A specific OMQ $\boldsymbol{q}(x)$ is $\mathcal{L}$-*rewritable over* Ξ-*ABoxes* if there is an $\mathcal{L}$-formula $\boldsymbol{Q}(x)$ with one free variable x such that, for any Ξ-ABox $\mathcal{A}$, k is a certain answer to $\boldsymbol{q}(x)$ over $\mathcal{A}$ iff $\mathfrak{S}_\mathcal{A} \models \boldsymbol{Q}(k)$. The sentence $\boldsymbol{Q}$ and formula $\boldsymbol{Q}(x)$ are called $\mathcal{L}$-*rewritings* of the OMQs $\boldsymbol{q}$ and $\boldsymbol{q}(x)$, respectively.

We require four languages $\mathcal{L}$ for rewriting *LTL* OMQs, which are listed below in order of increasing expressive power:

$\mathsf{FO}(<)$**:** (monadic) first-order formulas with the built-in predicate $<$ for order;

$\mathsf{FO}(<, \equiv)$**:** $\mathsf{FO}(<)$-formulas with unary predicates $x \equiv 0 \,(\mathrm{mod}\ N)$, for all $N > 1$;

3. We allow structures with the empty domain, in which $\exists x\, (x = x)$ is false (e.g., Hodges, 1993).

$\mathsf{FO}(<, \mathsf{MOD})$: $\mathsf{FO}(<)$-formulas with quantifiers $\exists^N x$, for all $N > 1$, that are defined by taking $\mathfrak{S}_\mathcal{A} \models \exists^N x\, \psi(x)$ iff the cardinality of $\{n \in \mathsf{tem}(\mathcal{A}) \mid \mathfrak{S}_\mathcal{A} \models \psi(n)\}$ is divisible by N (note that $x \equiv 0 \pmod N$ is definable as $\exists^N y\, (y < x)$);

$\mathsf{FO}(\mathsf{RPR})$: $\mathsf{FO}(<)$ with relational primitive recursion (Compton & Laflamme, 1990).

As well-known, $\mathsf{FO}(<, \equiv)$ is strictly more expressive than $\mathsf{FO}(<)$ and strictly less expressive than $\mathsf{FO}(<, \mathsf{MOD})$, which is illustrated by the examples below.

Example 4. (i) An $\mathsf{FO}(<)$-rewriting of $\boldsymbol{q}_1(x)$ from Example 2 is

$$\boldsymbol{Q}_1(x) = D(x) \land [C(x) \lor \exists y\, (A(y) \land \forall z\, ((x < z \leq y) \to B(z)))],$$

$\exists x\, \boldsymbol{Q}_1(x)$ is an $\mathsf{FO}(<)$-rewriting of $\boldsymbol{q}_1$.

(ii) An $\mathsf{FO}(<, \equiv)$-rewriting of $\boldsymbol{q}_2(x)$ is

$$\begin{aligned}\boldsymbol{Q}_2(x) = \ & C(x) \lor \exists x, y\, [(A(x) \land A(y) \land \mathsf{odd}(x,y)) \lor \\ & \qquad\qquad (B(x) \land B(y) \land \mathsf{odd}(x,y)) \lor (A(x) \land B(y) \land \neg\mathsf{odd}(x,y))],\end{aligned}$$

where $\mathsf{odd}(x,y) = \big(x \equiv 0 \pmod 2 \leftrightarrow y \not\equiv 0 \pmod 2\big)$ implies that $|x - y|$ is odd; $\exists x\, \boldsymbol{Q}_2(x)$ is an $\mathsf{FO}(<, \equiv)$-rewriting of $\boldsymbol{q}_2$. Recall that odd is not $\mathsf{FO}(<)$-expressible (Libkin, 2004).

(iii) The OMQ $\boldsymbol{q}_3$ is not rewritable to an FO-formula with any numeric predicates as PARITY is not in AC^0 (Furst et al., 1984); the following sentence is an $\mathsf{FO}(<, \mathsf{MOD})$-rewriting of $\boldsymbol{q}_3$:

$$\begin{aligned}\boldsymbol{Q}_3 = \exists x, y\, \big[& E(x) \land (y \leq x) \land \forall z\, \big((y < z \leq x) \to A_0(z) \lor A_1(z)\big) \land \\ & \big((B_0(y) \land \exists^2 z\, ((y < z \leq x) \land A_1(z))) \lor (B_1(y) \land \neg\exists^2 z\, ((y < z \leq x) \land A_1(z)))\big)\big].\end{aligned}$$

(iv) An $\mathsf{FO}(<)$-rewriting of $\boldsymbol{q}_4(x)$ is $B(x) \lor A(x-1)$; an $\mathsf{FO}(<)$-rewriting of the Boolean query $\boldsymbol{q}_4$ is $\boldsymbol{Q}_4 = \exists x\, (A(x) \lor B(x))$.

(v) $\boldsymbol{Q}_4$ is also an $\mathsf{FO}(<)$-rewriting of $\boldsymbol{q}_5$; $B(x)$ is an $\mathsf{FO}(<)$-rewriting of $\boldsymbol{q}_5(x)$. $\dashv$

As shown by Artale et al. (2021), all Boolean and specific *LTL* OMQs are $\mathsf{FO}(\mathsf{RPR})$-rewritable and specific OMPQs can be classified syntactically by their rewritability type as shown in Table 2. This means, e.g., that all $\mathit{LTL}^{\Box\bigcirc}_{\mathit{core}}$ OMPQs are $\mathsf{FO}(<, \equiv)$-rewritable, with some of them being not $\mathsf{FO}(<)$-rewritable. It is to be noted that $\mathsf{FO}(<, \mathsf{MOD})$-rewritable OMQs such as $\boldsymbol{q}_3$ in Example 2 are not captured by these syntactic classes.

Our aim here is to understand how complex it is to decide the optimal type of FO-rewritability for a given *LTL* OMQ $\boldsymbol{q}$ over Ξ-ABoxes. As this will rely on an intimate connection between $\mathcal{L}$-rewritability of OMQs and $\mathcal{L}$-definability of certain regular languages, we briefly remind the reader of the basic algebraic and automata-theoretic notions that are used in the remainder of the article.

2.2 Monoids and Groups

A *semigroup* is a structure $\mathfrak{S} = (S, \cdot)$, where $\cdot$ is an associative binary operation. For $s, s' \in S$ and $n > 0$, we set $s^n = \underbrace{s \cdot s \cdot \ldots \cdot s}_{n}$ and often write ss' for $s \cdot s'$. An element s of

<table>
<tr><th rowspan="2">$\boldsymbol{c}$</th><th colspan="2">OMAQs</th><th colspan="2">OMPQs</th></tr>
<tr><th>$LTL_{\boldsymbol{c}}^{\Box}$</th><th>$LTL_{\boldsymbol{c}}^{\bigcirc}$ and $LTL_{\boldsymbol{c}}^{\Box\bigcirc}$</th><th>$LTL_{\boldsymbol{c}}^{\Box}$</th><th>$LTL_{\boldsymbol{c}}^{\bigcirc}$ and $LTL_{\boldsymbol{c}}^{\Box\bigcirc}$</th></tr>
<tr><td>bool</td><td rowspan="4">$\mathsf{FO}(<)$</td><td>$\mathsf{FO}(\mathsf{RPR})$</td><td rowspan="2">$\mathsf{FO}(\mathsf{RPR})$</td><td rowspan="3">FO(RPR)</td></tr>
<tr><td>krom</td><td>$\mathsf{FO}(<,\equiv)$</td></tr>
<tr><td>horn</td><td>$\mathsf{FO}(\mathsf{RPR})$</td><td rowspan="2">$\mathsf{FO}(<)$</td></tr>
<tr><td>core</td><td>$\mathsf{FO}(<,\equiv)$</td><td>$\mathsf{FO}(<,\equiv)$</td></tr>
</table>

Table 2: Rewritability of specific *LTL* OMQs.

$\mathfrak{S}$ is *idempotent* if $s^2 = s$. An element e is an *identity* in $\mathfrak{S}$ if $e \cdot x = x \cdot e = x$ for all $x \in S$ (such an e is unique, if exists). The identity element is clearly idempotent. A *monoid* is a semigroup with an identity element. For any element s in a monoid, we set $s^0 = e$. A monoid $\mathfrak{S} = (S, \cdot)$ is a *group* if, for any $x \in S$, there is $x^- \in S$—the *inverse of* x—such that $x \cdot x^- = x^- \cdot x = e$ (every element of a group has a unique inverse). A group is *trivial* if it has one element, and *nontrivial* otherwise.

Given two groups $\mathfrak{G} = (G, \cdot)$ and $\mathfrak{G}' = (G', \cdot')$, a map $h \colon G \to G'$ is a *group homomorphism from* $\mathfrak{G}$ *to* $\mathfrak{G}'$ if $h(g_1 \cdot g_2) = h(g_1) \cdot' h(g_2)$ for all $g_1, g_2 \in G$. (It is easy to see that any group homomorphism maps the identity of $\mathfrak{G}$ to the identity of $\mathfrak{G}'$ and preserves the inverses. The set $\{h(g) \mid g \in G\}$ is closed under $\cdot'$, and so is a group, the *image of* $\mathfrak{G}$ *under* h.) $\mathfrak{G}$ is a *subgroup of* $\mathfrak{G}'$ if $G \subseteq G'$ and the identity map id_G is a group homomorphism. Given $X \subseteq G$, the *subgroup of* $\mathfrak{G}$ *generated by* X is the smallest subgroup of $\mathfrak{G}$ containing X. The *order* $o_{\mathfrak{G}}(g)$ of an element g in $\mathfrak{G}$ is the smallest positive number n with $g^n = e$, which always exists. Clearly, $o_{\mathfrak{G}}(g) = o_{\mathfrak{G}}(g^-)$ and, if $g^k = e$ then $o_{\mathfrak{G}}(g)$ divides k. Also,

$$\text{if } g \text{ is a nonidentity element in a group } \mathfrak{G}\text{, then } g^k \neq g^{k+1} \text{ for any } k. \tag{9}$$

A semigroup $\mathfrak{S}' = (S', \cdot')$ is a *subsemigroup* of a semigroup $\mathfrak{S} = (S, \cdot)$ if $S' \subseteq S$ and $\cdot'$ is the restriction of $\cdot$ to S'. Given a monoid $\boldsymbol{M} = (M, \cdot)$ and a set $S \subseteq M$, we say that S *contains the group* $\mathfrak{G} = (G, \cdot')$, if $G \subseteq S$ and $\mathfrak{G}$ is a subsemigroup of $\boldsymbol{M}$. Note that we do **not** require the identity of $\boldsymbol{M}$ to be in $\mathfrak{G}$, even if it is in S. If $S = M$, we also say that $\boldsymbol{M}$ *contains the group* $\mathfrak{G}$, or $\mathfrak{G}$ *is in* $\boldsymbol{M}$. We call a monoid $\boldsymbol{M}$ *aperiodic* if it does not contain any nontrivial groups.

Let $\mathfrak{S} = (S, \cdot)$ be a finite semigroup and $s \in S$. By the pigeonhole principle, there exist $i, j \geq 1$ such that $i + j \leq |S| + 1$ and $s^i = s^{i+j}$. Take the minimal such numbers, that is, let $i_s, j_s \geq 1$ be such that $i_s + j_s \leq |S| + 1$ and $s^{i_s} = s^{i_s + j_s}$ but $s^{i_s}, s^{i_s+1}, \dots, s^{i_s+j_s-1}$ are all different. Then clearly $\mathfrak{G}_s = (G_s, \cdot)$, where $G_s = \{s^{i_s}, s^{i_s+1}, \dots, s^{i_s+j_s-1}\}$, is a subsemigroup of $\mathfrak{S}$. It is easy to see that there is $m \geq 1$ with $i_s \leq m \cdot j_s < i_s + j_s \leq |S| + 1$, and so $s^{m \cdot j_s}$ is idempotent. Thus, for every element s in a semigroup $\mathfrak{S}$, we have the following:

$$\text{there is } n \geq 1 \text{ such that } s^n \text{ is idempotent;} \tag{10}$$

$$\mathfrak{G}_s \text{ is a group in } \mathfrak{S} \text{ (isomorphic to the cyclic group } \mathbb{Z}_{j_s}\text{);} \tag{11}$$

$$\mathfrak{G}_s \text{ is nontrivial iff } s^n \neq s^{n+1} \text{ for any } n. \tag{12}$$

Let $\delta\colon Q \to Q$ be a function on a finite set $Q \neq \emptyset$. For any $p \in Q$, the subset $\{\delta^k(p) \mid k < \omega\}$ with the obvious multiplication is a semigroup, and so we have:

$$\text{for every } p \in Q \text{, there is } n_p \geq 1 \text{ such that } \delta^{n_p}\big(\delta^{n_p}(p)\big) = \delta^{n_p}(p); \tag{13}$$

$$\text{there exist } q \in Q \text{ and } n \geq 1 \text{ such that } q = \delta^n(q); \tag{14}$$

$$\text{for every } q \in Q \text{, if } q = \delta^k(q) \text{ for some } k \geq 1, \text{ then there is } n,\ 1 \leq n \leq |Q| \text{, with } q = \delta^n(q). \tag{15}$$

For a definition of *solvable* and *unsolvable* groups the reader is referred to Rotman (1999). In this article, we only use the fact that any homomorphic image of a solvable group is solvable and the Kaplan–Levy criterion (2010) (generalising Thompson's (1968, Corollary 3)) according to which a finite group $\mathfrak{G}$ is unsolvable iff it contains elements a, b, c such that $o_{\mathfrak{G}}(a) = 2$, $o_{\mathfrak{G}}(b)$ is an odd prime, $o_{\mathfrak{G}}(c) > 1$ and coprime to both 2 and $o_{\mathfrak{G}}(b)$, and abc is the identity of $\mathfrak{G}$.

A one-to-one and onto function on a finite set S is called a *permutation on S*. The *order of a permutation* δ is its order in the group of all permutations on S (whose operation is composition, and its identity element is the identity permutation id_S). We use the standard cycle notation for permutations.

Suppose that $\mathfrak{G}$ is a monoid of $Q \to Q$ functions, for some finite set $Q \neq \emptyset$. Let $S = \{q \in Q \mid e_{\mathfrak{G}}(q) = q\}$, where $e_{\mathfrak{G}}$ the identity element in $\mathfrak{G}$. For every function δ in $\mathfrak{G}$, let $\delta\restriction_S$ denote the restriction of δ to S. Then

$$\mathfrak{G} \text{ is a group iff } \delta\restriction_S \text{ is a permutation on } S \text{, for every } \delta \text{ in } \mathfrak{G}; \tag{16}$$

$$\text{if } \mathfrak{G} \text{ is a group and } \delta \text{ is a nonindentity element in it, then } \delta\restriction_S \neq \mathsf{id}_S \text{ and the order of the permutation } \delta\restriction_S \text{ divides } o_{\mathfrak{G}}(\delta). \tag{17}$$

2.3 Automata, Languages, and OMQs

A *two-way nondeterministic finite automaton* is a quintuple $\mathfrak{A} = (Q, \Sigma, \delta, Q_0, F)$ that consists of an alphabet Σ, a finite set Q of states with a subset $Q_0 \neq \emptyset$ of initial states and a subset F of accepting states, and a transition function $\delta\colon Q \times \Sigma \to 2^{Q\times\{-1,0,1\}}$ indicating the next state and whether the head should move left (-1), right (1), or stay put. If $Q_0 = \{q_0\}$ and $|\delta(q, a)| = 1$, for all $q \in Q$ and $a \in \Sigma$, then $\mathfrak{A}$ is *deterministic*, in which case we write $\mathfrak{A} = (Q, \Sigma, \delta, q_0, F)$. If $\delta(q, a) \subseteq Q \times \{1\}$, for all $q \in Q$ and $a \in \Sigma$, then $\mathfrak{A}$ is a *one-way* automaton, and we write $\delta\colon Q \times \Sigma \to 2^Q$. As usual, DFA and NFA refer to one-way deterministic and non-deterministic finite automata, respectively, while 2DFA and 2NFA to the corresponding two-way automata. Given a 2NFA $\mathfrak{A}$, we write $q \to_{a,d} q'$ if $(q', d) \in \delta(q, a)$; given an NFA $\mathfrak{A}$, we write $q \to_a q'$ if $q' \in \delta(q, a)$. A *run* of a 2NFA $\mathfrak{A}$ is a word in $(Q \times \mathbb{N})^*$. A run $(q_0, i_0), \dots, (q_m, i_m)$ is a *run of* $\mathfrak{A}$ *on a word* $w = a_0 \dots a_n \in \Sigma^*$ if $q_0 \in Q_0$, $i_0 = 0$ and there exist $d_0, \dots, d_{m-1} \in \{-1, 0, 1\}$ such that $q_j \to_{a_{i_j},d_j} q_{j+1}$ and $i_{j+1} = i_j + d_j$ for all j, $0 \leq j < m$. The run is *accepting* if $q_m \in F$, $i_m = n + 1$. $\mathfrak{A}$ *accepts* $w \in \Sigma^*$ if there is an accepting run of $\mathfrak{A}$ on w; the language $\boldsymbol{L}(\mathfrak{A})$ of $\mathfrak{A}$ is the set of all words accepted by $\mathfrak{A}$.

Given an NFA $\mathfrak{A}$, states $q, q' \in Q$, and $w = a_0 \dots a_n \in \Sigma^*$, we write $q \to_w q'$ if either $w = \varepsilon$ and $q' = q$ or there is a run of $\mathfrak{A}$ on w that starts with $(q_0, 0)$ and ends with $(q', n+1)$. We say that a state $q \in Q$ is *reachable* if $q' \to_w q$, for some $q' \in Q_0$ and $w \in \Sigma^*$.

Given a DFA $\mathfrak{A} = (Q, \Sigma, \delta, q_0, F)$ and a word $w \in \Sigma^*$, we define a function $\delta_w \colon Q \to Q$ by taking $\delta_w(q) = q'$ iff $q \to_w q'$. We also define an equivalence relation $\sim$ on the set $Q^r \subseteq Q$ of reachable states by taking $q \sim q'$ iff, for every $w \in \Sigma^*$, we have $\delta_w(q) \in F$ just in case $\delta_w(q') \in F$. We denote the $\sim$-class of q by $q/_\sim$, and let $X/_\sim = \{q/_\sim \mid q \in X\}$ for $X \subseteq Q^r$. Define $\tilde{\delta}_w \colon Q^r/_\sim \to Q^r/_\sim$ by taking $\tilde{\delta}_w(q/_\sim) = \delta_w(q)/_\sim$. Then $\big(Q^r/_\sim, \Sigma, \tilde{\delta}, q_0/_\sim, (F \cap Q^r)/_\sim\big)$ is the *minimal DFA* whose language coincides with the language of $\mathfrak{A}$. Given a regular language $\boldsymbol{L}$, we denote by $\mathfrak{A}_{\boldsymbol{L}}$ the minimal DFA whose language is $\boldsymbol{L}$.

The *transition monoid of* a DFA $\mathfrak{A}$ is $M(\mathfrak{A}) = (\{\delta_w \mid w \in \Sigma^*\}, \cdot)$ with $\delta_v \cdot \delta_w = \delta_{vw}$, for any v, w. The *syntactic monoid* $M(\boldsymbol{L})$ *of* $\boldsymbol{L}$ is the transition monoid $M(\mathfrak{A}_{\boldsymbol{L}})$ of $\mathfrak{A}_{\boldsymbol{L}}$. The *syntactic morphism of* $\boldsymbol{L}$ is the map $\eta_{\boldsymbol{L}}$ from Σ^* to the domain of $M(\boldsymbol{L})$ defined by $\eta_{\boldsymbol{L}}(w) = \tilde{\delta}_w$. We call $\eta_{\boldsymbol{L}}$ *quasi-aperiodic* if $\eta_{\boldsymbol{L}}(\Sigma^t)$ is aperiodic for every $t < \omega$.

Let $\mathcal{L} \in \{\mathsf{FO}(<), \mathsf{FO}(<, \equiv), \mathsf{FO}(<, \mathsf{MOD})\}$. A language $\boldsymbol{L}$ over Σ is $\mathcal{L}$-*definable* if there is an $\mathcal{L}$-sentence φ in the signature Σ, whose symbols are treated as unary predicates, such that, for any $w \in \Sigma^*$, we have $w = a_0 \dots a_n \in \boldsymbol{L}$ iff $\mathfrak{S}_w \models \varphi$, where $\mathfrak{S}_w$ is an FO-structure with domain $\{0, \dots, n\}$ ordered by $<$, in which $\mathfrak{S}_w \models a(i)$ iff $a = a_i$, for $0 \le i \le n$.

Table 3 summarises the known results that connect definability of a regular language $\boldsymbol{L}$ with properties of the syntactic monoid $M(\boldsymbol{L})$ and syntactic morphism $\eta_{\boldsymbol{L}}$ (Barrington et al., 1992) and with its circuit complexity under a reasonable binary encoding of $\boldsymbol{L}$'s alphabet (e.g., Bernátsky, 1997, Lemma 2.1) and the assumption that $\mathrm{ACC}^0 \neq \mathrm{NC}^1$. We also remind the reader that a regular language is $\mathsf{FO}(<)$-definable iff it is star-free (Straubing, 1994), and that $\mathrm{AC}^0 \subsetneq \mathrm{ACC}^0 \subseteq \mathrm{NC}^1$ (Straubing, 1994; Jukna, 2012).

definability of $\boldsymbol{L}$	algebraic characterisation of $\boldsymbol{L}$	circuit complexity
$\mathsf{FO}(<)$	$M(\boldsymbol{L})$ is aperiodic	in AC^0
$\mathsf{FO}(<, \equiv)$	$\eta_{\boldsymbol{L}}$ is quasi-aperiodic	
$\mathsf{FO}(<, \mathsf{MOD})$	all groups in $M(\boldsymbol{L})$ are solvable	in ACC^0
$\mathsf{FO}(\mathsf{RPR})$	arbitrary $M(\boldsymbol{L})$	in NC^1
not in $\mathsf{FO}(<, \mathsf{MOD})$	$M(\boldsymbol{L})$ has an unsolvable group	NC^1-hard

Table 3: Definability, algebraic characterisations and circuit complexity of regular language $\boldsymbol{L}$, where $M(\boldsymbol{L})$ is the syntactic monoid and $\eta_{\boldsymbol{L}}$ the syntactic morphism of $\boldsymbol{L}$.

We are now in a position to establish the connection between the rewritability of temporal OMQs and definability of regular languages mentioned above. For any OMQ $\boldsymbol{q}$ and $\Xi \subseteq \mathsf{sig}(\boldsymbol{q})$, we regard $\Sigma_\Xi = 2^\Xi$ as an *alphabet*. Any Ξ-ABox $\mathcal{A}$ can be given as a Σ_Ξ-word $w_{\mathcal{A}} = a_0 \dots a_n$ with $a_i = \{A \mid A(i) \in \mathcal{A}\}$. Conversely, any Σ_Ξ-word $w = a_0 \dots a_n$ gives the ABox $\mathcal{A}_w$ with $\mathsf{tem}(\mathcal{A}_w) = [0, n]$ and $A(i) \in \mathcal{A}_w$ iff $A \in a_i$. The word $\emptyset$ corresponds to $\mathcal{A}_\emptyset = \emptyset$ with $\mathsf{tem}(\mathcal{A}_\emptyset) = [0, 0]$. The *language* $\boldsymbol{L}_\Xi(\boldsymbol{q})$ is defined to be the set of Σ_Ξ-words $w_{\mathcal{A}}$ with a yes-answer to $\boldsymbol{q}$ over $\mathcal{A}$. For a specific OMQ $\boldsymbol{q}(x)$, we take $\Gamma_\Xi = \Sigma_\Xi \cup \Sigma'_\Xi$ with a disjoint copy Σ'_Ξ of Σ_Ξ and represent a pair $(\mathcal{A}, i)$ with a Ξ-ABox $\mathcal{A}$ and $i \in \mathsf{tem}(\mathcal{A})$ as a Γ_Ξ-word $w_{\mathcal{A},i} = a_0 \dots a'_i \dots a_n$, where $a'_i = \{A' \mid A(i) \in \mathcal{A}\} \in \Sigma'_\Xi$ and $a_j = \{A \mid A(j) \in \mathcal{A}\} \in \Sigma_\Xi$, for $j \neq i$. The *language* $\boldsymbol{L}_\Xi(\boldsymbol{q}(x))$ is the set of Γ_Ξ-words $w_{\mathcal{A},i}$ such that i is a certain answer to $\boldsymbol{q}(x)$ over $\mathcal{A}$. The following is proved similarly to Vardi and Wolper's (1986, Theorem 2.1).

Theorem 5. *Let $\boldsymbol{q} = (\mathcal{O}, \varkappa)$ be a Boolean and $\boldsymbol{q}(x) = (\mathcal{O}, \varkappa(x))$ a specific OMQ. Then*

(i) both $\boldsymbol{L}_\Xi(\boldsymbol{q})$ and $\boldsymbol{L}_\Xi(\boldsymbol{q}(x))$ are regular languages;

(ii) for any $\mathcal{L} \in \{\mathsf{FO}(<), \mathsf{FO}(<, \equiv), \mathsf{FO}(<, \mathsf{MOD})\}$ and $\Xi \subseteq \mathsf{sig}(\boldsymbol{q})$, the OMQ $\boldsymbol{q}$ is $\mathcal{L}$-rewritable over Ξ-ABoxes iff $\boldsymbol{L}_\Xi(\boldsymbol{q})$ is $\mathcal{L}$-definable; *similarly, $\boldsymbol{q}(x)$ is $\mathcal{L}$-rewritable over Ξ-ABoxes iff $\boldsymbol{L}_\Xi(\boldsymbol{q}(x))$ is $\mathcal{L}$-definable.*

Proof. (i) Let $\mathsf{sub}_{\boldsymbol{q}}$ (or $\mathsf{sub}_{\mathcal{O}}$) be the set of temporal concepts occurring in $\boldsymbol{q}$ (respectively, $\mathcal{O}$) and their negations. A *type for* $\boldsymbol{q}$ (respectively, $\mathcal{O}$) is any maximal subset $\boldsymbol{\tau} \subseteq \mathsf{sub}_{\boldsymbol{q}}$ (respectively, $\boldsymbol{\tau} \subseteq \mathsf{sub}_{\mathcal{O}}$) *consistent* with $\mathcal{O}$ in the sense that all formulas in $\boldsymbol{\tau}$ are true at some point of a model of $\mathcal{O}$. Let $\boldsymbol{T}$ be the set of all types for $\boldsymbol{q}$. Define an NFA $\mathfrak{A}$ over Σ_Ξ whose language $\boldsymbol{L}(\mathfrak{A})$ is $\Sigma_\Xi^* \setminus \boldsymbol{L}_\Xi(\boldsymbol{q})$. Its states are $Q_{\neg\varkappa} = \{\boldsymbol{\tau} \in \boldsymbol{T} \mid \neg\varkappa \in \boldsymbol{\tau}\}$. The transition relation $\to_a$, for $a \in \Sigma_\Xi$, is defined by taking $\boldsymbol{\tau}_1 \to_a \boldsymbol{\tau}_2$ if the following conditions hold:

(a) $a \subseteq \boldsymbol{\tau}_2$,

(b) $\bigcirc_{\!F} C \in \boldsymbol{\tau}_1$ iff $C \in \boldsymbol{\tau}_2$, for every $\bigcirc_{\!F} C \in \mathsf{sub}_{\boldsymbol{q}}$,

(c) $\Box_F C \in \boldsymbol{\tau}_1$ iff $C \in \boldsymbol{\tau}_2$ and $\Box_F C \in \boldsymbol{\tau}_2$, for every $\Box_F C \in \mathsf{sub}_{\boldsymbol{q}}$,

(d) $\Diamond_F C \in \boldsymbol{\tau}_1$ iff $C \in \boldsymbol{\tau}_2$ or $\Diamond_F C \in \boldsymbol{\tau}_2$, for every $\Diamond_F C \in \mathsf{sub}_{\boldsymbol{q}}$,

and symmetrically for the past-time operators. The initial (accepting) states are those $\boldsymbol{\tau} \in Q_{\neg\varkappa}$ for which $\boldsymbol{\tau} \cup \{\Box_P \neg\varkappa\}$ (and, respectively, $\boldsymbol{\tau} \cup \{\Box_F \neg\varkappa\}$) is consistent with $\mathcal{O}$. Then $w \in \boldsymbol{L}(\mathfrak{A})$ iff $(\mathcal{O}, \mathcal{A}_w) \not\models \exists x\, \varkappa(x)$, for any $w \in \Sigma_\Xi^*$. Indeed, if $w \in \boldsymbol{L}(\mathfrak{A})$, we take an accepting run $\boldsymbol{\tau}_0, \dots, \boldsymbol{\tau}_n$ of $\mathfrak{A}$ on w, a model $\mathcal{I}^-$ of $\mathcal{O}$ with $\mathcal{I}^-, k \models \boldsymbol{\tau}_0 \cup \{\Box_P \neg\varkappa\}$, a model $\mathcal{I}^+$ of $\mathcal{O}$ with $\mathcal{I}^+, l \models \boldsymbol{\tau}_n \cup \{\Box_F \neg\varkappa\}$, for some $k, l \in \mathbb{Z}$, and construct a new interpretation $\mathcal{I}$ that has the types $\boldsymbol{\tau}_0, \dots, \boldsymbol{\tau}_n$ in the interval $[0, n]$, before (after) which it has the same types as in $\mathcal{I}^-$ in $(-\infty, k)$ (respectively, $\mathcal{I}^+$ on (l, ∞)). One can readily check that $\mathcal{I}$ is a model of $\mathcal{O}$ and $\mathcal{A}_w$ such that $\varkappa^{\mathcal{I}} = \emptyset$, and so $(\mathcal{O}, \mathcal{A}_w) \not\models \exists x\, \varkappa(x)$. The opposite direction is obvious.

To show that $\boldsymbol{L}_\Xi(\boldsymbol{q}(x))$ is regular, we observe first that the language $\boldsymbol{L}$ over Γ_Ξ comprising words of the form $w_{\mathcal{A},i}$, for all non-empty Ξ-Aboxes $\mathcal{A}$ and $i \in \mathsf{tem}(\mathcal{A})$, is regular. Thus, it suffices to define an NFA $\mathfrak{A}$ over Γ_Ξ such that $\boldsymbol{L}_\Xi(\boldsymbol{q}(x)) = \boldsymbol{L} \setminus \boldsymbol{L}(\mathfrak{A})$. The set of states in $\mathfrak{A}$ is $\boldsymbol{T} \cup \boldsymbol{T}'$ with a disjoint copy $\boldsymbol{T}'$ of $\boldsymbol{T}$. The set of initial states is $\boldsymbol{T}$ and the set of accepting states is $\boldsymbol{T}'$. The transition relation $\to_a$, for $a \in \Sigma_\Xi$, is defined by taking $\boldsymbol{\tau}_1 \to_a \boldsymbol{\tau}_2$ if either $\boldsymbol{\tau}_1, \boldsymbol{\tau}_2 \in \boldsymbol{T}$ or $\boldsymbol{\tau}_1, \boldsymbol{\tau}_2 \in \boldsymbol{T}'$ and conditions (a)–(d) are satisfied; for $a' \in \Sigma'_\Xi$, we set $\boldsymbol{\tau}_1 \to_{a'} \boldsymbol{\tau}_2$ if $\boldsymbol{\tau}_1 \in \boldsymbol{T}$, $\boldsymbol{\tau}_2 \in \boldsymbol{T}'$, $\neg\varkappa \in \boldsymbol{\tau}_2$, $a' \subseteq \boldsymbol{\tau}_2$, and (b)–(d) hold. It is easy to see that, for any Ξ-ABox $\mathcal{A}$ and $i \in \mathsf{tem}(\mathcal{A})$, there exists a model $\mathcal{I}$ of $\mathcal{O}$ and $\mathcal{A}$ with $i \notin \varkappa^{\mathcal{I}}$ iff $w_{\mathcal{A},i} \in \boldsymbol{L}(\mathfrak{A})$.

The proof of (ii) is easy and can be found in Appendix A.1. ❑

Note that the number of states in the NFAs in the proof above is $2^{O(|\boldsymbol{q}|)}$ and that they can be constructed in exponential time in the size $|\boldsymbol{q}|$ of $\boldsymbol{q}$ as *LTL*-satisfiability is in PSpace.

By Theorem 5, we can reformulate the evaluation problem for $\boldsymbol{q}$ and $\boldsymbol{q}(x)$ over Ξ-ABoxes as the *word problem* for the regular languages $\boldsymbol{L}_\Xi(\boldsymbol{q})$ and $\boldsymbol{L}_\Xi(\boldsymbol{q}(x))$. Then Table 3 yields the following correspondences between the data complexity of answering and FO-rewritability of Boolean and specific *LTL* OMQs $\boldsymbol{q}$:

$\boldsymbol{q}$ is $\mathsf{FO}(<, \equiv)$-rewritable iff it can be answered in AC^0;

$\boldsymbol{q}$ is $\mathsf{FO}(<, \mathsf{MOD})$-rewritable iff it can be answered in ACC^0;

$\boldsymbol{q}$ is not $\mathsf{FO}(<, \mathsf{MOD})$-rewritable iff answering $\boldsymbol{q}$ in NC^1-complete (unless $\mathrm{ACC}^0 = \mathrm{NC}^1$);

$\boldsymbol{q}$ is $\mathsf{FO}(<, \mathsf{RPR})$-rewritable iff it can be answered in NC^1.

3. Characterising FO-rewritability of Regular Languages

In this section, we show that the algebraic characterisations of FO-definability of $\boldsymbol{L}(\mathfrak{A})$ in Table 3 can be captured by localisable properties of the transition monoid of $\mathfrak{A}$. Note that Theorem 6 (*i*) was already observed by Stern (1985) and used in proving that $\mathsf{FO}(<)$-definability of $\boldsymbol{L}(\mathfrak{A})$ is PSPACE-complete (Stern, 1985; Cho & Huynh, 1991; Bernátsky, 1997); criteria (*ii*) and (*iii*) of $\mathsf{FO}(<, \equiv)$- and $\mathsf{FO}(<, \mathsf{MOD})$-definability are novel.

Theorem 6. *For any DFA $\mathfrak{A} = (Q, \Sigma, \delta, q_0, F)$, the following criteria hold*:

(*i*) $\boldsymbol{L}(\mathfrak{A})$ *is not* $\mathsf{FO}(<)$*-definable iff* $\mathfrak{A}$ *contains a nontrivial cycle, that is, there exist a word* $u \in \Sigma^*$, *a state* $q \in Q^r$, *and a number* $k \leq |Q|$ *such that* $q \not\sim \delta_u(q)$ *and* $q = \delta_{u^k}(q)$;

(*ii*) $\boldsymbol{L}(\mathfrak{A})$ *is not* $\mathsf{FO}(<, \equiv)$*-definable iff there are words* $u, v \in \Sigma^*$, *a state* $q \in Q^r$, *and a number* $k \leq |Q|$ *such that* $q \not\sim \delta_u(q)$, $q = \delta_{u^k}(q)$, $|v| = |u|$, *and* $\delta_{u^i}(q) = \delta_{u^i v}(q)$, *for every* $i < k$;

(*iii*) $\boldsymbol{L}(\mathfrak{A})$ *is not* $\mathsf{FO}(<, \mathsf{MOD})$*-definable iff there exist words* $u, v \in \Sigma^*$, *a state* $q \in Q^r$ *and numbers* $k, l \leq |Q|$ *such that* k *is an odd prime,* $l > 1$ *and coprime to both* 2 *and* k, $q \not\sim \delta_u(q)$, $q \not\sim \delta_v(q)$, $q \not\sim \delta_{uv}(q)$ *and, for all* $x \in \{u, v\}^*$, *we have* $\delta_x(q) \sim \delta_{xu^2}(q) \sim \delta_{xv^k}(q) \sim \delta_{x(uv)^l}(q)$.

Proof. We use the algebraic criteria of Table 3 for $\boldsymbol{L} = \boldsymbol{L}(\mathfrak{A})$. Thus, $M(\boldsymbol{L})$ is the transition monoid of the minimal DFA $\mathfrak{A}_{\boldsymbol{L}(\mathfrak{A})}$, whose transition function is denoted by $\tilde{\delta}$.

(*i*) ($\Rightarrow$) Suppose $\mathfrak{G}$ is a nontrivial group in $M(\mathfrak{A}_{\boldsymbol{L}(\mathfrak{A})})$. Let $u \in \Sigma^*$ be such that $\tilde{\delta}_u$ is a nonidentity element in $\mathfrak{G}$. We claim that there is $p \in Q^r$ such that $\tilde{\delta}_{u^n}(p/_\sim) \neq \tilde{\delta}_{u^{n+1}}(p/_\sim)$ for any $n > 0$. Indeed, otherwise for every $p \in Q^r$ there is $n_p > 0$ with $\tilde{\delta}_{u^{n_p}}(p/_\sim) = \tilde{\delta}_{u^{n_p+1}}(p/_\sim)$. Let $n = \max\{n_p \mid p \in Q^r\}$. Then $\tilde{\delta}_{u^n} = \tilde{\delta}_{u^{n+1}}$, contrary to (9). By (13), there is $m \geq 1$ with $\tilde{\delta}_{u^{2m}}(p/_\sim) = \tilde{\delta}_{u^m}(p/_\sim)$. Let $s/_\sim = \tilde{\delta}_{u^m}(p/_\sim)$. Then $s/_\sim = \tilde{\delta}_{u^m}(s/_\sim)$, and so the restriction of δ_{u^m} to the subset $s/_\sim$ of Q^r is an $s/_\sim \to s/_\sim$ function. By (14), there are $q \in s/_\sim$ and $n \geq 1$ with $(\delta_{u^m})^n(q) = q$. Thus, $\delta_{u^{mn}}(q) = q$, and so by (15), there is $k \leq |Q|$ with $\delta_{u^k}(q) = q$. As $s/_\sim \neq \tilde{\delta}_u(s/_\sim)$, we also have $q \not\sim \delta_u(q)$, as required.

(*i*) ($\Leftarrow$) Suppose the condition holds for $\mathfrak{A}$. Then there are $u \in \Sigma^*$, $q \in Q^r/_\sim$, and $k < \omega$ with $q \neq \tilde{\delta}_u(q)$ and $q = \tilde{\delta}_{u^k}(q)$. So $\tilde{\delta}_{u^n} \neq \tilde{\delta}_{u^{n+1}}$ for any $n > 0$. Indeed, otherwise we would have some $n > 0$ with $\tilde{\delta}_{u^n}(q) = \tilde{\delta}_{u^{n+1}}(q)$. Let i, j be such that $n = i \cdot k + j$ and $j < k$. Then

$$q = \tilde{\delta}_{u^k}(q) = \tilde{\delta}_{u^{(i+1)k}}(q) = \tilde{\delta}_{u^n u^{k-j}}(q) = \tilde{\delta}_{u^{n+1} u^{k-j}}(q) = \tilde{\delta}_{u^{(i+1)k} u}(q) = \tilde{\delta}_u(q).$$

So, by (11) and (12), $\mathfrak{G}_{\tilde{\delta}_u}$ is a nontrivial group in $M(\mathfrak{A}_{\boldsymbol{L}(\mathfrak{A})})$.

(*ii*) ($\Rightarrow$) Let $\mathfrak{G}$ be a nontrivial group in $\eta_{\boldsymbol{L}}(\Sigma^t)$, for some $t < \omega$, and let $u \in \Sigma^t$ be such that $\tilde{\delta}_u$ is a nonidentity element in $\mathfrak{G}$. As shown in the proof of (*i*) ($\Rightarrow$), there exist $s \in Q^r$ and $m \geq 1$ such that $s/_\sim \neq \tilde{\delta}_u(s/_\sim)$ and $s/_\sim = \tilde{\delta}_{u^m}(s/_\sim)$. Now let $v \in \Sigma^t$ be such that

$\tilde{\delta}_v$ is the identity element in $\mathfrak{G}$, and consider δ_v. By (10), there is $\ell \geq 1$ such that δ_{v^ℓ} is idempotent. Then $\delta_{v^{2\ell-1}v^{2\ell}} = \delta_{v^{2\ell-1}}$. Thus, if we let $\bar{u} = uv^{2\ell-1}$ and $\bar{v} = v^{2\ell}$, then $|\bar{u}| = |\bar{v}|$ and $\delta_{\bar{u}^i} = \delta_{\bar{u}^i\bar{v}}$ for any $i < \omega$. Also, $\tilde{\delta}_{u^i} = \tilde{\delta}_{\bar{u}^i}$ for every $i \geq 1$, and so the restriction of $\delta_{\bar{u}^m}$ to $s/_\sim$ is an $s/_\sim \to s/_\sim$ function. By (14), there exist $q \in s/_\sim$ and $n \geq 1$ such that $(\delta_{\bar{u}^m})^n(q) = q$. Thus, $\delta_{\bar{u}^{mn}}(q) = q$, and so by (15), there is some $k \leq |Q|$ with $\delta_{\bar{u}^k}(q) = q$. As $s/_\sim \neq \tilde{\delta}_u(s/_\sim) = \tilde{\delta}_{\bar{u}}(s/_\sim)$, we also have $q \not\sim \delta_{\bar{u}}(q)$, as required.

(ii) ($\Leftarrow$) If the condition holds for $\mathfrak{A}$, then there exist $u, v \in \Sigma^*$, $q \in Q^r/_\sim$, and $k < \omega$ such that $q \neq \tilde{\delta}_u(q)$, $q = \tilde{\delta}_{u^k}(q)$, $|v| = |u|$, and $\tilde{\delta}_{u^i}(q) = \tilde{\delta}_{u^i v}(q)$, for every $i < k$. As $M(\mathfrak{A}_{\boldsymbol{L}(\mathfrak{A})})$ is finite, it has finitely many subsets. So there exist $i, j \geq 1$ such that $\eta_{\boldsymbol{L}}(\Sigma^{i|u|}) = \eta_{\boldsymbol{L}}(\Sigma^{(i+j)|u|})$. Let z be a multiple of j with $i \leq z < i+j$. Then $\eta_{\boldsymbol{L}}(\Sigma^{z|u|}) = \eta_{\boldsymbol{L}}(\Sigma^{(z|u|)^2})$, and so $\eta_{\boldsymbol{L}}(\Sigma^{z|u|})$ is closed under the composition of functions (that is, the semigroup operation of $M(\mathfrak{A}_{\boldsymbol{L}(\mathfrak{A})})$). Let $w = uv^{z-1}$ and consider the group $\mathfrak{G}_{\tilde{\delta}_w}$. Then $G_{\tilde{\delta}_w} \subseteq \eta_{\boldsymbol{L}}(\Sigma^{z|u|})$. We claim that $\mathfrak{G}_{\tilde{\delta}_w}$ is nontrivial. Indeed, we have $\tilde{\delta}_w(q) = \tilde{\delta}_{uv^{z-1}}(q) = \tilde{\delta}_u(q) \neq q$. On the other hand, $\tilde{\delta}_{w^k}(q) = \tilde{\delta}_{u^k}(q) = q$. By the proof of *(i)* ($\Leftarrow$), $\mathfrak{G}_{\tilde{\delta}_w}$ is nontrivial.

(iii) ($\Rightarrow$) Suppose $\mathfrak{G}$ is an unsolvable group in $M(\mathfrak{A}_{\boldsymbol{L}(\mathfrak{A})})$. By the Kaplan–Levy criterion, $\mathfrak{G}$ contains three functions a, b, c such that $o_{\mathfrak{G}}(a) = 2$, $o_{\mathfrak{G}}(b)$ is an odd prime, $o_{\mathfrak{G}}(c) > 1$ and coprime to both 2 and $o_{\mathfrak{G}}(b)$, and $c \circ b \circ a = e_{\mathfrak{G}}$ for the identity element $e_{\mathfrak{G}}$ of $\mathfrak{G}$. Let $u, v \in \Sigma^*$ be such that $a = \tilde{\delta}_u$, $b = \tilde{\delta}_v$ and $c = (\tilde{\delta}_{uv})^-$, and let $k = o_{\mathfrak{G}}(\tilde{\delta}_v)$ and $r = o_{\mathfrak{G}}(c) = o_{\mathfrak{G}}(\tilde{\delta}_{uv})$. Then $r > 1$ and coprime to both 2 and k. Let $S = \big\{p \in Q^r/_\sim \mid e_{\mathfrak{G}}(p) = p\big\}$. As $\tilde{\delta}_x$ is $\mathfrak{G}$ for every $x \in \{u, v\}^*$, we have $e_{\mathfrak{G}} \circ \tilde{\delta}_x = \tilde{\delta}_x$. Thus,

$$\tilde{\delta}_{xu^2}(q) = \tilde{\delta}_{u^2}\big(\tilde{\delta}_x(q)\big) = e_{\mathfrak{G}}\big(\tilde{\delta}_x(q)\big) = (e_{\mathfrak{G}} \circ \tilde{\delta}_x)(q) = \tilde{\delta}_x(q), \quad \text{and}$$
$$\tilde{\delta}_{xv^k}(q) = \tilde{\delta}_{v^k}\big(\tilde{\delta}_x(q)\big) = e_{\mathfrak{G}}\big(\tilde{\delta}_x(q)\big) = (e_{\mathfrak{G}} \circ \tilde{\delta}_x)(q) = \tilde{\delta}_x(q), \quad \text{for every } q \in S.$$

Then, by (16), each of $\tilde{\delta}_u\restriction_S$, $\tilde{\delta}_v\restriction_S$ and $\tilde{\delta}_{uv}\restriction_S$ is a permutation on S. By (17), the order of $\tilde{\delta}_u\restriction_S$ is 2, the order of $\tilde{\delta}_v\restriction_S$ is k, and the order l of $\tilde{\delta}_{uv}\restriction_S$ is a > 1 divisor of r, and so it is coprime to both 2 and k. Also, we have $k, l \leq |S| \leq |Q|$. Further, for every x, if q is in S then $\tilde{\delta}_x(q) \in S$ as well. So we have

$$\tilde{\delta}_{x(uv)^l}(q) = \tilde{\delta}_{(uv)^l}\big(\tilde{\delta}_x(q)\big) = (\tilde{\delta}_{uv}\restriction_S)^l\big(\tilde{\delta}_x(q)\big) = \mathsf{id}_S\big(\tilde{\delta}_x(q)\big) = \tilde{\delta}_x(q), \quad \text{for all } q \in S.$$

It remains to show that there is $q \in S$ with $q \neq \tilde{\delta}_u(q)$, $q \neq \tilde{\delta}_u(q)$, and $q \neq \tilde{\delta}_{uv}(q)$. Recall that the length of any cycle in a permutation divides its order. First, we show there is $q \in S$ with $q \neq \tilde{\delta}_u(q)$ and $q \neq \tilde{\delta}_u(q)$. Indeed, as $\tilde{\delta}_u\restriction_S \neq \mathsf{id}_S$, there is $q \in S$ such that $\tilde{\delta}_u(q) = q' \neq q$. As the order of $\tilde{\delta}_u\restriction_S$ is 2, $\tilde{\delta}_u(q') = q$. If both $\tilde{\delta}_v(q) = q$ and $\tilde{\delta}_v(q') = q'$ were the case, then $\tilde{\delta}_{uv}(q) = q'$ and $\tilde{\delta}_{uv}(q') = q$ would hold, and so (qq') would be a cycle in $\tilde{\delta}_{uv}\restriction_S$, contrary to l being coprime to 2. So take some $q \in S$ with $\tilde{\delta}_u(q) = q' \neq q$ and $\tilde{\delta}_v(q) \neq q$. If $\tilde{\delta}_v(q') \neq q$ then $\tilde{\delta}_{uv}(q) \neq q$, and so q is a good choice. Suppose $\tilde{\delta}_v(q') = q$, and let $q'' = \tilde{\delta}_v(q)$. Then $q'' \neq q'$, as k is odd. Thus, $\tilde{\delta}_{uv}(q') \neq q'$, and so q' is a good choice.

(iii) ($\Leftarrow$) Suppose $u, v \in \Sigma^*$, $q \in Q^r$, and $k, l < \omega$ are satisfying the conditions. For every $x \in \{u, v\}^*$, we define an equivalence relation $\approx_x$ on $Q^r/_\sim$ by taking $p \approx_x p'$ iff $\tilde{\delta}_x(p) = \tilde{\delta}_x(p')$. Then we clearly have that $\approx_x \subseteq \approx_{xy}$, for all $x, y \in \{u, v\}^*$. As Q is finite, there is $z \in \{u, v\}^*$ such that $\approx_z = \approx_{zy}$ for all $y \in \{u, v\}^*$. Take such a z. By (10), $\tilde{\delta}_z^n$ is idempotent for some $n \geq 1$. We let $w = z^n$. Then $\tilde{\delta}_w$ is idempotent and we also have that

$$\approx_w = \approx_{wy} \quad \text{for all } y \in \{u, v\}^*. \tag{18}$$

Let $G_{\{u,v\}} = \{\tilde{\delta}_{wxw} \mid x \in \{u,v\}^*\}$. Then $G_{\{u,v\}}$ is closed under composition. Let $\mathfrak{G}_{\{u,v\}}$ be the subsemigroup of $M(\mathfrak{A}_{\boldsymbol{L}(\mathfrak{A})})$ with universe $G_{\{u,v\}}$. Then $\tilde{\delta}_w = \tilde{\delta}_{w\varepsilon w}$ is an identity element in $\mathfrak{G}_{\{u,v\}}$. Let $S = \{p \in Q^r/_\sim \mid \tilde{\delta}_w(p) = p\}$. We show that

$$\text{for every } \tilde{\delta} \text{ in } \mathfrak{G}_{\{u,v\}},\ \tilde{\delta}{\restriction}_S \text{ is a permutation on } S, \tag{19}$$

and so $\mathfrak{G}_{\{u,v\}}$ is a group by (16). Indeed, take some $x \in \{u,v\}^*$. As $\tilde{\delta}_w\big(\tilde{\delta}_{wxw}(p)\big) = \tilde{\delta}_{wxww}(p) = \tilde{\delta}_{wxw}(p)$, for any $p \in Q^r/_\sim$, $\tilde{\delta}_{wxw}{\restriction}_S$ is an $S \to S$ function. Also, if $p, p' \in S$ and $\tilde{\delta}_{wxw}(p) = \tilde{\delta}_{wxw}(p')$ then $p \approx_{wxw} p'$. Thus, by (18), $p \approx_w p'$, that is, $p = \tilde{\delta}_w(p) = \tilde{\delta}_w(p') = p'$, proving (19).

We show that $\mathfrak{G}_{\{u,v\}}$ is unsolvable by finding an unsolvable homomorphic image of it. Let $R = \{p \in Q^r/_\sim \mid p = \tilde{\delta}_x(q) \text{ for some } x \in \{u,v\}^*\}$. We claim that, for every $\tilde{\delta}$ in $\mathfrak{G}_{\{u,v\}}$, $\tilde{\delta}{\restriction}_R$ is a permutation on R, and so the function h mapping every $\tilde{\delta}$ to $\tilde{\delta}{\restriction}_R$ is a group homomorphism from $\mathfrak{G}_{\{u,v\}}$ to the group of all permutations on R. Indeed, by (19), it is enough to show that $R \subseteq S$. Let $\overline{w} = \overline{z}_m \dots \overline{z}_1$, where $w = z_1 \dots z_m$ for some $z_i \in \{u,v\}$, $\overline{u} = u$ and $\overline{v} = v^{k-1}$. Since $\tilde{\delta}_x(q) = \tilde{\delta}_{x(u)^2}(q) = \tilde{\delta}_{x(v)^k}(q)$ for all $x \in \{u,v\}^*$, we obtain that

$$\begin{aligned}\tilde{\delta}_{yw\overline{w}}(q) = \tilde{\delta}_{\overline{z}_{m-1}\dots\overline{z}_1}\big(\tilde{\delta}_{yz_1\dots z_m\overline{z}_m}(q)\big) = \tilde{\delta}_{\overline{z}_{m-1}\dots\overline{z}_1}\big(\tilde{\delta}_{yz_1\dots z_{m-1}}(q)\big) = \dots \\ \dots = \tilde{\delta}_{\overline{z}_1}\big(\tilde{\delta}_{yz_1}(q)\big) = \tilde{\delta}_{xz_1\overline{z}_1}(q) = \tilde{\delta}_y(q), \quad \text{for all } y \in \{u,v\}^*.\end{aligned} \tag{20}$$

Now suppose $p \in R$, that is, $p = \tilde{\delta}_x(q)$ for some $x \in \{u,v\}^*$. Then, by (20),

$$\tilde{\delta}_w(p) = \tilde{\delta}_w\big(\tilde{\delta}_x(q)\big) = \tilde{\delta}_{xw}(q) = \tilde{\delta}_{xww\overline{w}}(q) = \tilde{\delta}_{xw\overline{w}}(q) = \tilde{\delta}_x(q) = p,$$

and so $p \in S$, as required.

Now let $\mathfrak{G}$ be the image of $\mathfrak{G}_{\{u,v\}}$ under h. We prove that $\mathfrak{G}$ is unsolvable by finding three elements a, b, c in it such that $o_{\mathfrak{G}}(a) = 2$, $o_{\mathfrak{G}}(b) = k$, $o_{\mathfrak{G}}(c)$ is coprime to both 2 and $o_{\mathfrak{G}}(b)$, and $c \circ b \circ a = \mathsf{id}_R$ (the identity element of $\mathfrak{G}$). So let $a = h(\tilde{\delta}_{wuw})$, $b = h(\tilde{\delta}_{wvw})$, and $c = h(\tilde{\delta}_{wuvw})^-$. Observe that, for every $x \in \{u,v\}^*$, $h(\tilde{\delta}_{wxw}) = \tilde{\delta}_x{\restriction}_R$, and so $c \circ b \circ a = \mathsf{id}_R$. Also, for any $\tilde{\delta}_x(q) \in R$, $a^2\big(\tilde{\delta}_x(q)\big) = (\tilde{\delta}_u{\restriction}_R)^2\big(\tilde{\delta}_x(q)\big) = \tilde{\delta}_{xu^2}(q) = \tilde{\delta}_x(q)$ by our assumption, so $a^2 = \mathsf{id}_R$. On the other hand, $q \in R$ as $\tilde{\delta}_\varepsilon(q) = q$, and $\mathsf{id}_R(q) = q \neq \tilde{\delta}_u(q)$ by assumption, so $a \neq \mathsf{id}_R$. As $o_{\mathfrak{G}}(a)$ divides 2, $o_{\mathfrak{G}}(a) = 2$ follows. Similarly, we can show that $o_{\mathfrak{G}}(b) = k$ (using that $\tilde{\delta}_{xv^k}(q) = \tilde{\delta}_x(q)$ for every $x \in \{u,v\}^*$, and $u \neq \tilde{\delta}_v(q)$). Finally (using that $\tilde{\delta}_{x(uv)^l}(q) = \tilde{\delta}_x(q)$ for every $x \in \{u,v\}^*$, and $u \neq \tilde{\delta}_{uv}(q)$), we obtain that $h(\tilde{\delta}_{wuvw})^l = \mathsf{id}_R$ and $h(\tilde{\delta}_{wuvw}) \neq \mathsf{id}_R$. Therefore, it follows that $o_{\mathfrak{G}}(c) = o_{\mathfrak{G}}\big(h(\tilde{\delta}_{wuvw})^-\big) = o_{\mathfrak{G}}\big(h(\tilde{\delta}_{wuvw})\big) > 1$ and divides l, and so coprime to both 2 and k, as required. ❑

The following technical observation will be used in Sections 6 and 7; its proof is given in Appendix A.2.

Lemma 7. *Suppose $\mathcal{L} \in \{\mathsf{FO}(<), \mathsf{FO}(<,\equiv), \mathsf{FO}(<,\mathsf{MOD})\}$ and Σ, Γ and Δ are alphabets such that $\Sigma \cup \{x,y\} \subseteq \Gamma \subseteq \Delta$, for some $x, y \notin \Sigma$. Then a regular language $\boldsymbol{L}$ over Σ is $\mathcal{L}$-definable iff the regular language $\boldsymbol{L}' = \{w_1xwyw_2 \mid w \in \boldsymbol{L},\ w_1, w_2 \in \Gamma^*\}$ is $\mathcal{L}$-definable over Δ.*

4. Deciding FO-definability of Regular Languages: PSpace-hardness

Kozen (1977) showed that deciding non-emptiness of the intersection of the languages recognised by a set of given deterministic DFAs is PSPACE-complete. By carefully analysing Kozen's lower bound proof and using the criterion of Theorem 6 (i), Cho and Huynh (1991) established that deciding $\mathsf{FO}(<)$-definability of $\boldsymbol{L}(\mathfrak{A})$, for any given minimal DFA $\mathfrak{A}$, is PSPACE-hard. We generalise their construction and use the criteria in Theorem 6 (ii)–(iii) to cover $\mathsf{FO}(<,\equiv)$- and $\mathsf{FO}(<,\mathsf{MOD})$-definability as well.

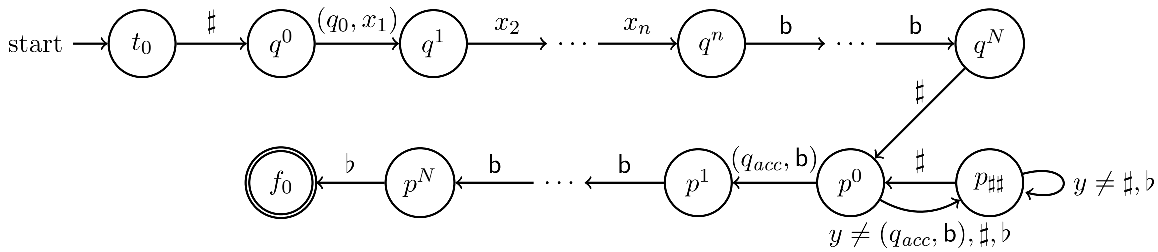
Theorem 8. *For any $\mathcal{L} \in \{\mathsf{FO}(<), \mathsf{FO}(<,\equiv), \mathsf{FO}(<,\mathsf{MOD})\}$, deciding $\mathcal{L}$-definability of the language $\boldsymbol{L}(\mathfrak{A})$ of a given minimal DFA $\mathfrak{A}$ is* PSPACE*-hard.*

Proof. Let $\boldsymbol{M}$ be a deterministic Turing machine that decides a language using at most $N = P_{\boldsymbol{M}}(n)$ tape cells on any input of size n, for some polynomial $P_{\boldsymbol{M}}$. Given such an $\boldsymbol{M}$ and an input $\boldsymbol{x}$, our aim is to define three minimal DFAs whose languages are, respectively, $\mathsf{FO}(<)$-, $\mathsf{FO}(<,\equiv)$-, and $\mathsf{FO}(<,\mathsf{MOD})$-definable iff $\boldsymbol{M}$ rejects $\boldsymbol{x}$, and whose sizes are polynomial in N and the size $|\boldsymbol{M}|$ of $\boldsymbol{M}$.

Suppose $\boldsymbol{M} = (Q, \Gamma, \gamma, \mathsf{b}, q_0, q_{acc})$ with a set Q of states, tape alphabet Γ with b for blank, transition function γ, initial state q_0 and accepting state q_{acc}. Without loss of generality we assume that $\boldsymbol{M}$ erases the tape before accepting, its head is at the left-most cell in an accepting configuration, and if $\boldsymbol{M}$ does not accept the input, it runs forever. Given an input word $\boldsymbol{x} = x_1 \dots x_n$ over Γ, we represent configurations $\mathfrak{c}$ of the computation of $\boldsymbol{M}$ on $\boldsymbol{x}$ by the N-long word written on the tape (with sufficiently many blanks at the end) in which the symbol y in the active cell is replaced by the pair (q, y) for the current state q. The accepting computation of $\boldsymbol{M}$ on $\boldsymbol{x}$ is encoded by a word $\sharp\, \mathfrak{c}_1 \,\sharp\, \mathfrak{c}_2 \,\sharp\, \dots \,\sharp\, \mathfrak{c}_{k-1} \,\sharp\, \mathfrak{c}_k \flat$ over the alphabet $\Sigma = \Gamma \cup (Q \times \Gamma) \cup \{\sharp, \flat\}$, with $\mathfrak{c}_1, \mathfrak{c}_2, \dots, \mathfrak{c}_k$ being the subsequent configurations. In particular, $\mathfrak{c}_1$ is the initial configuration on $\boldsymbol{x}$ (so it is of the form $(q_0, x_1)x_2 \dots x_n \mathsf{b} \dots \mathsf{b}$), and $\mathfrak{c}_k$ is the accepting configuration (so it is of the form $(q_{acc}, \mathsf{b})\mathsf{b} \dots \mathsf{b}$). As usual for this representation of computations, we may regard γ as a partial function from $\big(\Gamma \cup (Q \times \Gamma) \cup \{\sharp\}\big)^3$ to $\Gamma \cup (Q \times \Gamma)$ with $\gamma(\sigma^j_{i-1}, \sigma^j_i, \sigma^j_{i+1}) = \sigma^{j+1}_i$ for each $j < k$, where σ^j_i is the ith symbol of $\mathfrak{c}^j$.

Let $p_{\boldsymbol{M},\boldsymbol{x}} = p$ be the first prime such that $p \geq N+2$ and $p \not\equiv \pm 1 \pmod{10}$. By Corollary 1.6 of Bennett, Martin, O'Bryant, and Rechnitzer (2018), p is polynomial in N. Our first aim is to define a $p+1$-long sequence of disjoint minimal DFAs $\mathfrak{A}_i$ over Σ. Each $\mathfrak{A}_i$ has size polynomial in N, $|\boldsymbol{M}|$, and is constructible in logarithmic space; it checks certain properties of an accepting computation on $\boldsymbol{x}$ such that $\boldsymbol{M}$ accepts $\boldsymbol{x}$ iff the intersection of the $\boldsymbol{L}(\mathfrak{A}_i)$ is not empty and consists of the single word encoding the accepting computation on $\boldsymbol{x}$.

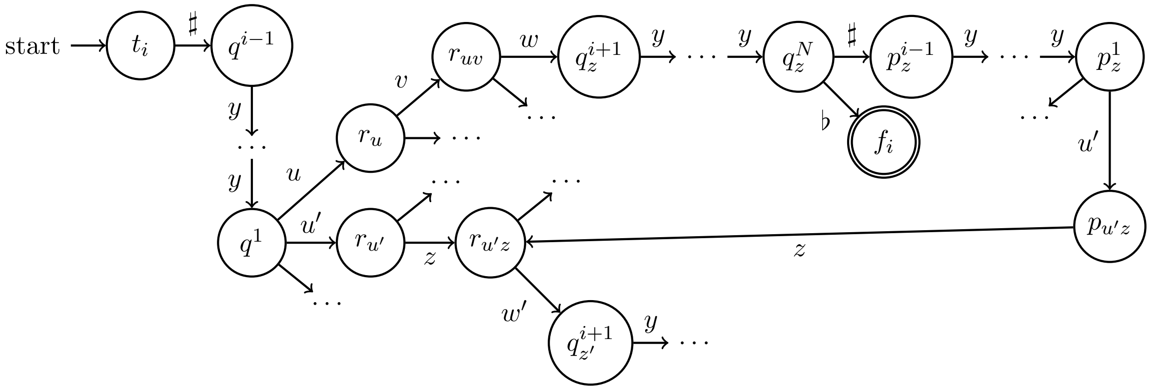
Formally, we define each $\mathfrak{A}_i$ as an NFA but bear in mind that it can standardly be turned to a DFA by adding to it a 'trash state' tr_i looping on itself with every character $\sigma \in \Sigma$, and also adding the missing transitions that all lead to the trash state tr_i. The DFA $\mathfrak{A}_0$ checks that an input starts with the initial configuration on $\boldsymbol{x}$ and ends with the accepting configuration:



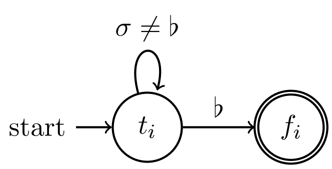
When $1 \le i \le N$, the DFA $\mathfrak{A}_i$ checks, for all $j < k$, whether the ith symbol of $\mathfrak{c}^j$ changes 'according to γ' in passing to $\mathfrak{c}^{j+1}$. The non-trash part of its transition function δ^i is as follows, for $1 < i < N$. (For $i = 1$ and $i = N$, some adjustments are needed.) For all $u, u', v, w, w', y, z \in \Gamma \cup (Q \times \Gamma)$,

$$\delta^i_\sharp(t_i) = q^{i-1}, \quad \delta^i_u(q^j) = q^{j-1}, \text{ for } 2 \le j \le i-1, \quad \delta^i_u(q^1) = r_u, \quad \delta^i_v(r_u) = r_{uv},$$
$$\delta^i_w(r_{uv}) = q^{i+1}_{\gamma(u,v,w)}, \quad \delta^i_y(q^j_z) = q^{j+1}_z, \text{ for } i+1 \le j \le N, \quad \delta^i_\sharp(q^N_z) = p^{i-1}_z$$
$$\delta^i_y(p^j_z) = p^{j-1}_z, \text{ for } 2 \le j \le i-1, \ \delta^i_\flat(q^N_z) = f_i, \quad \delta^i_{u'}(q^1_z) = p_{u'z}, \quad \delta^i_z(p_{u'z}) = r_{u'z};$$

see below, where $z = \gamma(u, v, w)$ and $z' = \gamma(u', z, w')$:



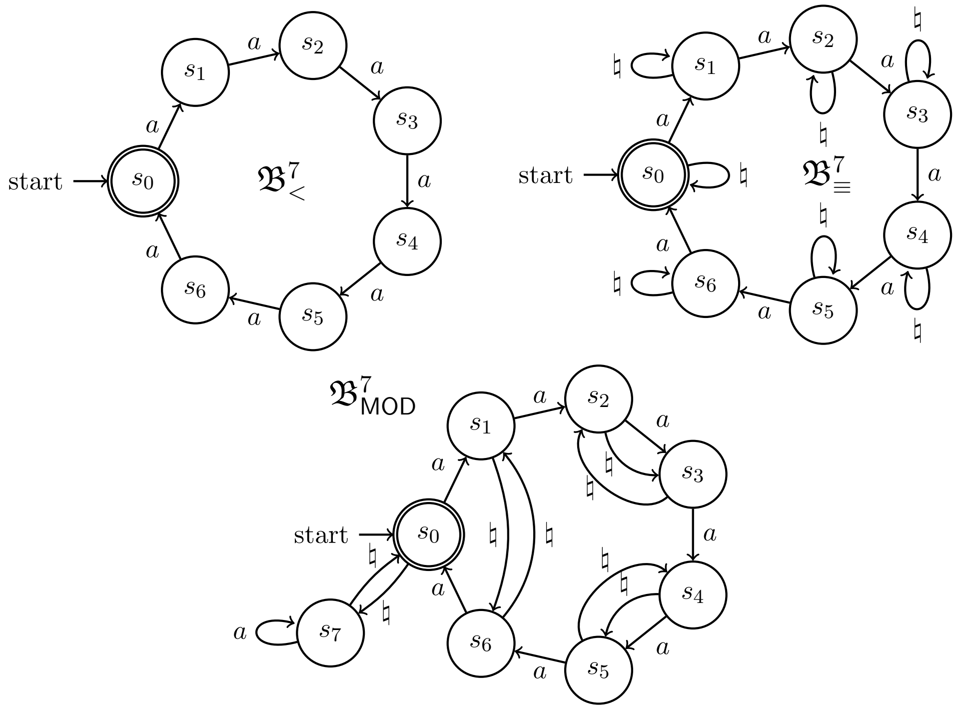
Finally, if $N + 1 \le i \le p$ then $\mathfrak{A}_i$ accepts all words over Σ with a single occurrence of $\flat$, which is the input's last character:



Note that $\mathfrak{A}_{p-1} = \mathfrak{A}_p$ as $p \ge N + 2$. It is not hard to check that each $\mathfrak{A}_i$ is a minimal DFA that does not contain nontrivial cycles and the following holds:

Lemma 9. *$\boldsymbol{M}$ accepts $\boldsymbol{x}$ iff $\bigcap_{i=0}^{p} \boldsymbol{L}(\mathfrak{A}_i) \neq \emptyset$, in which case this language consists of a single word that encodes the accepting computation of $\boldsymbol{M}$ on $\boldsymbol{x}$.*

Next, we require three sequences of DFAs $\mathfrak{B}^p_<$, $\mathfrak{B}^p_\equiv$ and $\mathfrak{B}^p_{\mathsf{MOD}}$, where $p > 5$ is a prime number with $p \not\equiv \pm 1 \pmod{10}$; see the picture below for $p = 7$:



In general, the first sequence is $\mathfrak{B}^p_< = \big(\{s_i \mid i < p\}, \{a\}, \delta^{\mathfrak{B}^p_<}, s_0, \{s_0\}\big)$, where $\delta_a^{\mathfrak{B}^p_<}(s_i) = s_j$ if $i, j < p$ and $j \equiv i+1 \pmod p$. Then $\boldsymbol{L}(\mathfrak{B}^p_<)$ comprises all words of the form $(a^p)^*$, $\mathfrak{B}^p_<$ is the minimal DFA for $\boldsymbol{L}(\mathfrak{B}^p_<)$, and the syntactic monoid $M(\mathfrak{B}^p_<)$ is the cyclic group of order p (generated by the permutation $\delta_a^{\mathfrak{B}^p_<}$).

The second sequence is $\mathfrak{B}^p_\equiv = \big(\{s_i \mid i < p\}, \{a, \natural\}, \delta^{\mathfrak{B}^p_\equiv}, s_0, \{s_0\}\big)$, where $\delta_\natural^{\mathfrak{B}^p_\equiv}(s_i) = s_i$ and $\delta_a^{\mathfrak{B}^p_\equiv}(s_i) = s_j$ if $i, j < p$ and $j \equiv i+1 \pmod p$. One can check that $\boldsymbol{L}(\mathfrak{B}^p_\equiv)$ comprises all words of a's and $\natural$'s where the number of a's is divisible by p, $\mathfrak{B}^p_\equiv$ is the minimal DFA for $\boldsymbol{L}(\mathfrak{B}^p_\equiv)$, and $M(\mathfrak{B}^p_\equiv)$ is the cyclic group of order p (generated by the permutation $\delta_a^{\mathfrak{B}^p_\equiv}$).

The third sequence is $\mathfrak{B}^p_{\mathsf{MOD}} = \big(\{s_i \mid i \leq p\}, \{a, \natural\}, \delta^{\mathfrak{B}^p_{\mathsf{MOD}}}, s_0, \{s_0\}\big)$, where

– $\delta_a^{\mathfrak{B}^p_{\mathsf{MOD}}}(s_p) = s_p$, and $\delta_a^{\mathfrak{B}^p_{\mathsf{MOD}}}(s_i) = s_j$ if $i, j < p$ and $j \equiv i+1 \pmod p$;

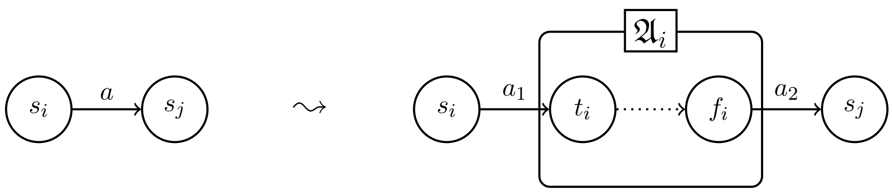
– $\delta_\natural^{\mathfrak{B}^p_{\mathsf{MOD}}}(s_0) = s_p$, $\delta_\natural^{\mathfrak{B}^p_{\mathsf{MOD}}}(s_p) = s_0$, and $\delta_\natural^{\mathfrak{B}^p_{\mathsf{MOD}}}(s_i) = s_j$ whenever $1 \leq i, j < p$ and $i \cdot j \equiv p-1 \pmod p$, that is, $j = -1/i$ in the finite field $\mathbb{F}_p$.

One can check that $\mathfrak{B}^p_{\mathsf{MOD}}$ is the minimal DFA for its language, and the syntactic monoid $M(\mathfrak{B}^p_{\mathsf{MOD}})$ is the permutation group generated by $\delta_a^{\mathfrak{B}^p_{\mathsf{MOD}}}$ and $\delta_\natural^{\mathfrak{B}^p_{\mathsf{MOD}}}$.

Lemma 10. *For any prime $p > 5$ with $p \not\equiv \pm 1 \pmod{10}$, the group $M(\mathfrak{B}^p_{\mathsf{MOD}})$ is unsolvable, but all of its proper subgroups are solvable.*

Proof. It is readily seen that the order of the permutation $\delta_\natural^{\mathfrak{B}^p_{\mathsf{MOD}}}$ is 2, that of $\delta_a^{\mathfrak{B}^p_{\mathsf{MOD}}}$ is p, while the order of the inverse of $\delta_{\natural a}^{\mathfrak{B}^p_{\mathsf{MOD}}}$ is the same as the order of $\delta_{\natural a}^{\mathfrak{B}^p_{\mathsf{MOD}}}$, which is 3. So $M(\mathfrak{B}^p_{\mathsf{MOD}})$ is unsolvable, for any prime p, by the Kaplan–Levy criterion. To prove that all proper subgroups of $M(\mathfrak{B}^p_{\mathsf{MOD}})$ are solvable, we show that $M(\mathfrak{B}^p_{\mathsf{MOD}})$ is a subgroup of the *projective* *special linear group* $\mathrm{PSL}_2(p)$. If p is a prime with $p > 5$ and $p \not\equiv \pm 1 \pmod{10}$, then all proper subgroups of $\mathrm{PSL}_2(p)$ are solvable (e.g., King, 2005, Theorem 2.1). (So $M(\mathfrak{B}^p_{\mathsf{MOD}})$ is in fact isomorphic to the unsolvable group $\mathrm{PSL}_2(p)$.) Consider the set $P = \{0, 1, \dots, p-1, \infty\}$ of all points of the projective line over the field $\mathbb{F}_p$. By identifying s_i with i for $i < p$, and s_p with ∞, we may regard the elements of $M(\mathfrak{B}^p_{\mathsf{MOD}})$ as $P \to P$ functions. The group $\mathrm{PSL}_2(p)$ consists of all $P \to P$ functions of the form $i \mapsto \frac{w \cdot i + x}{y \cdot i + z}$, where $w \cdot z - x \cdot y = 1$, with the field arithmetic of $\mathbb{F}_p$ extended by $i + \infty = \infty$ for any $i \in P$, $0 \cdot \infty = 1$ and $i \cdot \infty = \infty$ for $i \neq 0$. The two generators of $M(\mathfrak{B}^p_{\mathsf{MOD}})$ are in $\mathrm{PSL}_2(p)$: take $w = 1$, $x = 1$, $y = 0$, $z = 1$ for $\delta_a^{\mathfrak{B}^p_{\mathsf{MOD}}}$, and $w = 0$, $x = 1$, $y = p-1$, $z = 0$ for $\delta_\natural^{\mathfrak{B}^p_{\mathsf{MOD}}}$. ❑

Finally, we define automata $\mathfrak{A}_<$, $\mathfrak{A}_\equiv$, $\mathfrak{A}_{\mathsf{MOD}}$ over the tape alphabet $\Sigma_+ = \Sigma \cup \{a_1, a_2, \natural\}$, where a_1, a_2 are fresh symbols. We take, respectively, $\mathfrak{B}^p_<$, $\mathfrak{B}^p_\equiv$, $\mathfrak{B}^p_{\mathsf{MOD}}$ and replace each transition $s_i \to_a s_j$ in them by a fresh copy of $\mathfrak{A}_i$, for $i \leq p$, as shown in the picture below:



We make $\mathfrak{A}_<$, $\mathfrak{A}_\equiv$, $\mathfrak{A}_{\mathsf{MOD}}$ deterministic by adding a trash state *tr* looping on itself with every $y \in \Sigma_+$, and adding the missing transitions leading to *tr*. It follows that $\mathfrak{A}_<$, $\mathfrak{A}_\equiv$, $\mathfrak{A}_{\mathsf{MOD}}$ are minimal DFAs of size polynomial in N and $|\boldsymbol{M}|$, which can clearly be constructed in logarithmic space.

Lemma 11. *(i)* $\boldsymbol{L}(\mathfrak{A}_<)$ *is* $\mathsf{FO}(<)$*-definable iff* $\bigcap_{i=0}^p \boldsymbol{L}(\mathfrak{A}_i) = \emptyset$.
(ii) $\boldsymbol{L}(\mathfrak{A}_\equiv)$ *is* $\mathsf{FO}(<, \equiv)$*-definable iff* $\bigcap_{i=0}^p \boldsymbol{L}(\mathfrak{A}_i) = \emptyset$.
(iii) $\boldsymbol{L}(\mathfrak{A}_{\mathsf{MOD}})$ *is* $\mathsf{FO}(<, \mathsf{MOD})$*-definable iff* $\bigcap_{i=0}^p \boldsymbol{L}(\mathfrak{A}_i) = \emptyset$.

Proof. As $\mathfrak{A}_<, \mathfrak{A}_\equiv, \mathfrak{A}_{\mathsf{MOD}}$ are minimal, we can replace $\sim$ by $=$ in the conditions of Theorem 6. For the ($\Rightarrow$) directions, given some $w \in \bigcap_{i=0}^p \boldsymbol{L}(\mathfrak{A}_i)$, in each case we show how to satisfy the corresponding condition of Theorem 6: (i) take $u = a_1 w a_2$, $q = s_0$, and $k = p$; (ii) take $u = a_1 w a_2$, $v = \natural^{|u|}$, $q = s_0$, and $k = p$; (iii) take $u = \natural$, $v = a_1 w a_2$, $q = s_0$, $k = p$ and $l = 3$.

($\Leftarrow$) We show that the corresponding condition of Theorem 6 implies non-emptiness of $\bigcap_{i=0}^p \boldsymbol{L}(\mathfrak{A}_i)$. To this end, we define a $\Sigma_+^* \to \{a, \natural\}^*$ homomorphism by taking $h(\natural) = \natural$, $h(a_1) = a$, and $h(b) = \varepsilon$ for all other $b \in \Sigma_+$.

(i) and (ii): Let $\circ \in \{<, \equiv\}$ and suppose q is a state in $\mathfrak{A}^p_\circ$ and $u' \in \Sigma_+^*$ such that $q \neq \delta_{u'}^{\mathfrak{A}^p_\circ}(q)$ and $q = \delta_{(u')^k}^{\mathfrak{A}^p_\circ}(q)$ for some k. Let $S = \{s_0, s_1, \dots, s_{p-1}\}$. We claim that there exist $s \in S$ and $u \in \Sigma_+^*$ such that

$$s \neq \delta_u^{\mathfrak{A}^p_\circ}(s), \tag{21}$$

$$\delta_x^{\mathfrak{A}^p_\circ}(s) \in S, \quad \text{for every } x \in \{u\}^*. \tag{22}$$

Indeed, observe that none of the states along the cyclic $q \to_{(u')^k} q$ path Π in $\mathfrak{A}^p_\circ$ is *tr*. So there is some state along Π that is in S, as otherwise one of the $\mathfrak{A}_i$ would contain a nontrivial cycle. Therefore, u' must be of the form $w\natural^n a_1 w'$ for some $w \in \Sigma^*$, $n < \omega$ and $w' \in \Sigma_+^*$. It is easy to see that $s = \delta_{(u')^{k-1}w}^{\mathfrak{A}_\circ^p}(q)$ and $u = \natural^n a_1 w' w$ is as required in (21) and (22).

As $M(\mathfrak{B}_\circ^p)$ is a finite group, $\{\delta_{h(x)}^{\mathfrak{B}_\circ^p} \mid x \in \{u\}^*\}$ forms a subgroup $\mathfrak{G}$ in it (the subgroup generated by $\delta_{h(u)}^{\mathfrak{B}_\circ^p}$). We show that $\mathfrak{G}$ is nontrivial by finding its nontrivial homomorphic image. By (22), for any $x \in \{u\}^*$, the restriction $\delta_x^{\mathfrak{A}_\circ^p}\restriction_{S'}$ of $\delta_x^{\mathfrak{A}_\circ^p}$ to $S' = \{\delta_y^{\mathfrak{A}_\circ^p}(s) \mid y \in \{u\}^*\}$ is an $S' \to S'$ function and $\delta_x^{\mathfrak{A}_\circ^p}\restriction_{S'} = \delta_{h(x)}^{\mathfrak{B}_\circ^p}\restriction_{S'}$. As $M(\mathfrak{B}_\circ^p)$ is a group of permutations on a set containing S', $\delta_{h(x)}^{\mathfrak{B}_\circ^p}\restriction_{S'}$ is a permutation of S', for every $x \in \{u\}^*$. Thus, $\{\delta_{h(x)}^{\mathfrak{B}_\circ^p}\restriction_{S'} \mid x \in \{u\}^*\}$ is a homomorphic image of $\mathfrak{G}$ that is nontrivial by (21).

As $\mathfrak{G}$ is a nontrivial subgroup of the cyclic group $M(\mathfrak{B}_\circ^p)$ of order p and p is a prime, $\mathfrak{G} = M(\mathfrak{B}_\circ^p)$. Then there is $x \in \{u\}^*$ with $\delta_{h(x)}^{\mathfrak{B}_\circ^p} = \delta_a^{\mathfrak{B}_\circ^p}$ (a permutation containing the p-cycle $(s_0 s_1 \dots s_{p-1})$ 'around' all elements of S), and so $S' = S$ and $x = \natural^n a_1 w a_2 w'$ for some $n < \omega$, $w \in \Sigma^*$, and $w' \in \Sigma_+^*$. As $n = 0$ when $\circ =<$ and $\delta_{\natural^n}^{\mathfrak{A}_\equiv^p}(s)$ for every $s \in S$, $S' = S$ implies that $w \in \bigcap_{i=0}^{p-1} \boldsymbol{L}(\mathfrak{A}_i) = \bigcap_{i=0}^{p} \boldsymbol{L}(\mathfrak{A}_i)$.

(iii) Suppose q is a state in $\mathfrak{A}_{\mathsf{MOD}}^p$ and $u', v' \in \Sigma_+^*$ such that $q \neq \delta_{u'}^{\mathfrak{A}_{\mathsf{MOD}}^p}(q)$, $q \neq \delta_{v'}^{\mathfrak{A}_{\mathsf{MOD}}^p}(q)$, $q \neq \delta_{u'v'}^{\mathfrak{A}_{\mathsf{MOD}}^p}(q)$, and $\delta_x^{\mathfrak{A}_{\mathsf{MOD}}^p}(q) = \delta_{x(u')^2}^{\mathfrak{A}_{\mathsf{MOD}}^p}(q) = \delta_{x(v')^k}^{\mathfrak{A}_{\mathsf{MOD}}^p}(q) = \delta_{x(u'v')^l}^{\mathfrak{A}_{\mathsf{MOD}}^p}(q)$ for some odd prime k and number l that is coprime to both 2 and k. Take $S = \{s_0, s_1, \dots, s_p\}$. We claim that there exist $s \in S$ and $u, v \in \Sigma_+^*$ such that

$$s \neq \delta_u^{\mathfrak{A}_{\mathsf{MOD}}^p}(s),\ \ s \neq \delta_v^{\mathfrak{A}_{\mathsf{MOD}}^p}(s),\ \ s \neq \delta_{uv}^{\mathfrak{A}_{\mathsf{MOD}}^p}(s), \tag{23}$$

$$\delta_x^{\mathfrak{A}_{\mathsf{MOD}}^p}(s) \in S, \quad \text{for every } x \in \{u, v\}^*, \tag{24}$$

$$\delta_x^{\mathfrak{A}_{\mathsf{MOD}}^p}(s) = \delta_{xu^2}^{\mathfrak{A}_{\mathsf{MOD}}^p}(s) = \delta_{xv^k}^{\mathfrak{A}_{\mathsf{MOD}}^p}(s) = \delta_{x(uv)^l}^{\mathfrak{A}_{\mathsf{MOD}}^p}(s), \quad \text{for every } x \in \{u, v\}^*. \tag{25}$$

Indeed, by an argument similar to the one in the proof of *(i)* and *(ii)* above, we must have $u' = w_u \natural^n a_1 w'_u$ and $v' = w_v \natural^m a_1 w'_v$ for some $w_u, w_v \in \Sigma^*$, $n, m < \omega$ and $w'_u, w'_v \in \Sigma_+^*$. For every $x \in \{u, v\}^*$, as both $\delta_{xw_u}^{\mathfrak{A}_{\mathsf{MOD}}^p}(q)$ and $\delta_{xw_v}^{\mathfrak{A}_{\mathsf{MOD}}^p}(q)$ are in S, they must be the same state. Using this it is not hard to see that $s = \delta_{u'w_u}^{\mathfrak{A}_{\mathsf{MOD}}^p}(q)$, $u = \natural^n a_1 w'_u w_u$ and $v = \natural^m a_1 w'_v w_v$ are as required in (23)–(25).

As $M(\mathfrak{B}_{\mathsf{MOD}}^p)$ is a finite group, the set $\{\delta_{h(x)}^{\mathfrak{B}_{\mathsf{MOD}}^p} \mid x \in \{u, v\}^*\}$ forms a subgroup $\mathfrak{G}$ in it (the subgroup generated by $\delta_{h(u)}^{\mathfrak{B}_{\mathsf{MOD}}^p}$ and $\delta_{h(v)}^{\mathfrak{B}_{\mathsf{MOD}}^p}$). We show that $\mathfrak{G}$ is unsolvable by finding an unsolvable homomorphic image of it. To this end, we let $S' = \{\delta_y^{\mathfrak{A}_{\mathsf{MOD}}^p}(s) \mid y \in \{u, v\}^*\}$. Then (24) implies that $S' \subseteq S$ and

$$\delta_{h(x)}^{\mathfrak{B}_{\mathsf{MOD}}^p}(s') = \delta_x^{\mathfrak{A}_{\mathsf{MOD}}^p}(s') \in S', \quad \text{for all } s' \in S \text{ and } x \in \{u, v\}^*, \tag{26}$$

and so the restriction $\delta_x^{\mathfrak{A}_{\mathsf{MOD}}^p}\restriction_{S'}$ of $\delta_x^{\mathfrak{A}_{\mathsf{MOD}}^p}$ to S' is an $S' \to S'$ function and $\delta_x^{\mathfrak{A}_{\mathsf{MOD}}^p}\restriction_{S'} = \delta_{h(x)}^{\mathfrak{B}_{\mathsf{MOD}}^p}\restriction_{S'}$. As $M(\mathfrak{B}_{\mathsf{MOD}}^p)$ is a group of permutations on a set containing S', $\delta_{h(x)}^{\mathfrak{B}_{\mathsf{MOD}}^p}\restriction_{S'}$ is a permutation of S', for any $x \in \{u, v\}^*$. It follows that $\{\delta_{h(x)}^{\mathfrak{B}_{\mathsf{MOD}}^p}\restriction_{S'} \mid x \in \{u, v\}^*\}$ is a homomorphic image of $\mathfrak{G}$, which is unsolvable by the Kaplan–Levy criterion: by (23), (25), and 2 and k being primes, the order of the permutation $\delta_{h(u)}^{\mathfrak{B}^p_{\mathsf{MOD}}}\lceil_{S'}$ is 2, the order of $\delta_{h(v)}^{\mathfrak{B}^p_{\mathsf{MOD}}}\lceil_{S'}$ is k, and the order of $\delta_{h(uv)}^{\mathfrak{B}^p_{\mathsf{MOD}}}\lceil_{S'}$ (which is the same as the order of its inverse) is a > 1 divisor of l, and so coprime to both 2 and k.

As $\mathfrak{G}$ is an unsolvable subgroup of $M(\mathfrak{B}^p_{\mathsf{MOD}})$, Lemma 10 implies that $\mathfrak{G} = M(\mathfrak{B}^p_{\mathsf{MOD}})$, so $\{u,v\}^* \not\subseteq \natural^*$. We claim that $S' = S$. Indeed, let $x \in \{u,v\}^*$ be such that $\delta_{h(x)}^{\mathfrak{B}^p_{\mathsf{MOD}}} = \delta_{a}^{\mathfrak{B}^p_{\mathsf{MOD}}}$. As $|S'| \geq 2$ by (23), $s \in \{s_0, \dots, s_{p-1}\}$, and so $\{s_0, \dots, s_{p-1}\} \subseteq S'$ follows by (26). As there is $y \in \{u,v\}^*$ with $\delta_{h(y)}^{\mathfrak{B}^p_{\mathsf{MOD}}} = \delta_{\natural}^{\mathfrak{B}^p_{\mathsf{MOD}}}$, $s_p \in S'$ also follows by (26). Finally, as $\{u,v\}^* \not\subseteq \natural^*$, there is $x \in \{u,v\}^*$ of the form $\natural^n a_1 w a_2 w'$, for some $n < \omega$, $w \in \Sigma$ and $w' \in \Sigma_+^*$. As $S' = S$, $\delta_x^{\mathfrak{B}^p_{\mathsf{MOD}}}(s_i) \in S$ for every $i \leq p$, and so $w \in \bigcap_{i=0}^p \boldsymbol{L}(\mathfrak{A}_i)$. ❑

Now Theorem 8 clearly follows from Lemmas 9 and 11. ❑

5. Deciding FO-definability of 2NFAs in PSpace

Using the criterion of Theorem 6 (i), Stern (1985) showed that deciding whether the language of any given DFA is $\mathsf{FO}(<)$-definable can be done in PSPACE. In this section, we also apply the criteria of Theorem 6 to provide PSPACE-algorithms deciding whether the language of any given 2NFA is $\mathcal{L}$-definable, for $\mathcal{L} \in \{\mathsf{FO}(<), \mathsf{FO}(<,\equiv), \mathsf{FO}(<,\mathsf{MOD})\}$.

Let $\mathfrak{A} = (Q, \Sigma, \delta, Q_0, F)$ be a 2NFA. Similarly to Carton and Dartois (2015), we first construct an exponential-size DFA $\mathfrak{A}'$ with $\boldsymbol{L}(\mathfrak{A}) = \boldsymbol{L}(\mathfrak{A}')$. To this end, for any $w \in \Sigma^+$, we introduce four binary relations $\mathsf{b}_{lr}(w)$, $\mathsf{b}_{rl}(w)$, $\mathsf{b}_{rr}(w)$, and $\mathsf{b}_{ll}(w)$ on Q describing the *left-to-right*, *right-to-left*, *right-to-right*, and *left-to-left behaviour of* $\mathfrak{A}$ *on* w. Namely,

- $(q,q') \in \mathsf{b}_{lr}(w)$ if there is a run of $\mathfrak{A}$ on w from $(q,0)$ to $(q',|w|)$;
- $(q,q') \in \mathsf{b}_{rr}(w)$ if there is a run of $\mathfrak{A}$ on w from $(q,|w|-1)$ to $(q',|w|)$;
- $(q,q') \in \mathsf{b}_{rl}(w)$ if, for some $a \in \Sigma$, there is a run on aw from $(q,|aw|-1)$ to $(q',0)$ such that no $(q'',0)$ occurs in it before $(q',0)$;
- $(q,q') \in \mathsf{b}_{ll}(w)$ if, for some $a \in \Sigma$, there is a run on aw from $(q,1)$ to $(q',0)$ such that no $(q'',0)$ occurs in it before $(q',0)$.

For $w = \varepsilon$ (the empty word), we define the $\mathsf{b}_{ij}(w)$ as the identity relation on Q.

Example 12. For the 2NFA $\mathfrak{A}$ over $\Sigma = \{a,b\}$ shown in Figure 1, we have:

$$\mathsf{b}_{lr}(ab) = \{(q_0,s),(s,q),(t,q),(w,q),(y,p)\}, \quad \mathsf{b}_{rl}(ab) = \{(v,u),(u,h)\},$$
$$\mathsf{b}_{rr}(ab) = \{(r,s),(u,y),(v,q),(z,p)\}, \quad \mathsf{b}_{ll}(ab) = \{(s,u),(t,u),(w,u)\}. \qquad \dashv$$

Now, let $\mathsf{b} = (\mathsf{b}_{lr}, \mathsf{b}_{rl}, \mathsf{b}_{rr}, \mathsf{b}_{ll})$, where the b_{ij} are the behaviours of $\mathfrak{A}$ on some $w \in \Sigma^*$, in which case we can also write $\mathsf{b}(w)$, and let $\mathsf{b}' = \mathsf{b}(w')$, for some $w' \in \Sigma^*$. We define the composition $\mathsf{b} \cdot \mathsf{b}' = \mathsf{b}''$ with components b''_{ij} as follows. Let X and Y be the reflexive and transitive closures of the relations $\mathsf{b}'_{ll} \circ \mathsf{b}_{rr}$ and $\mathsf{b}_{rr} \circ \mathsf{b}'_{ll}$ on Q, respectively. Then we set:

$$\mathsf{b}''_{lr} = \mathsf{b}_{lr} \circ X \circ \mathsf{b}'_{lr}, \qquad \mathsf{b}''_{rl} = \mathsf{b}'_{rl} \circ Y \circ \mathsf{b}_{rl},$$
$$\mathsf{b}''_{rr} = \mathsf{b}'_{rr} \cup \mathsf{b}'_{rl} \circ Y \circ \mathsf{b}_{rr} \circ \mathsf{b}'_{lr}, \qquad \mathsf{b}''_{ll} = \mathsf{b}_{ll} \cup \mathsf{b}_{lr} \circ X \circ \mathsf{b}'_{ll} \circ \mathsf{b}_{rl}.$$

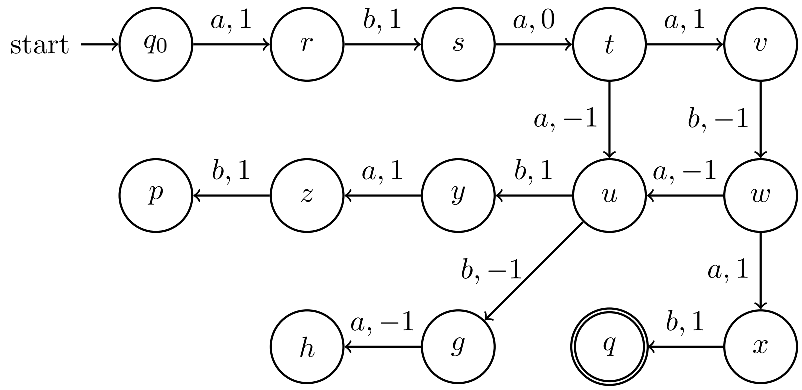


Figure 1: The 2NFA $\mathfrak{A}$ for Example 12.

One can check that $\mathsf{b}'' = \mathsf{b}(ww')$.

Example 13. Consider again the 2NFA $\mathfrak{A}$ from Example 12, where we computed $\mathsf{b}(ab)$. One can readily check that $\mathsf{b}(ab) \cdot \mathsf{b}(ab) = (\mathsf{b}_{lr}, \mathsf{b}_{rl}, \mathsf{b}_{rr}, \mathsf{b}_{ll})$, where

$$\mathsf{b}_{lr} = \mathsf{b}_{lr}(ab) \circ (\{(s,y),(t,y),(w,y)\} \cup \{(q,q) \mid q \in Q\}) \circ \mathsf{b}_{lr}(ab) = \{(q_0,q),(q_0,p)\},$$
$$\mathsf{b}_{rl} = \{(v,h)\},$$
$$\mathsf{b}_{rr} = \mathsf{b}_{rr}(ab) \cup \mathsf{b}_{rl}(ab) \circ (\{(r,u)\} \cup \{(q,q) \mid q \in Q\}) \circ \mathsf{b}_{rr}(ab) \circ \mathsf{b}_{lr}(ab) = \mathsf{b}_{rr}(ab) \cup \{(v,p)\},$$
$$\mathsf{b}_{ll} = \mathsf{b}_{ll}(ab) \cup \mathsf{b}_{lr}(ab) \circ (\{(s,y),(t,y),(w,y)\} \cup \{(q,q) \mid q \in Q\}) \circ \mathsf{b}_{ll}(ab) \circ \mathsf{b}_{rl}(ab) = \mathsf{b}_{ll}(ab) \cup \{(q_0,h)).$$

Clearly, $\mathsf{b}(ab) \cdot \mathsf{b}(ab)$ coincides with $\mathsf{b}(abab) = (\mathsf{b}_{lr}, \mathsf{b}_{rl}, \mathsf{b}_{rr}, \mathsf{b}_{ll})$, where $\mathsf{b}_{lr} = \{(q_0,p),(q_0,q)\}$, $\mathsf{b}_{rl} = \{(v,h)\}$, $\mathsf{b}_{rr} = \{(r,s),(u,y),(v,q),(v,p)\}$ and $\mathsf{b}_{ll} = \{(q_0,h),(s,u),(t,u),(w,u)\}$; see the picture above. ⊣

Define a DFA $\mathfrak{A}' = (Q', \Sigma, \delta', q_0', F')$ by taking

$$Q' = \big\{(B_{lr}, B_{rr}) \mid B_{lr} \subseteq Q_0 \times Q,\ B_{rr} \subseteq Q \times Q\big\}, \quad q_0' = \big(\big\{(q,q) \mid q \in Q_0\big\}, \emptyset\big),$$
$$F' = \big\{(B_{lr}, B_{rr}) \mid (q_0,q) \in B_{lr}, \text{ for some } q_0 \in Q_0 \text{ and } q \in F\big\},$$
$$\delta_a'\big((B_{lr}, B_{rr})\big) = (B_{lr}', B_{rr}'), \text{ with } B_{lr}' = B_{lr} \circ X(a) \circ \mathsf{b}_{lr}(a),$$
$$B_{rr}' = \mathsf{b}_{rr}(a) \cup \mathsf{b}_{rl}(a) \circ Y(a) \circ B_{rr} \circ \mathsf{b}_{lr}(a),$$

where $X(a)$ and $Y(a)$ are the reflexive and transitive closures of $\mathsf{b}_{ll}(a) \circ B_{rr}$ and $B_{rr} \circ \mathsf{b}_{ll}(a)$ respectively.

Example 14. We illustrate the construction of the DFA $\mathfrak{A}'$ using the 2NFA $\mathfrak{A}$ from Example 12. We have $q_0' = (\{(q_0,q_0)\}, \emptyset)$ and

$$\delta_a'(q_0') = (\{(q_0,r)\}, \{(q_0,r),(s,v),(t,v),(w,x),(y,z)\}) = q_1',$$
$$\delta_b'(q_1') = (\{(q_0,s)\}, \{(r,s),(u,y),(x,q),(z,p)\} \cup \{(v,q)\}) = q_2',$$
$$\delta_a'(q_2') = (\{(q_0,z),(q_0,v)\}, \{(q_0,r),(s,v),(t,v),(w,x),(y,z)\} \cup \{(s,z),(w,z),(t,z)\}) = q_3',$$
$$\delta_b'(q_3') = (\{(q_0,q),(q_0,p)\}, \{(r,s),(u,y),(x,q),(z,p)\} \cup \{(v,q),(v,p)\}) = q_4'.$$

Note that $q_4' \in F'$. $\dashv$

Returning to our general construction, we observe that, for any $w \in \Sigma^*$,

$$\begin{aligned}\delta'_w\big((B_{lr}, B_{rr})\big) = (B'_{lr}, B'_{rr}) \text{ iff } B'_{lr} &= B_{lr} \circ X(w) \circ \mathsf{b}_{lr}(w) \text{ and}\\ B'_{rr} &= \mathsf{b}_{rr}(w) \cup \mathsf{b}_{rl}(w) \circ Y(w) \circ B_{rr} \circ \mathsf{b}_{lr}(w),\end{aligned} \tag{27}$$

where $X(w)$ and $Y(w)$ are the reflexive and transitive closures of $\mathsf{b}_{ll}(w) \circ B_{rr}$ and $B_{rr} \circ \mathsf{b}_{ll}(w)$. (To illustrate, for $\mathfrak{A}'$ in Example 14, we have just shown that $\delta'_{abab}(q_0') = (B'_{lr}, B'_{rr}) = q_4'$, and (B'_{lr}, B'_{rr}) can be computed by applying (27) to q_0' and $\mathsf{b}(abab)$ defined in Example 13.) Similarly to Shepherdson (1959), Vardi (1989) one can show that

$$\boldsymbol{L}(\mathfrak{A}) = \boldsymbol{L}(\mathfrak{A}'). \tag{28}$$

Intuitively, $q_0' \to_w (B_{lr}, B_{rr})$ in $\mathfrak{A}'$ and $q \in B_{lr}$ iff there exists a (two-way) run of $\mathfrak{A}$ on w from $(q_0, 0)$ to $(q, |w|)$ on w.

Next, we prove that, even though the size of $\mathfrak{A}'$ is exponential in $\mathfrak{A}$, we can still use Theorem 6 to decide $\mathcal{L}$-definability of $\boldsymbol{L}(\mathfrak{A})$ in PSpace:

Theorem 15. *For $\mathcal{L} \in \{\mathsf{FO}(<), \mathsf{FO}(<, \equiv), \mathsf{FO}(<, \mathsf{MOD})\}$, deciding $\mathcal{L}$-definability of $\boldsymbol{L}(\mathfrak{A})$, for any 2NFA $\mathfrak{A}$, can be done in* PSpace.

Proof. Let $\mathfrak{A}'$ be the DFA defined above for the given 2NFA $\mathfrak{A}$. By Theorem 6 (i) and (28), $\boldsymbol{L}(\mathfrak{A})$ is not $\mathsf{FO}(<)$-definable iff there exist a word $u \in \Sigma^*$, a reachable state $q \in Q'$, and a number $k \leq |Q'|$ such that $q \not\sim \delta'_u(q)$ and $q = \delta'_{u^k}(q)$. We guess the required k in binary, q and a quadruple $\mathsf{b}(u)$ of binary relations on Q. Clearly, they all can be stored in polynomial space in $|\mathfrak{A}|$. To check that our guesses are correct, we first check that $\mathsf{b}(u)$ indeed corresponds to some $u \in \Sigma^*$. This is done by guessing a sequence $\mathsf{b}_0, \dots, \mathsf{b}_n$ of distinct quadruples of binary relations on Q such that $\mathsf{b}_0 = \mathsf{b}(u_0)$ and $\mathsf{b}_{i+1} = \mathsf{b}_i \cdot \mathsf{b}(u_{i+1})$, for some $u_0, \dots, u_n \in \Sigma$. (Any sequence with a subsequence starting after b_i and ending with b_{i+m}, for some i and m such that $\mathsf{b}_i = \mathsf{b}_{i+m}$, is equivalent, in the context of this proof, to the sequence with such a subsequence removed.) Thus, we can assume that $n \leq 2^{O(|Q|)}$, and so n can be guessed in binary and stored in PSpace. So the stage of our algorithm checking that $\mathsf{b}(u)$ corresponds to some $u \in \Sigma^*$ makes n iterations and continues to the next stage if $\mathsf{b}_n = \mathsf{b}(u)$ or terminates with an answer no otherwise. Now, using $\mathsf{b}(u)$, we compute $\mathsf{b}(u^k)$ by means of a sequence $\mathsf{b}_0, \dots, \mathsf{b}_k$, where $\mathsf{b}_0 = \mathsf{b}(u)$ and $\mathsf{b}_{i+1} = \mathsf{b}_i \cdot \mathsf{b}(u)$. With $\mathsf{b}(u)$ ($\mathsf{b}(u^k)$), we compute $\delta'_u(q)$ (respectively, $\delta'_{u^k}(q)$) in PSpace using (27). If $\delta'_{u^k}(q) \neq q$, the algorithm terminates with an answer no. Otherwise, in the final stage of the algorithm, we check that $\delta'_u(q) \not\sim q$. This is done by guessing $v \in \Sigma^*$ such that $\delta'_v(q) = q_1$, $\delta'_v\big(\delta'_u(q)\big) = q_2$, and $q_1 \in F'$ iff $q_1 \notin F'$. We guess such v (if any) in the form of $\mathsf{b}(v)$ using an algorithm analogous to that for guessing u.

By Theorem 6 (ii) and (28), $\boldsymbol{L}(\mathfrak{A})$ is not $\mathsf{FO}(<, \equiv)$-definable iff there there exist words $u, v \in \Sigma^*$, a reachable state $q \in Q'$, and a number $k \leq |Q'|$ such that $q \not\sim \delta'_u(q)$, $q = \delta'_{u^k}(q)$, $|v| = |u|$, and $\delta'_{u^i}(q) = \delta'_{u^i v}(q)$, for all $i < k$. We outline how to modify the algorithm for $\mathsf{FO}(<)$ above to check $\mathsf{FO}(<, \equiv)$-definability. First, we need to guess and check v in the form of $\mathsf{b}(v)$ in parallel with guessing and checking u in the form of $\mathsf{b}(u)$, making sure that $|v| = |u|$. For that, we guess a sequence of distinct pairs $(\mathsf{b}_0, \mathsf{b}_0'), \dots, (\mathsf{b}_n, \mathsf{b}_n')$ such that the

b_i are as above, $\mathsf{b}'_0 = \mathsf{b}(v_0)$ and $\mathsf{b}'_{i+1} = \mathsf{b}'_i \cdot \mathsf{b}(v_{i+1})$, for some $v_0, \dots, v_n \in \Sigma$. (Any such sequence with a subsequence starting after $(\mathsf{b}_i, \mathsf{b}'_i)$ and ending with $(\mathsf{b}_{i+m}, \mathsf{b}'_{i+m})$, for some i and m such that $(\mathsf{b}_i, \mathsf{b}'_i) = (\mathsf{b}_{i+m}, \mathsf{b}'_{i+m})$, is equivalent to the sequence with that subsequence removed.) So $n \leq 2^{O(|Q|)}$. For each $i < k$, we can then compute $\delta'_{u^i}(q)$ and $\delta'_{u^i v}(q)$, using (27), and check whether whether they are equal.

Finally, by Theorem 6 (*iii*) and (28), $\boldsymbol{L}(\mathfrak{A})$ is not $\mathsf{FO}(<, \mathsf{MOD})$-definable iff there exist $u, v \in \Sigma^*$, a reachable state $q \in Q'$ and $k, l \leq |Q'|$ such that k is an odd prime, $l > 1$ and coprime to both 2 and k, $q \not\sim \delta'_u(q)$, $q \not\sim \delta'_v(q)$, $q \not\sim \delta'_{uv}(q)$, and $\delta'_x(q) \sim \delta'_{xu^2}(q) \sim \delta'_{xv^k}(q) \sim \delta'_{x(uv)^l}(q)$, for all $x \in \{u, v\}^*$. We start by guessing $u, v \in \Sigma^*$ in the form of $\mathsf{b}(u)$ and $\mathsf{b}(v)$, respectively. Also, we guess k and l in binary and check that k is an odd prime and l is coprime to both 2 and k. By (27), δ'_x is determined by $\mathsf{b}(x)$, for any $x \in \{u, v\}^*$. Thus, to check that u, v, k, l are as required, we perform the following steps, for *each* quadruple b of binary relations on Q. First, we check whether $\mathsf{b} = \mathsf{b}(x)$, for some $x \in \{u, v\}^*$ (we discuss the algorithm for this below). If this is not the case, we construct the *next* quadruple b' and process it as b above. If it is the case, we compute all the states $\delta'_x(q)$, $\delta'_{xu^2}(q)$, $\delta'_{xv^k}(q)$, $\delta'_{x(uv)^l}(q)$, $\delta'_u(q)$, $\delta'_v(q)$, $\delta'_{uv}(q)$, and check their required (non)equivalences with respect to $\sim$, using the same method as for checking $\delta'_u(q) \not\sim q$ above. If they do not hold, our algorithm terminates with an answer no. Otherwise, we construct the *next* quadruple b' and process it as this b. When all possible quadruples b of binary relations of Q have been processed, the algorithm terminates with an answer yes.

Now, to check that a given quadruple b is equal to $\mathsf{b}(x)$, for some $x \in \{u, v\}^*$, we simply guess a sequence $\mathsf{b}_0, \dots, \mathsf{b}_n$ of quadruples of binary relations on Q such that $\mathsf{b}_0 = \mathsf{b}(w_0)$, $\mathsf{b}_n = \mathsf{b}$ and $\mathsf{b}_{i+1} = \mathsf{b}_i \cdot \mathsf{b}(w_{i+1})$, where $w_i \in \{u, v\}$. It follows from the argument above that it is enough to take $n \leq 2^{O(|Q|)}$. ❑

6. Deciding FO-rewritability of *LTL* OMQs

In this section, we use the results obtained above to establish the complexity of recognising the rewritability type of an arbitrary $\mathit{LTL}^{\Box\bigcirc}_{\mathit{bool}}$ OMQ.

Theorem 16. *For any $\mathcal{L} \in \{\mathsf{FO}(<), \mathsf{FO}(<, \equiv), \mathsf{FO}(<, \mathsf{MOD})\}$, deciding $\mathcal{L}$-rewritability of (Boolean and specific) $\mathit{LTL}^{\Box\bigcirc}_{\mathit{bool}}$ OMQs over Ξ-ABoxes is* ExpSpace*-complete. The lower bound holds already for $\mathit{LTL}^{\bigcirc}_{\mathit{horn}}$ OMAQs.*

Proof. The upper bound follows from Theorem 15 and the proof of Theorem 5. We now establish the matching lower bound for $\mathit{LTL}^{\bigcirc}_{\mathit{horn}}$ OMAQs. We only consider specific OMAQs, leaving the easier case of Boolean OMAQs to the reader. (In fact, ExpSpace-hardness for Boolean OMAQs follows from ExpSpace-hardness for specific OMAQs by Lemma 20 and Proposition 21 (*i*) to be proved in the next section). With this in mind, we first show how one can store and compute numerical values of polynomial length using $\mathit{LTL}^{\bigcirc}_{\mathit{horn}}$-ontologies.

A *counter* is a set $\mathbb{A} = \{A^i_j \mid i = 0, 1,\ j = 1, \dots, k\}$ of atomic concepts that will be used to store values between 0 and $2^k - 1$, which can be different at different time points. The counter $\mathbb{A}$ is *well-defined* at a time point $n \in \mathbb{Z}$ in an interpretation $\mathcal{I}$ if $\mathcal{I}, n \models A^0_j \land A^1_j \to \bot$ and $\mathcal{I}, n \models A^0_j \lor A^1_j$, for any $j = 1, \dots, k$. In this case, the *value of* $\mathbb{A}$ at n in $\mathcal{I}$ is given by the unique binary number $b_k \dots b_1$ for which $\mathcal{I}, n \models A^{b_1}_1 \land \dots \land A^{b_k}_k$. We require the following formulas, for $c = b_k \dots b_1$ and a well-defined counter $\mathbb{A}$:

– $[\mathbb{A} = c] = A_1^{b_1} \wedge \dots \wedge A_k^{b_k}$ with $\mathcal{I}, n \models [\mathbb{A} = c]$ iff the value of $\mathbb{A}$ is c;

– $[\mathbb{A} < c] = \displaystyle\bigvee_{\substack{k \geq i \geq 1 \\ b_i = 1}} \big(A_i^0 \wedge \bigwedge_{j=i+1}^{k} A_j^{b_j}\big)$ with $\mathcal{I}, n \models [\mathbb{A} < c]$ iff the value of $\mathbb{A}$ is $< c$;

– $[\mathbb{A} > c] = \displaystyle\bigvee_{\substack{k \geq i \geq 1 \\ b_i = 0}} \big(A_i^1 \wedge \bigwedge_{j=i+1}^{k} A_j^{b_j}\big)$ with $\mathcal{I}, n \models [\mathbb{A} > c]$ iff the value of $\mathbb{A}$ is $> c$.

We regard the set $(\bigcirc_F \mathbb{A}) = \{\bigcirc_F A_j^i \mid i = 0, 1,\ j = 1, \dots, k\}$ as another counter that stores at n in $\mathcal{I}$ the value stored by $\mathbb{A}$ at $n+1$ in $\mathcal{I}$. This allows us to use formulas such as $[\mathbb{A} > c_1] \to [(\bigcirc_F \mathbb{A}) = c_2]$, which says that if the value of $\mathbb{A}$ at n in $\mathcal{I}$ is greater than c_1, then the value of $\mathbb{A}$ at $n+1$ in $\mathcal{I}$ is c_2.

Given two counters $\mathbb{A}$ and $\mathbb{B}$, we set

$$[\mathbb{A} = \mathbb{B}] = \bigwedge_{j=1}^{k} \big((B_j^0 \to A_j^0) \wedge (B_j^1 \to A_j^1)\big),$$

$$[\mathbb{A} = \mathbb{B} + 1] = \bigwedge_{i=1}^{k} \big((B_i^0 \wedge B_{i-1}^1 \wedge \dots \wedge B_1^1 \to A_i^1 \wedge A_{i-1}^0 \wedge \dots \wedge A_1^0) \wedge \bigwedge_{j<i} ((B_i^0 \wedge B_j^0 \to A_i^0) \wedge (B_i^1 \wedge B_j^0 \to A_i^1))\big).$$

Then $\mathcal{I}, n \models [\mathbb{A} = \mathbb{B}]$ iff the values of $\mathbb{A}$ and $\mathbb{B}$ at n in $\mathcal{I}$ coincide, and $\mathcal{I}, n \models [\mathbb{A} = \mathbb{B} + 1]$ iff the value of $\mathbb{A}$ at n is equal to the value of $\mathbb{B}$ at n plus one. In a similar way, we define the formula $[\mathbb{A} = \mathbb{B} - 1]$.

Consider a deterministic Turing machine $\boldsymbol{M}$ with exponential space bound, which behaves as described in the proof of Theorem 8. Given an input word $\boldsymbol{x} = x_1 \dots x_n$, let N be the number of tape cells needed for the computation of $\boldsymbol{M}$ on $\boldsymbol{x}$, and let p be the first prime such that $p \geq N + 2$ and $p \not\equiv \pm 1 \pmod{10}$. Our aim is to construct $LTL_{horn}^{\bigcirc}$ ontologies $\mathcal{O}_<$, $\mathcal{O}_\equiv$ and $\mathcal{O}_{\mathsf{MOD}}$ of polynomial size that simulate the exponential-size, $O(p)$, DFAs $\mathfrak{A}_<$, $\mathfrak{A}_\equiv$ and $\mathfrak{A}_{\mathsf{MOD}}$ from the proof of Theorem 8, whose languages are $\mathcal{L}$-definable (for the corresponding $\mathcal{L}$) iff $\boldsymbol{M}$ rejects $\boldsymbol{x}$. The polynomial size of the ontologies can be achieved due to the repetitive structure of the automata $\mathfrak{A}_<$, $\mathfrak{A}_\equiv$ and $\mathfrak{A}_{\mathsf{MOD}}$ as we can capture an exponential number of transitions by using only polynomially-many axioms.

First we define $\mathcal{O}_<$. Let $k = \lceil \log_2 p \rceil + 1$. The ontology $\mathcal{O}_<$ uses the following atomic concepts: the symbols in $\Sigma' = \Gamma \cup (Q \times \Gamma) \cup \{\sharp, \flat, a_1, a_2\}$ (see the proof of Theorem 8) and additional symbols S, T, Q, P, Q_a, R_a, R_{ab}, P_a, P_{ab}, for $a, b \in \Sigma'$, F, X, Y, and F_{end}. We also use counters $\mathbb{A}$ and $\mathbb{L}$ with atomic concepts A_j^i and L_j^i, for $i = 0, 1$, $j = 1, \dots, k$. Set $\Xi = \Sigma' \cup \{X, Y\}$.

In the DFA $\mathfrak{A}_i$ from the proof Theorem 8, we represent

– the state t_i as $[\mathbb{A} = i] \wedge T$;

– each state q^j of $\mathfrak{A}_i$ as $[\mathbb{A} = i] \wedge Q \wedge [\mathbb{L} = j]$;

– each state q_a^j of $\mathfrak{A}_i$ as $[\mathbb{A} = i] \wedge Q_a \wedge [\mathbb{L} = j]$;
– each state p^j of $\mathfrak{A}_0$ as $[\mathbb{A} = 0] \wedge P \wedge [\mathbb{L} = j]$;
– each state p_a^j of $\mathfrak{A}_i$ as $[\mathbb{A} = i] \wedge P_a \wedge [\mathbb{L} = j]$;
– each state p_{ab} of $\mathfrak{A}_i$ as $[\mathbb{A} = i] \wedge P_{ab}$;
– each state r_a of $\mathfrak{A}_i$ as $[\mathbb{A} = i] \wedge R_a$;
– each state r_{ab} of $\mathfrak{A}_i$ as $[\mathbb{A} = i] \wedge R_{ab}$;
– f_i as $[\mathbb{A} = i] \wedge F$.

We refer to these formulas and also $[\mathbb{A} = i] \wedge S$ representing s_i in $\mathfrak{A}_<$ as *state formulas.*

The ontology $\mathcal{O}_<$ simulating $\mathfrak{A}_<$ consists of the following axioms, which are equivalent to polynomially-many $LTL^{\bigcirc}_{horn}$ axioms (see Lemma 17):

– $a \wedge b \to \bot$, for distinct $a, b \in \Xi$; $(\star_1)$

– $X \to [(\bigcirc_F \mathbb{A}) = 0] \wedge \bigcirc_F S$ to simulate the initial state of $\mathfrak{A}_<$; $(\star_2)$

– $[\mathbb{A} = 0] \wedge S \wedge Y \to F_{end}$ to simulate the accepting state of $\mathfrak{A}_<$; $(\star_3)$

– the axioms

$$\begin{aligned}
&[\mathbb{A} < p] \wedge S \wedge a_1 \to [(\bigcirc_F \mathbb{A}) = \mathbb{A}] \wedge \bigcirc_F T \wedge [(\bigcirc_F \mathbb{L}) = \mathbb{A}],\\
&[\mathbb{A} < p-1] \wedge F \wedge a_2 \to [(\bigcirc_F \mathbb{A}) = \mathbb{A}+1] \wedge \bigcirc_F S,\\
&[\mathbb{A} = p-1] \wedge F \wedge a_2 \to [(\bigcirc_F \mathbb{A}) = 0] \wedge \bigcirc_F S;
\end{aligned}$$

describing the behaviour of $\mathfrak{A}_<$ in states s_i and f_i;

– the axioms describing the transitions of $\mathfrak{A}_i$, $0 \le i \le N$, that are given in Appendix A.3;

– and the following axioms for $a \neq \flat$:

$$\begin{aligned}
&[\mathbb{A} > N] \wedge [\mathbb{A} < p] \wedge T \wedge a \to [(\bigcirc_F \mathbb{A}) = \mathbb{A}] \wedge \bigcirc_F T,\\
&[\mathbb{A} > N] \wedge [\mathbb{A} < p] \wedge T \wedge \flat \to [(\bigcirc_F \mathbb{A}) = \mathbb{A}] \wedge \bigcirc_F F
\end{aligned}$$

simulating the transitions of $\mathfrak{A}_i$, for $N < i < p$.

Next, we define the ontology $\mathcal{O}_\equiv$ by adding to $\mathcal{O}_<$ the axiom

$$[\mathbb{A} < p] \wedge S \wedge \natural \to [(\bigcirc_F \mathbb{A}) = \mathbb{A}] \wedge \bigcirc_F S$$

simulating the $\natural$-transitions in $\mathfrak{A}_\equiv$. We also extend Ξ with $\natural$.

To define $\mathcal{O}_{\mathsf{MOD}}$, more work is needed. First, we extend $\mathcal{O}_<$ with

– the following axioms regarding $\mathfrak{A}_p$:

$$\begin{aligned}
&[\mathbb{A} = p] \wedge S \wedge a_1 \to [(\bigcirc_F \mathbb{A}) = p] \wedge \bigcirc_F T,\\
&[\mathbb{A} = p] \wedge F \wedge a_2 \to [(\bigcirc_F \mathbb{A}) = p] \wedge \bigcirc_F S,
\end{aligned}$$

– and the following axioms handling $\natural$:

$$[\mathbb{A} = 0] \wedge S \wedge \natural \to [(\bigcirc_F \mathbb{A}) = p] \wedge \bigcirc_F S,$$
$$[\mathbb{A} = p] \wedge S \wedge \natural \to [(\bigcirc_F \mathbb{A}) = 0] \wedge S,$$
$$[\mathbb{A} > 0] \wedge [\mathbb{A} < p] \wedge S \wedge \natural \to [(\bigcirc_F \mathbb{A}) = \mathbb{J}] \wedge \bigcirc_F S.$$

Here, $\mathbb{J}$ is a new counter that stores the value $j = -1/i$ in the field $\mathbb{F}_p$, which is required to make sure that, for $i \neq 0, p$, we have

$$\mathcal{O}_{\mathsf{MOD}} \models [\mathbb{A} = i] \wedge S \wedge \natural \to [(\bigcirc_F \mathbb{A}) = j] \wedge \bigcirc_F S.$$

We achieve this as follows. We compute the number r such that $ir = 1(\operatorname{mod} N')$ using the following modified version of Penk's algorithm (e.g., Knuth, 1998, Exercise 4.5.2.39). The algorithm starts with $u = p$, $v = i$, $r = 0$, $s = 1$. In the course of the algorithm, u and v decrease, with the following conditions being met: $\mathrm{GCD}(u, v) = 1$, $u = ri \,(\operatorname{mod} p)$, and $v = si \,(\operatorname{mod} p)$. The algorithm repeats the following steps until $v = 0$:

- if v is even, replace it with $v/2$, and replace s with either $s/2$ or $(s + p)/2$, whichever is a whole number;
- if u is even, replace it with $u/2$, and replace r with either $r/2$ or $(r + p)/2$, whichever is a whole number;
- if u, v are odd and $u > v$, replace u with $(u - v)/2$ and r with either $(r - s)/2$ or $(r - s + p)/2$, whichever is a whole number;
- if u, v are odd and $v \geq u$, replace v with $(v - u)/2$ and s with either $(s - r)/2$ or $(s - r + p)/2$, whichever is a whole number.

The binary length of the larger of u and v is reduced by at least one bit, guaranteeing that the procedure terminates in at most $2k$ iterations while maintaining the conditions. At termination, $v = 0$ as otherwise a reduction is still possible. If $u = 1$, we get $1 = ri \,(\operatorname{mod} p)$ and $r = 1/i$ in the field $\mathbb{F}_p$, so we can set $j = p - r$.

To compute the value of j, we need to halve the number in a counter, compare two counters (using an additional counter), add and subtract (using extra counters for carries). This can be done by means of $O(k)$ counters (a fixed number of counters per $O(k)$ steps of the algorithm) with polynomially-many additional axioms. So we compute j when required and store it in counter $\mathbb{J}$. Appendix A.3 provides a full list of counters and axioms we need.

For $\mathcal{L} \in \{\mathsf{FO}(<), \mathsf{FO}(<, \equiv), \mathsf{FO}(<, \mathsf{MOD})\}$, we use $\mathfrak{A}_{\mathcal{L}}$ and $\mathcal{O}_{\mathcal{L}}$ to denote the corresponding automaton and ontology defined above. Observe that, by the proof of Theorem 8,

$$\boldsymbol{L}(\mathfrak{A}_{\mathcal{L}}) \text{ is } \mathcal{L}\text{-definable} \quad \text{iff} \quad \boldsymbol{M} \text{ rejects } \boldsymbol{x}. \tag{29}$$

The connection between $\mathfrak{A}_{\mathcal{L}}$ and $\mathcal{O}_{\mathcal{L}}$ is explained by the following lemma.

Lemma 17. *Let $\mathcal{A}$ be a Ξ-ABox and let Ψ be a state formula. Then*

(*i*) *$\mathcal{A}$ is inconsistent with $\mathcal{O}_{\mathcal{L}}$ iff there is i such that $a(i), b(i) \in \mathcal{A}$ for different $a, b \in \Xi$;*

(ii) if $\mathcal{A}$ is consistent with $\mathcal{O}_\mathcal{L}$, then $\mathcal{O}_\mathcal{L}, \mathcal{A} \models \Psi(l)$ iff $\mathcal{A}$ contains a subset

$$\{X(l-m-1), b_1(l-m), b_2(l-m+1), b_3(l-m+2), \dots, b_m(l-1)\}, \tag{30}$$

where $m \geq 0$, $b_k \in \Sigma'$ for all $k \in [1, m]$, and $\mathfrak{A}_\mathcal{L}$, having read the word $b_1 \dots b_m$, is in the state represented by Ψ.

Proof. We obtain (i) because the only axiom with $\bot$ is $(\star_1)$ and, for consistent $\mathcal{A}$ and $\mathcal{O}_\mathcal{L}$, $b \in \Xi$ and $n \in \mathbb{Z}$, we have $(\mathcal{O}, \mathcal{A}) \models b(n)$ iff $b(n) \in \mathcal{A}$.

(ii) $(\Leftarrow)$ If there is such a subset of $\mathcal{A}$, then $(\mathcal{O}_\mathcal{L}, \mathcal{A}) \models \big([\mathbb{A} = 0] \wedge S\big)(l-m)$. One can check by induction on j that if the automaton is in a state q after reading $b_1 \dots b_{j-1}$ and q is represented by a state formula Ψ', then $(\mathcal{O}, \mathcal{A}) \models \Psi'(l-m+j)$.

$(\Rightarrow)$ If $(\mathcal{O}_\mathcal{L}, \mathcal{A}) \models A^{\iota_1}_{j_1}(l)$, for some $A^{\iota_1}_{j_1} \in \mathbb{A}$, then $(\mathcal{O}_\mathcal{L}, \mathcal{A}) \models b(l-1)$, for some $b \in \Xi$. There are two possibilities: either $b = X$ or $b \in \Sigma'$ and there exists $A^{\iota_2}_{j_2} \in \mathbb{A}$ such that $(\mathcal{O}_\mathcal{L}, \mathcal{A}) \models A^{\iota_2}_{j_2}(l-1)$. So there is a unique subset of $\mathcal{A}$ of the form (30). By induction on $j \in [1, m+1]$, we can prove that there exists a unique state formula Ψ_j such that $(\mathcal{O}_\mathcal{L}, \mathcal{A}) \models \Psi_j(l-m+j)$ and Ψ_j represents the state $\mathfrak{A}_\mathcal{L}$ is in after reading $b_1 \dots b_{j-1}$. ❑

To complete the proof of Theorem 16, we need one more lemma.

Lemma 18. *Let $\boldsymbol{q}_\mathcal{L}(x) = (\mathcal{O}_\mathcal{L}, F_{end}(x))$. For the signature Ξ above, $\boldsymbol{L}(\mathfrak{A}_\mathcal{L})$ is $\mathcal{L}$-definable iff $\boldsymbol{L}_\Xi(\boldsymbol{q}_\mathcal{L}(x))$ is $\mathcal{L}$-definable.*

Proof. Recall that the alphabet of $\boldsymbol{L}_\Xi(\boldsymbol{q}_\mathcal{L}(x))$ is $\Gamma_\Xi = \Sigma_\Xi \cup \Sigma'_\Xi$. As $(\star_3)$ is the only rule that produces the target concept F_{end} and $F_{end} \notin \Xi$, k is a certain answer to $\boldsymbol{q}_\mathcal{L}(x)$ over a Ξ-ABox $\mathcal{A}$ iff either $\mathcal{A}$ is inconsistent with $\mathcal{O}_\mathcal{L}$ or $(\mathcal{O}_\mathcal{L}, \mathcal{A}) \models \big([\mathbb{A} = 0] \wedge S \wedge Y\big)(k)$ iff, by Lemma 17, there are $a(i), b(i) \in \mathcal{A}$, for $a, b \in \Xi$ with $a \neq b$, or $\mathcal{A}$ contains a subset

$$\{X(k-m-1), b_1(k-m), \dots, b_m(k-1), Y(k)\},$$

where $b_1 \dots b_m \in \boldsymbol{L}(\mathfrak{A}_\mathcal{L})$.

Let $\Xi^{\{\}}$ and $\boldsymbol{L}^{\{\}}(\mathfrak{A}_\mathcal{L})$ stand for Ξ and, respectively, $\boldsymbol{L}(\mathfrak{A}_\mathcal{L})$, in which every $a \in \Xi$ is replaced by the set $\{a\}$. It follows that $\boldsymbol{L}_\Xi(\boldsymbol{q}_\mathcal{L}(x)) = \boldsymbol{L}_0 \cup \boldsymbol{L}_1$, where

$$\begin{aligned}
\boldsymbol{L}_0 &= \big\{\mathcal{A}a'\mathcal{B} \mid \mathcal{AB} \in \Sigma_\Xi^*, a' \in \Sigma'_\Xi\big\} \cap \big\{\mathcal{A}a\mathcal{B} \mid \mathcal{A}a\mathcal{B} \in \Gamma_\Xi^*, |a| > 1\big\}\\
\boldsymbol{L}_1 &= \big\{w_1\{X\}w\{Y'\}w_2 \mid w \in \boldsymbol{L}^{\{\}}(\mathfrak{A}_\mathcal{L}),\ w_1, w_2 \in \big(\Xi^{\{\}} \cup \{\emptyset\}\big)^*\big\}.
\end{aligned}$$

(Indeed, $\boldsymbol{L}_0$ describes the inconsistent ABoxes and $\boldsymbol{L}_1$ the consistent ones.) Clearly, the language $\boldsymbol{L}_0$ is $\mathcal{L}$-definable. Let φ be an $\mathcal{L}$-formula defining it. If $\boldsymbol{L}(\mathfrak{A}_\mathcal{L})$ is definable by an $\mathcal{L}$-formula, then so are $\boldsymbol{L}^{\{\}}(\mathfrak{A}_\mathcal{L})$ and, by Lemma 7, $\boldsymbol{L}_1$. Let ψ be the $\mathcal{L}$-formula defining $\boldsymbol{L}_1$. Then $\varphi \vee \psi$ defines $\boldsymbol{L}_\Xi(\boldsymbol{q}_\mathcal{L}(x))$. If $\boldsymbol{L}_\Xi(\boldsymbol{q}_\mathcal{L}(x))$ is definable by an $\mathcal{L}$-formula χ, then $\chi \wedge \neg\varphi$ defines $\boldsymbol{L}_1$. Thus, by Lemma 7, the language $\boldsymbol{L}^{\{\}}(\mathfrak{A}_\mathcal{L})$ is $\mathcal{L}$-definable, and so is $\boldsymbol{L}(\mathfrak{A}_\mathcal{L})$. ❑

By Theorem 5 (ii), $\boldsymbol{q}_\mathcal{L}(x)$ is $\mathcal{L}$-rewritable over Ξ-ABoxes iff $\boldsymbol{L}(\mathfrak{A}_\mathcal{L})$ is $\mathcal{L}$-definable. By (29), $\boldsymbol{L}(\mathfrak{A}_\mathcal{L})$ is $\mathcal{L}$-definable iff $\boldsymbol{M}$ rejects $\boldsymbol{x}$, which completes the proof of Theorem 16. ❑

We also observe that $LTL^{\bigcirc}_{horn}$ ontologies can be encoded by positive existential queries mediated by covering axioms that are available in LTL_{krom}:

Theorem 19. *Deciding $\mathcal{L}$-rewritability of (Boolean and specific) LTL_{krom} OMPEQs over Ξ-ABoxes is* ExpSpace*-complete.*

Proof. By Theorem 16, we only need to show the lower bound, which can be done by reduction of $LTL^{\bigcirc}_{horn}$ OMAQs $\boldsymbol{q} = (\mathcal{O}, A)$ to $LTL^{\bigcirc}_{krom}$ OMPEQs. By Remark 3, we can assume that the axioms of $\mathcal{O}$ take the form $\boldsymbol{C} \to \bot$ or $\boldsymbol{C} \to B$, for some $\boldsymbol{C} = C_1 \land \dots \land C_n$ and atomic B. We construct an $LTL^{\bigcirc}_{krom}$ OMPQ $\boldsymbol{q}' = (\mathcal{O}', \varkappa)$ that is Ξ-equivalent to $\boldsymbol{q}$ by taking $\mathcal{O}'$ with the axioms $B \land \bar{B} \to \bot$ and $\top \to B \lor \bar{B}$, for all $B \in \mathsf{sig}(\boldsymbol{q})$, where $\bar{B}$ is a fresh atom, and

$$\varkappa \;=\; A \quad \lor \bigvee_{\boldsymbol{C}\to\bot \text{ in } \mathcal{O}} \Diamond_F \Diamond_P \boldsymbol{C} \quad \lor \bigvee_{\boldsymbol{C}\to B \text{ in } \mathcal{O}} \Diamond_F \Diamond_P (\boldsymbol{C} \land \bar{B}).$$

Intuitively, $\bar{B}$ represents the negation of B and $\varkappa$ is equivalent to the formula

$$\Big[\bigwedge_{\boldsymbol{C}\to\bot \text{ in } \mathcal{O}} \Box_F \Box_P (\boldsymbol{C} \to \bot) \land \bigwedge_{\boldsymbol{C}\to B \text{ in } \mathcal{O}} \Box_F \Box_P (\boldsymbol{C} \to B) \Big] \to A.$$

It is readily seen that, for any Ξ-ABox $\mathcal{A}$, the certain answer to $\boldsymbol{q}$ over $\mathcal{A}$ is yes iff the answer to $\boldsymbol{q}'$ over $\mathcal{A}$ is yes, and k is a certain answer to $\boldsymbol{q}(x)$ over $\mathcal{A}$ iff it is also a certain answer to $\boldsymbol{q}'(x)$. It follows that $\boldsymbol{q}'$ is $\mathcal{L}$-rewritable over Ξ-ABoxes iff $\boldsymbol{q}$ is $\mathcal{L}$-rewritable. ❑

7. Deciding $\mathcal{L}$-rewritability of Linear $LTL^{\bigcirc}_{horn}$ OMPQs

As well known, deciding FO-rewritability of monadic datalog queries is 2ExpTime-complete (Cosmadakis et al., 1988; Benedikt et al., 2015; Kikot et al., 2021), which becomes PSpace for the important class of linear monadic queries (Cosmadakis et al., 1988; van der Meyden, 2000). In this section, we focus on linear $LTL^{\bigcirc}_{horn}$ OMPQs. First, in Section 7.1, we show that it suffices to consider $\bot$-free OMQs only and that deciding $\mathcal{L}$-rewritability of specific $LTL^{\Box\bigcirc}_{horn}$ OMPQs is polynomially reducible to the same problem for Boolean $LTL^{\Box\bigcirc}_{horn}$ OMPQs and the other way round. Then, in Section 7.2, for any linear $LTL^{\bigcirc}_{horn}$ OMAQ $\boldsymbol{q}$, we construct in polynomial space a DFA $\mathfrak{A}'$ such that $\boldsymbol{q}$ is $\mathcal{L}$-rewritable iff $\boldsymbol{L}(\mathfrak{A}')$ is $\mathcal{L}$-definable. So, by Theorem 15, deciding $\mathcal{L}$-rewritability of linear $LTL^{\bigcirc}_{horn}$ OMAQs can be done in PSpace. An essential part of this proof is the construction of a (polynomial-size) 2NFA $\mathfrak{A}^{\Xi}_{\boldsymbol{q}}$ that recognises a certain encoding of the language of $\boldsymbol{q}$. We also show that any DFA can be simulated by a linear $LTL^{\bigcirc}_{horn}$ OMAQ, which yields a PSpace lower bound for deciding $\mathcal{L}$-rewritability. Section 7.3 gives semantic criteria of $\mathsf{FO}(<)$- and $\mathsf{FO}(<,\equiv)$-rewritiability of $LTL^{\bigcirc}_{horn}$ OMPQs and a PSpace algorithm for checking these criteria based on $\mathfrak{A}^{\Xi}_{\boldsymbol{q}}$.

7.1 Two Useful Reductions

We start with two technical observations. The first one rids ontologies of $\bot$.

Lemma 20. *Let $\mathcal{O}$ be an $LTL^{\Box\bigcirc}_{bool}$ ontology, let $\mathcal{O}'$ result from $\mathcal{O}$ by removing every axiom of the form $C_1 \land \dots \land C_k \to \bot$, and let $\mathcal{O}''$ result from $\mathcal{O}$ by replacing every axiom of the form $C_1 \land \dots \land C_k \to \bot$ with $C_1 \land \dots \land C_k \to A'$, $A' \to \bigcirc_F A'$, $A' \to \bigcirc_P A'$, $A' \to A$, for a fresh atom A'. Let Ξ be a signature that does not contain the newly introduced atoms A'.*

(*i*) *Every Boolean OMAQ $\boldsymbol{q} = (\mathcal{O}, A)$ is Ξ-equivalent to $\boldsymbol{q}' = (\mathcal{O}'', A)$. Every specific OMAQ $\boldsymbol{q}(x) = (\mathcal{O}, A(x))$ is Ξ-equivalent to $\boldsymbol{q}'(x) = (\mathcal{O}'', A(x))$.*

(*ii*) *Every Boolean OMQ* $\boldsymbol{q} = (\mathcal{O}, \varkappa)$ *is* Ξ*-equivalent to* $\boldsymbol{q}'' = (\mathcal{O}', \varkappa')$*, where*

$$\varkappa' = \varkappa \vee \bigvee_{C_1 \wedge \dots \wedge C_k \to \bot \in \mathcal{O}} \Diamond_F \Diamond_P (C_1 \wedge \dots \wedge C_k)$$

Every specific OMQ $\boldsymbol{q}(x) = (\mathcal{O}, \varkappa(x))$ *is* Ξ*-equivalent to* $\boldsymbol{q}''(x) = (\mathcal{O}', \varkappa'(x))$.

Proof. We only show the first claim in (*i*); the other claims are similar and left to the reader. Let $\mathcal{A}$ be any Ξ-ABox. Suppose the certain answer to $\boldsymbol{q}'$ over $\mathcal{A}$ is no. This means that there is a model $\mathcal{I}$ of $\mathcal{O}''$ and $\mathcal{A}$ such that $\mathcal{I}, n \not\models A$ for all $n \in \mathbb{Z}$. Then $\mathcal{I}$ is also a model of $\mathcal{O}$ and $\mathcal{A}$. Indeed, if $\mathcal{I}, n \models C_1 \wedge \dots \wedge C_k$, for some axiom $C_1 \wedge \dots \wedge C_k \to \bot$ in $\mathcal{O}$ and $n \in \mathbb{Z}$, then $\mathcal{I}, n \models A'$, and so $\mathcal{I}, n \models A$, which is a contradiction. It follows that the answer to $\boldsymbol{q}$ over $\mathcal{A}$ is no. Conversely, suppose the answer to $\boldsymbol{q}$ over $\mathcal{A}$ is no. Let $\mathcal{I}$ be a model of $\mathcal{O}$ and $\mathcal{A}$ such that $\mathcal{I}, n \not\models A$ for all $n \in \mathbb{Z}$. Extend $\mathcal{I}$ to the fresh atoms A' by setting $\mathcal{I}, n \not\models A'$. Then $\mathcal{I}$ is a model of $\mathcal{O}''$ and $\mathcal{A}$, as required. ❑

The next proposition, which will be used in the proofs of Theorems 16 and 22, shows that deciding $\mathcal{L}$-rewritability of specific $LTL^{\Box\bigcirc}_{horn}$-OMPQs is polynomially reducible to deciding $\mathcal{L}$-rewritability of Boolean $LTL^{\Box\bigcirc}_{horn}$-OMPQs. Recall from (e.g., Artale et al., 2021) that, for any $LTL^{\Box\bigcirc}_{horn}$-ontology $\mathcal{O}$ and any ABox $\mathcal{A}$ consistent with $\mathcal{O}$, there is a *canonical* (or *minimal*) *model* $\mathcal{C}_{\mathcal{O},\mathcal{A}}$ of $\mathcal{O}$ and $\mathcal{A}$ such that, for any positive concept $\varkappa$ and any $k \in \mathbb{Z}$,

$$(\mathcal{O}, \mathcal{A}) \models \varkappa(k) \quad \text{iff} \quad \mathcal{C}_{\mathcal{O},\mathcal{A}} \models \varkappa(k). \tag{31}$$

Proposition 21. *Let* $\mathcal{O}$ *be an* $LTL^{\Box\bigcirc}_{horn}$*-ontology without occurrences of* $\bot$*,* A *an atom,* $\varkappa$ *a positive concept, and* Ξ *a signature. Let* X, X' *be fresh atomic concepts and* $\Xi_X = \Xi \cup \{X\}$*. Then the following hold:*

(*i*) *The specific OMAQ* $\boldsymbol{q}(x) = (\mathcal{O}, A(x))$ *is* $\mathcal{L}$*-rewritable over* Ξ*-ABoxes iff the Boolean OMAQ* $\boldsymbol{q}_X = (\mathcal{O} \cup \{A \wedge X \to X'\}, X')$ *is* $\mathcal{L}$*-rewritable over* Ξ_X*-ABoxes.*

(*ii*) *The specific OMPQ* $\boldsymbol{q}(x) = (\mathcal{O}, \varkappa(x))$ *is* $\mathcal{L}$*-rewritable over* Ξ*-ABoxes iff the Boolean OMPQ* $\boldsymbol{q}_X = (\mathcal{O}, \varkappa \wedge X)$ *is* $\mathcal{L}$*-rewritable over* Ξ_X*-ABoxes.*

Proof. We only prove (*ii*). Suppose $\boldsymbol{Q}(x)$ is an $\mathcal{L}$-rewriting of $\boldsymbol{q}(x) = (\mathcal{O}, \varkappa(x))$ over Ξ-ABoxes. We show that $\exists x\, (\boldsymbol{Q}(x) \wedge X(x))$ is an $\mathcal{L}$-rewriting of $\boldsymbol{q}_X$ over Ξ_X-ABoxes, that is, for every Ξ_X-ABox $\mathcal{A}$, we have $\mathfrak{S}_{\mathcal{A}} \models \exists x\, (\boldsymbol{Q}(x) \wedge X(x))$ iff $\mathcal{C}_{\mathcal{O},\mathcal{A}} \models (\varkappa \wedge X)(n)$, for some $n \in \mathbb{Z}$. If $\mathfrak{S}_{\mathcal{A}} \models \exists x\, (\boldsymbol{Q}(x) \wedge X(x))$, then $\mathfrak{S}_{\mathcal{A}} \models \boldsymbol{Q}(n)$ and $\mathfrak{S}_{\mathcal{A}} \models X(n)$, for some $n \in \mathsf{tem}(\mathcal{A})$. Since $\boldsymbol{Q}(x)$ is a rewriting of $\boldsymbol{q}(x)$, we have $\mathcal{C}_{\mathcal{O},\mathcal{A}} \models \varkappa(n)$, and since X does not occur in $\mathcal{O}$, we must have $X(n) \in \mathcal{A}$. Conversely, suppose $\mathcal{C}_{\mathcal{O},\mathcal{A}} \models (\varkappa \wedge X)(n)$, for some $n \in \mathbb{Z}$. Then clearly $X(n) \in \mathcal{A}$ and, by (31), n is a certain answer to $\boldsymbol{q}(x)$ over $\mathcal{A}$, from which $\mathfrak{S}_{\mathcal{A}} \models \exists x\, (\boldsymbol{Q}(x) \wedge X(x))$.

Suppose $\boldsymbol{Q}$ is an $\mathcal{L}$-rewriting of $\boldsymbol{q}_X$ over Ξ_X-ABoxes. Fix a variable x that does not occur in $\boldsymbol{Q}$ and let $\boldsymbol{Q}^-(x)$ be the result of replacing every occurrence of $X(y)$ in $\boldsymbol{Q}$ with $(x = y)$. We show that $\boldsymbol{Q}^-(x)$ is an $\mathcal{L}$-rewriting of $\boldsymbol{q}(x)$ over Ξ-ABoxes. Given a Ξ-ABox $\mathcal{A}$, for any $k \in \mathsf{tem}(\mathcal{A})$, we have

$$\mathcal{C}_{\mathcal{O},\mathcal{A}} \models \varkappa(k) \quad \text{iff} \quad \mathcal{C}_{\mathcal{O},\mathcal{A} \cup \{X(k)\}} \models (\varkappa \wedge X)(k) \quad \text{iff} \quad \mathfrak{S}_{\mathcal{A} \cup \{X(k)\}} \models \boldsymbol{Q} \quad \text{iff} \quad \mathfrak{S}_{\mathcal{A}} \models \boldsymbol{Q}^-(k)$$

as required. ❑

7.2 Deciding FO-rewritability of Linear $LTL^{\bigcirc}_{horn}$ OMAQs

In this section, we use $\mathcal{A}$ to refer to both the ABox $\mathcal{A}$ and its representation as the word $w_{\mathcal{A}}$ over the alphabet Σ_Ξ. For a linear $LTL^{\bigcirc}_{horn}$ ontology $\mathcal{O}$, let $idb(\mathcal{O})$ be the set of atoms that occur on the right-hand side of axioms in $\mathcal{O}$. For an atom A and $j \in \mathbb{Z}$, we define $\bigcirc^0 A = A$ and, inductively, $\bigcirc^{j+1} A = \bigcirc_F \bigcirc^j A$ for $j \geq 0$, and $\bigcirc^{j-1} A = \bigcirc_P \bigcirc^j A$ for $j \leq 0$. Let $\boldsymbol{q} = (\mathcal{O}, \varkappa)$ be an $LTL^{\bigcirc}_{horn}$ OMPQ. For a type $\boldsymbol{\tau}$ for $\boldsymbol{q}$ (see Proposition 5), we denote by $\boldsymbol{\tau}^\Xi$ its restriction to atoms in Ξ and their negations. Given a model $\mathcal{I}$ of $\mathcal{O}$ and $n \in \mathbb{Z}$, we denote by $\boldsymbol{\tau}_{\mathcal{I}}(n)$ the type for $\boldsymbol{q}$ that is true in $\mathcal{I}$ at n. For a $\bot$-free $\mathcal{O}$, we write $\boldsymbol{\tau}_{\mathcal{O},\mathcal{A}}(n)$ instead of $\boldsymbol{\tau}_{\mathcal{C}_{\mathcal{O},\mathcal{A}}}(n)$, where $\mathcal{C}_{\mathcal{O},\mathcal{A}}$ is the canonical model of $\mathcal{O}$ and $\mathcal{A}$ with the key property (31).

Theorem 22. *For $\mathcal{L} \in \{\mathsf{FO}(<), \mathsf{FO}(<,\equiv), \mathsf{FO}(<,\mathsf{MOD})\}$, deciding $\mathcal{L}$-rewritability of linear $LTL^{\bigcirc}_{horn}$ OMAQs over Ξ-ABoxes is* PSpace*-complete.*

Proof. To show the upper bound, it suffices, by Lemma 20 (i) and Proposition 21, to consider Boolean $LTL^{\bigcirc}_{horn}$ OMAQs $\boldsymbol{q} = (\mathcal{O}, B)$ with a $\bot$-free $\mathcal{O}$. In view of Remark 3, we can also assume that the axioms in $\mathcal{O}$ are of two types:

$$C_1 \wedge \dots \wedge C_k \to A', \tag{32}$$

$$C_1 \wedge \dots \wedge C_k \wedge \bigcirc^i A \to A', \tag{33}$$

where $k \geq 0$, $C_1, \dots, C_k$ contain no IDB atoms, $A \in idb(\mathcal{O})$ and $i \in \{-1, 0, 1\}$.

We define a quadruple $\mathfrak{A}^\Xi_{\mathcal{O}} = (Q, \Sigma_\Xi, \delta, Q_0)$—a 2NFA without final states—giving the transition function δ as a set of transitions of the form $q \to_{a,d} q'$. Namely, we set $Q_0 = \{q_0\}$, $Q = \bigcup_{\alpha \in \mathcal{O}} Q_\alpha \cup \{q_0, q_h\} \cup \{q_A \mid A \in idb(\mathcal{O})\}$ and

$$\delta = \bigcup_{\alpha \in \mathcal{O}} \delta_\alpha \cup \{q_0 \to_{a,1} q_0 \mid a \in \Sigma_\Xi\}.$$

The states in Q_α and transitions in δ_α are defined as follows. If $\alpha \in \mathcal{O}$ is of the form (32) and $C_i = \bigcirc^{j_i} A_i$, $1 \leq i \leq k$, then $Q_\alpha = \{q_\alpha\} \cup Q'_\alpha$ and $\delta_\alpha = \{q_0 \to_{a,0} q_\alpha \mid a \in \Sigma_\Xi\} \cup \delta'_\alpha$, where Q'_α and δ'_α are defined below. If $j_1 < 0$ (the cases $j_1 = 0$ and $j_1 > 0$ are analogous), then δ'_α is such that $\mathfrak{A}^\Xi_{\mathcal{O}}$ makes $|j_1|$ steps to the left by reading any symbols from Σ_Ξ. If after that the 2NFA reads any symbol a with $A_1 \notin a$ (remember that $C_1 = \bigcirc^{j_1} A_1$), it moves to the 'dead-end' state q_h. Otherwise, it makes $|j_1|$ steps to the right and repeats the same process for $C_2 = \bigcirc^{j_2} A_2$, etc. After executing the transitions for $C_k = \bigcirc^{j_k} A_k$ and provided that q_h has been avoided, the 2NFA comes to state $q_{A'}$. For α of the form (33), Q_α is the same as above but $\delta_\alpha = \{q_A \to_{a,0} q_\alpha \mid a \in \Sigma_\Xi\} \cup \delta'_\alpha$ for the same δ'_α as above, leading to either q_h or $q_{A'}$.

In what follows, $\mathsf{b}_\bullet(\mathcal{A})$ and $\mathsf{b}(\mathcal{A})$, for $\bullet \in \{lr, rr, rl, ll\}$ and $\mathcal{A} \in \Sigma^*_\Xi$, are defined with respect to $\mathfrak{A}^\Xi_{\mathcal{O}}$ (see Section 5 taking into account that the final states of the 2NFA are not relevant in the definition of $\mathsf{b}_\bullet(\mathcal{A})$). Let $X_{\mathcal{A}}(\ell)$ be the reflexive and transitive closure of $\mathsf{b}_{ll}(\mathcal{A}^{>\ell}) \circ \mathsf{b}_{rr}(\mathcal{A}^{\leq \ell})$, for $0 \leq \ell < |\mathcal{A}|$. Let $N = M + 2M^2$, where M is the number of occurrences of $\bigcirc_F$ and $\bigcirc_P$ in $\mathcal{O}$. The proof of the following technical result can be found in Appendix A.4:

Lemma 23. *Let $\mathcal{A} \in \Sigma^*_\Xi$ be of the form $\emptyset^N \mathcal{B} \emptyset^N$. Then $A \in \boldsymbol{\tau}^{\mathsf{sig}(\mathcal{O})}_{\mathcal{O},\mathcal{A}}(\ell)$ iff there exists a run $(q_0, 0), \dots, (q, \ell), (q_A, i)$ of $\mathfrak{A}^\Xi_{\mathcal{O}}$ on $\mathcal{A}$, for all ℓ with $N \leq \ell < |\mathcal{A}| - N$.*

As $\mathfrak{A}^{\Xi}_{\mathcal{O}}$ has a run $(q_0, 0), \dots, (q, \ell), (q_A, i)$ on $\mathcal{A}$ iff $(q_0, q_A) \in \mathsf{b}_{lr}(\mathcal{A}^{\leq \ell}) \circ X_{\mathcal{A}}(\ell)$, for all $\ell < |\mathcal{A}|$ and $A \in \mathsf{sig}(\mathcal{O})$, we immediately obtain that

$$\boldsymbol{\tau}^{\mathsf{sig}(\mathcal{O})}_{\mathcal{O},\mathcal{A}}(\ell) = \{A \mid (q_0, q_A) \in \mathsf{b}_{lr}(\mathcal{A}^{\leq \ell}) \circ X_{\mathcal{A}}(\ell)\}. \tag{34}$$

Define a 2NFA $\mathfrak{A}^{\Xi}_{\boldsymbol{q}} = (\Sigma_{\Xi}, Q', \delta', Q_0, F)$ with $Q' = Q \cup \{q_B\}$, $\delta' = \delta \cup \{q_B \to_{a,1} q_B \mid a \in \Sigma_{\Xi}\}$, and $F = \{q_B\}$. Using Lemma 23, we obtain:

$$\boldsymbol{L}_{\Xi}(\boldsymbol{q}) = \{\mathcal{A} \in \Sigma^*_{\Xi} \mid \emptyset^N \mathcal{A} \emptyset^N \in \boldsymbol{L}(\mathfrak{A}^{\Xi}_{\boldsymbol{q}})\}. \tag{35}$$

Our aim is to construct in polynomial space a DFA $\mathfrak{A}'$ with $\boldsymbol{L}_{\Xi}(\boldsymbol{q}) = \boldsymbol{L}(\mathfrak{A}')$ whose $\mathcal{L}$-definability can be decided in PSpace. We construct $\mathfrak{A}'$ from $\mathfrak{A}^{\Xi}_{\boldsymbol{q}}$ in the same way as in Section 5 except the definition of q_0' and F', which is now as follows: $q_0' = (\{(q_0, q_0)\}, \mathsf{b}_{rr}(\emptyset^N))$ and $F' = \{(B_{lr}, B_{rr}) \mid (q_0, q_1) \in B_{lr} \circ X\}$, where X is the reflexive and transitive closure of $\mathsf{b}_{ll}(\emptyset^N) \circ B_{rr}$. By (35), we have $\boldsymbol{L}_{\Xi}(\boldsymbol{q}) = \boldsymbol{L}(\mathfrak{A}')$, and it is readily seen that $\mathfrak{A}'$ is constructible from $\boldsymbol{q}$ in PSpace. That $\mathcal{L}$-definability of $\mathfrak{A}'$ is decidable in PSpace, follows from the proof of Theorem 15.

We now establish a matching lower bound. By Lemma 20 and Proposition 21 (i), it suffices to show it for *specific* linear $LTL^{\bigcirc}_{horn}$ OMAQs $\boldsymbol{q}(x) = (\mathcal{O}, F_{end}(x))$, which will be done by reduction of $\mathcal{L}$-rewritability for DFAs $\mathfrak{A} = (Q, \Omega, \delta, q_0, F)$. We set $\Xi = \Omega \cup \{X, Y\}$ with fresh X, Y and construct a linear $LTL^{\bigcirc}_{horn}$ ontology $\mathcal{O}$ with $idb(\mathcal{O}) \subseteq \{\bar{q} \mid q \in Q\} \cup \{F_{end}\}$ (treating $\bar{q}$ as an atomic concept) that simulates the behaviour of the DFA $\mathfrak{A}$ by means of the axioms $X \to \bigcirc_F \bar{q}_0$, $\bar{q} \wedge Y \to F_{end}$ for all $q \in F$, $\bar{q} \wedge A \to \bigcirc_F \bar{r}$ for all transitions $q \to_A r$ in δ, $A \wedge C \to \bot$ for all distinct $A, C \in \Xi$. Then $\boldsymbol{L}(\mathfrak{A})$ is $\mathcal{L}$-definable iff $\boldsymbol{L}_{\Xi}(\boldsymbol{q}(x))$ is $\mathcal{L}$-definable, which is proved similarly to Lemma 18. ❑

7.3 Deciding FO-rewritability of Linear $LTL^{\bigcirc}_{horn}$ OMPQs

We next show that $\mathsf{FO}(<)$- and $\mathsf{FO}(<, \equiv)$-definability of linear $LTL^{\bigcirc}_{horn}$ OMPQs can be recognised in PSpace. By Lemma 20 and Proposition 21, it suffices to do this for Boolean OMPQs $\boldsymbol{q} = (\mathcal{O}, \varkappa)$ with $\bot$-free $\mathcal{O}$, in which case we can assume that $\varkappa = \Diamond_P \Diamond_F \varkappa'$. Let $\boldsymbol{T}_{\boldsymbol{q}}$ be the set of all types for $\boldsymbol{q}$.

We start with $\mathsf{FO}(<)$-definability. Recall that we established the PSpace upper bound for deciding $\mathsf{FO}(<)$-definability of the language of a given DFA $\mathfrak{A}$ in two steps. First, in Theorem 6 (i), we gave a criterion in terms of words in the alphabet of $\mathfrak{A}$, and then, in Theorem 15, we showed how to check that criterion in PSpace. Similarly, in Theorem 24 below, we prove a criterion of $\mathsf{FO}(<)$-rewritability of OMPQs we are dealing with in terms of Σ_{Ξ}-ABoxes. Then, in Theorem 25, we show how this criterion can be checked in PSpace.

Theorem 24. *Let $\boldsymbol{q} = (\mathcal{O}, \varkappa)$ be an OMPQ with a $\bot$-free $LTL^{\Box\bigcirc}_{horn}$-ontology $\mathcal{O}$. Then $\boldsymbol{q}$ is not $\mathsf{FO}(<)$-rewritable over Ξ-ABoxes iff there exist $\mathcal{A}, \mathcal{B}, \mathcal{D} \in \Sigma^*_{\Xi}$ and $k \geq 2$ such that the following conditions hold:*

(i) $\neg\varkappa \in \boldsymbol{\tau}_{\mathcal{O},\mathcal{A}\mathcal{B}^k\mathcal{D}}(|\mathcal{A}| - 1) = \boldsymbol{\tau}_{\mathcal{O},\mathcal{A}\mathcal{B}^k\mathcal{D}}(|\mathcal{A}\mathcal{B}^k| - 1)$;

(ii) $\varkappa \in \boldsymbol{\tau}_{\mathcal{O},\mathcal{A}\mathcal{B}^{k+1}\mathcal{D}}(|\mathcal{A}\mathcal{B}| - 1) = \boldsymbol{\tau}_{\mathcal{O},\mathcal{A}\mathcal{B}^{k+1}\mathcal{D}}(|\mathcal{A}\mathcal{B}^{k+1}| - 1)$.

Moreover, we can find $\mathcal{A}, \mathcal{B}, \mathcal{D}$ and k such that $|\mathcal{A}|, |\mathcal{D}|, k \leq 2^{O(|\boldsymbol{q}|)}$.

Proof. Define a DFA $\mathfrak{A} = (Q, \Sigma_\Xi, \delta, q_{-1}, F)$ by taking $Q = 2^{\boldsymbol{T}_{\boldsymbol{q}}}$, $q_{-1} = \boldsymbol{T}_{\boldsymbol{q}}$, $F = \{q \in Q \mid \varkappa \in \boldsymbol{\tau}$ for all $\boldsymbol{\tau} \in q\}$, and $\delta(q, a) = \{\boldsymbol{\tau} \mid \boldsymbol{\tau}' \to_a \boldsymbol{\tau}$ for some $\boldsymbol{\tau}' \in q\}$, where $\to_a$ was defined in the proof of Proposition 5. As in that proof we can show that $\boldsymbol{L}_\Xi(\boldsymbol{q}) = \boldsymbol{L}(\mathfrak{A})$. We write $q \Rightarrow_{\mathcal{A}} q'$ to say that, having started in state q and read $\mathcal{A} \in \Sigma_\Xi^*$, the DFA $\mathfrak{A}$ is in state q'.

We require the following property of $\mathfrak{A}$. For a set $\{\boldsymbol{\tau}_i \mid i \in I\}$ of types for $\boldsymbol{q}$, let $\bigoplus_{i\in I} \boldsymbol{\tau}_i = \bigcap_{i\in I} \boldsymbol{\tau}_i^+ \cup \bigcup_{i\in I} \boldsymbol{\tau}_i^-$, where $\boldsymbol{\tau}_i^+$ and $\boldsymbol{\tau}_i^-$ are the sets of positive and negated concepts in $\boldsymbol{\tau}_i$, respectively. Suppose now $q_{-1} \Rightarrow_{\mathcal{A}_0} q_0 \Rightarrow_{\mathcal{A}_1} \cdots \Rightarrow_{\mathcal{A}_{n-1}} q_{n-1} \Rightarrow_{\mathcal{A}_n} q_n$ is a run of $\mathfrak{A}$ on $\mathcal{A} = \mathcal{A}_0 \dots \mathcal{A}_n$, and let $\bar{q}_i = \{\boldsymbol{\tau} \in q_i \mid \boldsymbol{\tau} \to_{\mathcal{A}_{i+1}\dots\mathcal{A}_n} \boldsymbol{\tau}',$ for some $\boldsymbol{\tau}' \in q_n\}$. Then

$$\boldsymbol{\tau}_{\mathcal{O},\mathcal{A}}(i) = \bigoplus \bar{q}_i, \qquad \text{for } -1 \le i \le n. \tag{36}$$

($\Rightarrow$) Suppose $\boldsymbol{q}$ is not $\mathsf{FO}(<)$-rewritable. By applying Theorem 6 (*i*) to $\mathfrak{A}$, we find $\mathcal{A}, \mathcal{B}, \mathcal{D} \in \Sigma_\Xi^*$ and $k \ge 2$ such that $q_{-1} \Rightarrow_{\mathcal{A}} q_0$, $q_0 \Rightarrow_{\mathcal{B}} q_1$, $q_0 \Rightarrow_{\mathcal{B}^k} q_0$ and $q_0 \Rightarrow_{\mathcal{D}} q_0'$, $q_1 \Rightarrow_{\mathcal{D}} q_1'$, for some $q_0, q_1, q_0', q_1' \in Q$ such that $q_0' \notin F$ and $q_1' \in F$. Since $q_0' \notin F$, by (36), we obtain $\neg\varkappa \in \boldsymbol{\tau}_{\mathcal{O},\mathcal{A}\mathcal{B}^k\mathcal{D}}(|\mathcal{A}| - 1) = \boldsymbol{\tau}_{\mathcal{O},\mathcal{A}\mathcal{B}^k\mathcal{D}}(|\mathcal{A}\mathcal{B}^k| - 1)$ as required in (*i*). And since $q_1' \in F$, (36) yields $\varkappa \in \boldsymbol{\tau}_{\mathcal{O},\mathcal{A}\mathcal{B}^{k+1}\mathcal{D}}(|\mathcal{A}\mathcal{B}| - 1) = \boldsymbol{\tau}_{\mathcal{O},\mathcal{A}\mathcal{B}^{k+1}\mathcal{D}}(|\mathcal{A}\mathcal{B}^{k+1}| - 1)$, as required in (*ii*).

($\Leftarrow$) Assuming (*i*) and (*ii*), let q_0, q_1, q_2 be states in $\mathfrak{A}$ with $q_{-1} \Rightarrow_{\mathcal{A}} q_0 \Rightarrow_{\mathcal{B}} q_1 \Rightarrow_{\mathcal{B}^{k-1}} q_2 \Rightarrow_{\mathcal{B}} q_2'$. Let q_3, q_3' be such that $q_2 \Rightarrow_{\mathcal{D}} q_3$ and $q_2' \Rightarrow_{\mathcal{D}} q_3'$. It follows by (36) that $q_3 \notin F$ and $q_3' \in F$. Observe that, if we had $q_0 = q_2$, we could conclude that $\boldsymbol{q}$ is not $\mathsf{FO}(<)$-rewritable, as the conditions of aperiodicity for $\mathfrak{A}$ (see the proof of ($\Rightarrow$)) would be satisfied. Since we are not guaranteed that, we use the following property of the canonical models that follow from (*i*) and (*ii*): (a) $\boldsymbol{\tau}_{\mathcal{O},\mathcal{A}\mathcal{B}^k\mathcal{D}}(|\mathcal{A}\mathcal{B}^k| - 1) = \boldsymbol{\tau}_{\mathcal{O},\mathcal{A}\mathcal{B}^{kj}\mathcal{D}}(|\mathcal{A}\mathcal{B}^{kj}| - 1)$, for any $j \ge 1$; (b) $\boldsymbol{\tau}_{\mathcal{O},\mathcal{A}\mathcal{B}^{k+1}\mathcal{D}}(|\mathcal{A}\mathcal{B}^{k+1}| - 1) = \boldsymbol{\tau}_{\mathcal{O},\mathcal{A}\mathcal{B}^{kj+1}\mathcal{D}}(|\mathcal{A}\mathcal{B}^{kj+1}| - 1)$, for any $j \ge 1$. Take some $i, j \ge 1$ that satisfy $q_0 \Rightarrow_{\mathcal{A}\mathcal{B}^{ki}} q_4 \Rightarrow_{\mathcal{B}} q_4' \Rightarrow_{\mathcal{B}^{kj}} q_4 \Rightarrow_{\mathcal{B}} q_4'$, for some $q_4, q_4' \in Q$. By (*i*), (*ii*), (a) and (b), we have $q_5 \notin F$ and $q_5' \in F$ for such q_5 and q_5' that $q_4 \Rightarrow_{\mathcal{D}} q_5$ and $q_4' \Rightarrow_{\mathcal{D}} q_5'$. Therefore, $\boldsymbol{q}$ is not $\mathsf{FO}(<)$-rewritable, as the conditions of aperiodicity for $\mathfrak{A}$ are satisfied (as in the ($\Rightarrow$)-proof with $\mathcal{A}$, $\mathcal{B}$, $\mathcal{D}$ and k being $\mathcal{A}\mathcal{B}^{ki}$, $\mathcal{B}$, $\mathcal{D}$ and kj, respectively).

To establish the bounds on the size of $\mathcal{A}$, $\mathcal{D}$ and k, we first notice that there is $\mathcal{A}$ with $|\mathcal{A}| \le 2|\boldsymbol{T}_{\boldsymbol{q}}|^2$. Indeed, consider the sequence

$$\big(\boldsymbol{\tau}_{\mathcal{O},\mathcal{A}\mathcal{B}^k\mathcal{D}}(0), \boldsymbol{\tau}_{\mathcal{O},\mathcal{A}\mathcal{B}^{k+1}\mathcal{D}}(0)\big), \dots, \big(\boldsymbol{\tau}_{\mathcal{O},\mathcal{A}\mathcal{B}^k\mathcal{D}}(|\mathcal{A}| - 2), \boldsymbol{\tau}_{\mathcal{O},\mathcal{A}\mathcal{B}^{k+1}\mathcal{D}}(|\mathcal{A}| - 2)\big).$$

If the i-th member of this sequence is equal to its j-th member, for $i < j$, then we take $\mathcal{A}' = \mathcal{A}^{<i}\mathcal{A}^{\ge j}$, where $\mathcal{A}^{<i}$ is the prefix of $\mathcal{A}$ before i and $\mathcal{A}^{\ge j}$ the suffix of $\mathcal{A}$ starting at j. Then $\boldsymbol{\tau}_{\mathcal{O},\mathcal{A}'\mathcal{B}^k\mathcal{D}}(|\mathcal{A}'| - 1) = \boldsymbol{\tau}_{\mathcal{O},\mathcal{A}\mathcal{B}^k\mathcal{D}}(|\mathcal{A}| - 1)$ and $\boldsymbol{\tau}_{\mathcal{O},\mathcal{A}'\mathcal{B}^{k+1}\mathcal{D}}(|\mathcal{A}'\mathcal{B}| - 1) = \boldsymbol{\tau}_{\mathcal{O},\mathcal{A}\mathcal{B}^k\mathcal{D}}(|\mathcal{A}\mathcal{B}| - 1)$, and conditions (*i*) and (*ii*) are satisfied with $\mathcal{A}'$ in place of $\mathcal{A}$. In the same way we obtain the upper bound for $\mathcal{D}$. To show that there exists $k \le 2|\boldsymbol{T}_{\boldsymbol{q}}|^2$, we consider the sequence

$$\big(\boldsymbol{\tau}_{\mathcal{O},\mathcal{A}\mathcal{B}^k\mathcal{D}}(|\mathcal{A}\mathcal{B}| - 1), \boldsymbol{\tau}_{\mathcal{O},\mathcal{A}\mathcal{B}^{k+1}\mathcal{D}}(|\mathcal{A}\mathcal{B}^2| - 1)\big), \dots,$$
$$\big(\boldsymbol{\tau}_{\mathcal{O},\mathcal{A}\mathcal{B}^k\mathcal{D}}(|\mathcal{A}\mathcal{B}^{k-1}| - 1), \boldsymbol{\tau}_{\mathcal{O},\mathcal{A}\mathcal{B}^{k+1}\mathcal{D}}(|\mathcal{A}\mathcal{B}^k| - 1)\big).$$

Clearly, if the i-th member of this sequence is equal to its j-th member, for $i < j$, then conditions (*i*) and (*ii*) are satisfied with $k - (j - i)$ in place of k. ❑

In the theorem above, we did not claim that there is $\mathcal{B}$ with $|\mathcal{B}| \le 2^{O(|\boldsymbol{q}|)}$. However, this is indeed the case for linear $LTL^{\bigcirc}_{horn}$-ontologies, as follows from the proof of the next result:

Theorem 25. *Deciding* $\mathsf{FO}(<)$*-rewritability of OMPQs* $\boldsymbol{q} = (\mathcal{O}, \varkappa)$ *with a linear* $LTL^{\bigcirc}_{horn}$*-ontology* $\mathcal{O}$ *over* Ξ*-ABoxes can be done in* PSpace.

Proof. By Theorem 24, we need to check the existence of $\mathcal{A}, \mathcal{B}, \mathcal{D}$, $k \geq 2$, such that $|\mathcal{A}|, |\mathcal{D}|, k \leq 2^{O(|\boldsymbol{q}|)}$ and conditions (i) and (ii) hold. Without loss of generality, we assume that $\mathcal{A}$ has a prefix $\emptyset^N$ and $\mathcal{D}$ has a suffix $\emptyset^N$. (As before, $N = M + 2M^2$, where M is the number of occurrences of $\bigcirc_F$ and $\bigcirc_P$ in $\mathcal{O}$.)

We start by guessing numbers $N_\mathcal{A} = |\mathcal{A}|$, $N_\mathcal{D} = |\mathcal{D}|$ and k. We guess two types $\boldsymbol{\tau}_0$ and $\boldsymbol{\tau}_1$ that represent $\boldsymbol{\tau}_{\mathcal{O},\mathcal{A}\mathcal{B}^k\mathcal{D}}(N)$ and $\boldsymbol{\tau}_{\mathcal{O},\mathcal{A}\mathcal{B}^k\mathcal{D}}(|\mathcal{A}|-1)$, respectively, and three types $\boldsymbol{\tau}'_0$, $\boldsymbol{\tau}''_0$, $\boldsymbol{\tau}'_1$ that represent $\boldsymbol{\tau}_{\mathcal{O},\mathcal{A}\mathcal{B}^{k+1}\mathcal{D}}(N)$, $\boldsymbol{\tau}_{\mathcal{O},\mathcal{A}\mathcal{B}^{k+1}\mathcal{D}}(|\mathcal{A}|-1)$ and, respectively, $\boldsymbol{\tau}_{\mathcal{O},\mathcal{A}\mathcal{B}^{k+1}\mathcal{D}}(|\mathcal{A}\mathcal{B}|-1)$. Next, we compute $\mathsf{b}(\emptyset^N)$ and guess $\mathsf{b}(\mathcal{A})$, $\mathsf{b}(\mathcal{B})$, $\mathsf{b}(\mathcal{D})$. Note that, given $\mathsf{b}(\mathcal{B})$, we are able to compute $\mathsf{b}(\mathcal{X})$ for each $\mathcal{X} \in \{\mathcal{B}^i \mid 1 \leq i \leq k+1\}$. Now, we guess $\mathcal{A}$—symbol by symbol—by means of a sequence of pairs

$$(\mathsf{b}(\mathcal{A}^{\leq 0}), \mathsf{b}(\mathcal{A}^{>0})), \dots, (\mathsf{b}(\mathcal{A}^{\leq N_\mathcal{A}-1}), \mathsf{b}(\mathcal{A}^{>N_\mathcal{A}-1}))$$

such that $\mathsf{b}(\mathcal{A}^{\leq i}) \cdot \mathsf{b}(\mathcal{A}^{>i}) = \mathsf{b}(\mathcal{A})$, for all i, and there exist $a_i \in \Sigma_\Xi$ with $\mathsf{b}(\mathcal{A}^{\leq i+1}) = \mathsf{b}(\mathcal{A}^{\leq i}) \cdot \mathsf{b}(a_i)$ and $\mathsf{b}(\mathcal{A}^{>i}) = \mathsf{b}(a_i) \cdot \mathsf{b}(\mathcal{A}^{>i+1})$; we also require that $a_i = \emptyset$ for $i < N$. Observe that, by (34), the pairs of the sequence with $i \geq N$ together with $\mathsf{b}(\mathcal{B})$ and $\mathsf{b}(\mathcal{D})$ give $\boldsymbol{\tau}^{\mathsf{sig}(\mathcal{O})}_{\mathcal{O},\mathcal{A}\mathcal{B}^k\mathcal{D}}(i)$. When computing $\boldsymbol{\tau}^{\mathsf{sig}(\mathcal{O})}_{\mathcal{O},\mathcal{A}\mathcal{B}^k\mathcal{D}}(N)$, we check whether it is subsumed by $\boldsymbol{\tau}_0$ (if not, the algorithm terminates with an answer no). We also need to check that $\varkappa' \in \boldsymbol{\tau}_{\mathcal{C}_{\mathcal{O},\{A(0) \mid A \in \boldsymbol{\tau}_0\}}}(0)$ implies $\varkappa' \in \boldsymbol{\tau}_0$, for each $\varkappa'$ of the form $\Box_P \varkappa''$, $\Diamond_P \varkappa''$ from $\mathsf{sub}_{\boldsymbol{q}}$ (if not, the algorithm terminates and returns no). We have now checked that the type $\boldsymbol{\tau}_0$ is potentially guessed correctly (subject to further checks). We can apply the same method to check that $\boldsymbol{\tau}'_0$ is potentially guessed correctly. For the remaining $N < i < N_\mathcal{A}$, since $\boldsymbol{\tau}_{\mathcal{O},\mathcal{A}\mathcal{B}^k\mathcal{D}}(i)$ is determined by $\boldsymbol{\tau}^{\mathsf{sig}(\mathcal{O})}_{\mathcal{O},\mathcal{A}\mathcal{B}^k\mathcal{D}}(i)$ and $\boldsymbol{\tau}_{\mathcal{O},\mathcal{A}\mathcal{B}^k\mathcal{D}}(i-1)$, we are able to compute $\boldsymbol{\tau}_{\mathcal{O},\mathcal{A}\mathcal{B}^k\mathcal{D}}(|\mathcal{A}|-1)$ or obtain a conflict, e.g., $\neg\Diamond_F A \in \boldsymbol{\tau}_{\mathcal{O},\mathcal{A}\mathcal{B}^k\mathcal{D}}(i-1)$ and $A \in \boldsymbol{\tau}^{\mathsf{sig}(\mathcal{O})}_{\mathcal{O},\mathcal{A}\mathcal{B}^k\mathcal{D}}(i)$. In the latter case, the algorithm terminates answering no. In the former case, we check if $\boldsymbol{\tau}_{\mathcal{O},\mathcal{A}\mathcal{B}^k\mathcal{D}}(|\mathcal{A}|-1)$ is equal to $\boldsymbol{\tau}_1$, in which case $\boldsymbol{\tau}_1$ is guessed correctly, and if not, the algorithm terminates answering no. In the same way, we check if $\boldsymbol{\tau}''_0$ is guessed correctly using $\mathcal{C}_{\mathcal{O},\mathcal{A}\mathcal{B}^{k+1}\mathcal{D}}$.

Now, we show how to check that the types $\boldsymbol{\tau}_{\mathcal{O},\mathcal{A}\mathcal{B}^k\mathcal{D}}(i)$, for $|\mathcal{A}| \leq i < |\mathcal{A}\mathcal{B}^k|$, are correct, that $\boldsymbol{\tau}'_1$ is guessed correctly, and that the types $\boldsymbol{\tau}_{\mathcal{O},\mathcal{A}\mathcal{B}^{k+1}\mathcal{D}}(i)$ with $|\mathcal{A}\mathcal{B}| \leq i < |\mathcal{A}\mathcal{B}^{k+1}|$ are correct. We only demonstrate the algorithm for $\boldsymbol{\tau}_{\mathcal{O},\mathcal{A}\mathcal{B}^k\mathcal{D}}(i)$. Observe that $\varkappa' \in \boldsymbol{\tau}_{\mathcal{O},\mathcal{A}\mathcal{B}^k\mathcal{D}}(i)$ iff $\varkappa' \in \boldsymbol{\tau}_{\mathcal{O},\mathcal{A}\mathcal{B}^k\mathcal{D}}(j)$ iff $\varkappa' \in \boldsymbol{\tau}_1$, for any $\varkappa'$ of the form $\Box\varkappa''$ and $\Diamond\varkappa''$ from $\mathsf{sub}_{\boldsymbol{q}}$ and $|\mathcal{A}|-1 \leq i, j < |\mathcal{A}\mathcal{B}^k|$. To do the required check, we need to guess a sequence of pairs

$$\big(\mathsf{b}(\mathcal{B}^{\leq 0}), \mathsf{b}(\mathcal{B}^{>0})\big), \dots, \big(\mathsf{b}(\mathcal{B}^{\leq |\mathcal{B}|-1}), \mathsf{b}(\mathcal{B}^{>|\mathcal{B}|-1})\big) \tag{37}$$

such that $\mathsf{b}(\mathcal{B}^{\leq i}) \cdot \mathsf{b}(\mathcal{B}^{>i}) = \mathsf{b}(\mathcal{B})$, for all i, and there are $a \in \Sigma_\Xi$ with $\mathsf{b}(\mathcal{B}^{\leq i+1}) = \mathsf{b}(\mathcal{B}^{\leq i}) \cdot \mathsf{b}(a)$ and $\mathsf{b}(\mathcal{B}^{>i}) = \mathsf{b}(a) \cdot \mathsf{b}(\mathcal{B}^{>i+1})$. While we do not have any bound on $|\mathcal{B}|$ yet, we can easily observe that any sequence (37) with repeating members at positions $0 \leq i' < i'' \leq |\mathcal{B}|-1$ is equivalent for the purposes of this proof to the sequence with all the members $i', \dots, i''-1$ removed. Thus, we can assume that $|\mathcal{B}| \leq 2^{O(|\boldsymbol{q}|)}$, if $\mathcal{B}$ exists at all. By (34), using an element i of this sequence, we can compute $\boldsymbol{\tau}^{\mathsf{sig}(\mathcal{O})}_{\mathcal{O},\mathcal{A}\mathcal{B}^k\mathcal{D}}(|\mathcal{A}\mathcal{B}^j| + i)$, for all $j < k$. We only need to check that such an atomic type is not in conflict with the temporal concepts in

$\boldsymbol{\tau}_1$, e.g., $\Box_P A \in \boldsymbol{\tau}_1$ and $\neg A \in \boldsymbol{\tau}^{\mathsf{sig}(\mathcal{O})}_{\mathcal{O},\mathcal{A}\mathcal{B}^k\mathcal{D}}(|\mathcal{A}\mathcal{B}^j| + i)$. If a conflict is detected for some i and j, the algorithm answers no. Here, we also verify that $\boldsymbol{\tau}_{\mathcal{O},\mathcal{A}\mathcal{B}^k\mathcal{D}}(|\mathcal{A}\mathcal{B}^k| - 1) = \boldsymbol{\tau}_1$ and $\boldsymbol{\tau}_{\mathcal{O},\mathcal{A}\mathcal{B}^{k+1}\mathcal{D}}(|\mathcal{A}\mathcal{B}^{k+1}| - 1) = \boldsymbol{\tau}'_1$. Finally, we check that all the types $\boldsymbol{\tau}_{\mathcal{O},\mathcal{A}\mathcal{B}^k\mathcal{D}}(|\mathcal{A}\mathcal{B}^k| + i)$ and $\boldsymbol{\tau}_{\mathcal{O},\mathcal{A}\mathcal{B}^{k+1}\mathcal{D}}(|\mathcal{A}\mathcal{B}^{k+1}| + i)$ are correct, for $0 \le i < N_{\mathcal{D}} - N$. Details are left to the reader. ❑

A criterion for $\mathsf{FO}(<, \equiv)$-definability of linear $\mathit{LTL}^{\Box\bigcirc}_{\mathit{horn}}$ OMPQs (cf. Theorem 6 (*ii*)) is given by the next theorem whose (rather technical) proof can be found in Appendix A.5:

Theorem 26. *Let $\boldsymbol{q} = (\mathcal{O}, \varkappa)$ be an OMPQ with a $\bot$-free $\mathit{LTL}^{\Box\bigcirc}_{\mathit{horn}}$-ontology $\mathcal{O}$. Then $\boldsymbol{q}$ is not $\mathsf{FO}(<, \equiv)$-rewritable over Ξ-ABoxes iff there are $\mathcal{A}, \mathcal{B}, \mathcal{D} \in \Sigma^*_\Xi$ and $k \ge 2$, such that (i) and (ii) from Theorem 24 hold and there are $\mathcal{U}, \mathcal{V} \in \Sigma^*_\Xi$, such that $\mathcal{B} = \mathcal{V}\mathcal{U}$, $|\mathcal{U}| = |\mathcal{V}|$,*

(*iii*) $\boldsymbol{\tau}_{\mathcal{O},\mathcal{A}\mathcal{B}^k\mathcal{D}}(|\mathcal{A}\mathcal{B}^i| - 1) = \boldsymbol{\tau}_{\mathcal{O},\mathcal{A}\mathcal{B}^k\mathcal{D}}(|\mathcal{A}\mathcal{B}^i\mathcal{V}| - 1)$, *for all* $i < k$, *and*

(*iv*) $\boldsymbol{\tau}_{\mathcal{O},\mathcal{A}\mathcal{B}^{k+1}\mathcal{D}}(|\mathcal{A}\mathcal{B}^i| - 1) = \boldsymbol{\tau}_{\mathcal{O},\mathcal{A}\mathcal{B}^{k+1}\mathcal{D}}(|\mathcal{A}\mathcal{B}^i\mathcal{V}| - 1)$, *for all* i, $1 \le i \le k$.

This result allows us to obtain a PSpace algorithm by a straightforward modification of the proof of Theorem 25. Thus, we obtain:

Theorem 27. *Deciding $\mathsf{FO}(<, \equiv)$-rewritability of OMPQs $\boldsymbol{q} = (\mathcal{O}, \varkappa)$ with a linear $\mathit{LTL}^{\bigcirc}_{\mathit{horn}}$-ontology $\mathcal{O}$ over Ξ-ABoxes can be done in* PSpace.

At present, we do not know how to transform Theorem 6 (*iii*) into PSpace-checkable conditions on the canonical models and ABoxes, so the complexity of deciding $\mathsf{FO}(<, \mathsf{MOD})$-rewritability of linear $\mathit{LTL}^{\bigcirc}_{\mathit{horn}}$ OMPQs remains open.

8. $\mathsf{FO}(<)$-rewritability of $\boldsymbol{LTL}^{\bigcirc}_{\boldsymbol{krom}}$ OMAQs and $\boldsymbol{LTL}^{\bigcirc}_{\boldsymbol{core}}$ OMPQs

Our final aim is to look for non-trivial classes of OMQs deciding $\mathsf{FO}(<)$-rewritability of which could be 'easier' than PSpace. Syntactically, the simplest type of axioms (8) are binary clauses $C_1 \to C_2$ and $C_1 \land C_2 \to \bot$, known as *core* axioms, which together with $C_1 \lor C_2$ form the class Krom. In the atemporal case, the W3C standard language *OWL 2 QL*, designed specifically for ontology-based data access, admits core clauses only and uniformly guarantees FO-rewritability (Calvanese et al., 2007; Artale et al., 2009).

In this section, we use NFAs with ε-transitions that can be defined as 2NFAs where backward transitions $q \to_{a,-1} q'$ are disallowed and transitions of the form $q \to_{a,0} q'$ hold for all $a \in \Sigma$, in which case we write $q \to_\varepsilon q'$.

As we saw in the proof of Theorem 19, OMPEQs with disjunctive axioms can simulate $\mathit{LTL}^{\bigcirc}_{\mathit{horn}}$ OMAQs, and so are too complex for the purposes of this section. On the other hand, $\mathit{LTL}^{\bigcirc}_{\mathit{krom}}$ OMAQs and $\mathit{LTL}^{\bigcirc}_{\mathit{core}}$ OMPQs are all $\mathsf{FO}(<, \equiv)$-rewritable (Artale et al., 2021). Below, we focus on deciding $\mathsf{FO}(<)$-rewritability of OMQs in these classes.

8.1 $\boldsymbol{LTL}^{\bigcirc}_{\boldsymbol{krom}}$ OMAQs

Theorem 28. *Deciding $\mathsf{FO}(<)$-rewritability of Boolean and specific $\mathit{LTL}^{\bigcirc}_{\mathit{krom}}$ OMAQs over Ξ-ABoxes is* coNP*-complete.*

Proof. Suppose $\boldsymbol{q} = (\mathcal{O}, A)$ ($\boldsymbol{q}(x) = (\mathcal{O}, A(x))$) is a Boolean (respectively, specific) $LTL^{\bigcirc}_{krom}$ OMAQ. Using the form of Krom axioms, one can show (e.g., Artale et al., 2021) that, for any ABox $\mathcal{A}$ and $l \in \mathbb{Z}$ (respectively, $l \in \mathsf{tem}(\mathcal{A})$), we have $(\mathcal{O}, \mathcal{A}) \models A(l)$ iff at least one of the following holds: (i) there is $B(k) \in \mathcal{A}$ such that $\mathcal{O} \models B \to \bigcirc^{l-k} A$ (the $\bigcirc^n$ notation was defined in Section 7.2); (ii) $\mathcal{O}$ and $\mathcal{A}$ are inconsistent, i.e., there exist $k_1 \leq k_2$, $B(k_1) \in \mathcal{A}$ and $C(k_2) \in \mathcal{A}$ such that $\mathcal{O} \models B \to \bigcirc_F^{k_2-k_1} \neg C$.

Let $\mathit{lit}(\boldsymbol{q}) = \{C, \neg C \mid C \in \mathsf{sig}(\boldsymbol{q})\}$. For any $L_1, L_2 \in \mathit{lit}(\boldsymbol{q})$, we construct a unary NFA $\mathfrak{A}_{L_1L_2}$ of size $O(|\boldsymbol{q}|)$ that accepts the language $\boldsymbol{L}_{L_1L_2} = \{a^n \mid \mathcal{O} \models L_1 \to \bigcirc_F^n L_2,\ n \geq 0\}$ over the alphabet $\{a\}$. The set of its states is $\mathit{lit}(\boldsymbol{q})$, L_1 is the initial state, L_2 the only accepting state, and the transitions are $L \to_a L'$ if $\mathcal{O} \models L \to \bigcirc_F L'$, and $L \to_\varepsilon L'$ if $\mathcal{O} \models L \to L'$. For $\Xi \subseteq \mathsf{sig}(\boldsymbol{q})$, we define two sets: $\Xi_A^\exists = \{B \in \Xi \mid (\mathcal{O}, \{B(0)\}) \models \exists x\, A(x)\}$ and $\Xi_A^\forall = \{B \in \Xi \mid (\mathcal{O}, \{B(0)\}) \models \forall x\, A(x)\}$.

Lemma 29. (a) *The Boolean OMAQ $\boldsymbol{q}$ is* $\mathsf{FO}(<)$*-rewritable over Ξ-ABoxes iff, for any $B, C \in \Xi \setminus \Xi_A^\exists$, the language $\boldsymbol{L}_{B\neg C}$ is* $\mathsf{FO}(<)$*-definable.*

(b) *The specific OMAQ $\boldsymbol{q}(x)$ is* $\mathsf{FO}(<)$*-rewritable over Ξ-ABoxes iff the following conditions are satisfied:*

(b_1) *for all $B \in \Xi$, the languages $\boldsymbol{L}_{BA}$ and $\boldsymbol{L}_{\neg A \neg B}$ are* $\mathsf{FO}(<)$*-definable;*

(b_2) *for all $B, C \in \Xi \setminus \Xi_A^\forall$ such that at least one of the $\boldsymbol{L}_{BA}$ and $\boldsymbol{L}_{\neg A \neg C}$ is finite, the language $\boldsymbol{L}_{B\neg C}$ is* $\mathsf{FO}(<)$*-definable.*

Proof. $(a, \Rightarrow)$ If $\boldsymbol{q}$ is $\mathsf{FO}(<)$-rewritable, then $\boldsymbol{L}_\Xi(\boldsymbol{q})$ over the alphabet Σ_Ξ is $\mathsf{FO}(<)$-definable, and so is the language $\boldsymbol{L}_\Xi(\boldsymbol{q}) \cap \boldsymbol{L}(\{B\}\emptyset^*\{C\})$, for any $B, C \in \Xi$. For $B, C \in \Xi \setminus \Xi_A^\exists$, we have $\{B\}\emptyset^n\{C\} \in \boldsymbol{L}_\Xi(\boldsymbol{q})$ iff $\mathcal{O} \models B \to \bigcirc_F^{n+1} \neg C$ iff $a^{n+1} \in \boldsymbol{L}_{B\neg C}$. Therefore, $\boldsymbol{L}_{B\neg C}$ is $\mathsf{FO}(<)$-definable.

$(a, \Leftarrow)$ For a Ξ-ABox $\mathcal{A}$, the certain answer to $\boldsymbol{q}$ is yes iff either there is $B(k) \in \mathcal{A}$, for some $B \in \Xi_A^\exists$, or there are $B, C \in \Xi \setminus \Xi_A^\exists$ and $k \leq l$ such that $B(k), C(l) \in \mathcal{A}$ and $\mathcal{O} \models B \to \bigcirc_F^{k-l} \neg C$. As these conditions are $\mathsf{FO}(<)$-definable, $\boldsymbol{q}$ is $\mathsf{FO}(<)$-rewritable.

$(b, \Rightarrow)$ If $\boldsymbol{q}(x)$ is $\mathsf{FO}(<)$-rewritable, then $\boldsymbol{L}_\Xi(\boldsymbol{q}(x))$ over the alphabet Γ_Ξ is $\mathsf{FO}(<)$-definable, and so are the languages $\boldsymbol{L}_\Xi(\boldsymbol{q}(x)) \cap \boldsymbol{L}(\{B\}\emptyset^*\emptyset')$ and $\boldsymbol{L}_\Xi(\boldsymbol{q}(x)) \cap \boldsymbol{L}(\emptyset'\emptyset^*\{B\})$, for any $B \in \Xi$. We have $\{B\}\emptyset^n\emptyset' \in \boldsymbol{L}_\Xi(\boldsymbol{q}(x))$ iff $\mathcal{O} \models B \to \bigcirc_F^{n+1} A$ and $\emptyset'\emptyset^*\{B\} \in \boldsymbol{L}_\Xi(\boldsymbol{q}(x))$ iff $\mathcal{O} \models B \to \bigcirc_P^{n+1} A$. Therefore, $\boldsymbol{L}_{BA}$ and $\boldsymbol{L}_{\neg A \neg B}$ are $\mathsf{FO}(<)$-definable.

Let $B, C \in \Xi \setminus \Xi_A^\forall$ and $\boldsymbol{L}_{BA}$ be finite. There is $l \in \mathbb{Z}$ with $(\mathcal{O}, \{C(0)\}) \not\models A(l)$ and there is k with $k > n$ for all $a^n \in \boldsymbol{L}_{BA}$. For $m > k + |l|$, we have $(\mathcal{O}, \{B(0), C(m)\}) \models A(m+l)$ iff $\mathcal{O} \models B \to \bigcirc_F^m \neg C$. So $\boldsymbol{L}_{B\neg C}$ is $\mathsf{FO}(<)$-definable. The case when $\boldsymbol{L}_{\neg A \neg C}$ is finite is similar.

$(b, \Leftarrow)$ Assuming that conditions (b_1) and (b_2) hold, we define formulas $\varphi_{B\neg C}$, for any $B, C \in \Xi$. If $\boldsymbol{L}_{B\neg C}$ is $\mathsf{FO}(<)$-definable, then set $\varphi_{B\neg C} = \exists x, y\, (B(x) \land C(y) \land \psi(x,y))$, where $\psi(x,y)$ is an $\mathsf{FO}(<)$-formula saying that $a^{y-x} \in \boldsymbol{L}_{B\neg C}$. Suppose $B, C \notin \Xi_A^\forall$ and $\boldsymbol{L}_{B\neg C}$ is not $\mathsf{FO}(<)$-definable. It follows from (b_2) that both $\boldsymbol{L}_{BA}$ and $\boldsymbol{L}_{\neg A \neg C}$ are infinite. By (b_1) and the folklore fact that every star-free language over a unary alphabet is either finite or cofinite, we have $n_1, n_2 \in \mathbb{N}$ such that $a^k \in \boldsymbol{L}_{BA}$ for all $k \geq n_1$ and $a^k \in \boldsymbol{L}_{\neg A \neg C}$ for all $k \geq n_2$. Then we set $\varphi_{B\neg C} = \exists x, y\, (B(x) \land C(y) \land \psi(x,y))$, where $\psi(x,y)$ is an $\mathsf{FO}(<)$-formula saying that $y - x < n_1 + n_2$ and $a^{y-x} \in \boldsymbol{L}_{A\neg C}$. Finally, for $B, C \in \Xi$ such that either B or C is not in $\Xi_A^\forall$ and $\boldsymbol{L}_{B\neg C}$ is not $\mathsf{FO}(<)$-definable, we set $\varphi_{B\neg C} = \bot$. For $B \in \Xi$, let $\varphi_{BA}(x) = \exists y\, (B(y) \land \psi(y,x))$, where $\psi(y,x)$ is an $\mathsf{FO}(<)$-formula saying that

$a^{x-y} \in \boldsymbol{L}_{BA}$, which exists by (b_1). Similarly, let $\varphi_{\neg A \neg B}(x) = \exists y\,(B(y) \wedge \psi(y,x))$, where $\psi(y,x)$ is an $\mathsf{FO}(<)$-formula saying that $a^{y-x} \in \boldsymbol{L}_{\neg A \neg B}$. We claim that

$$\varphi(x) = \bigvee_{B \in \Xi} (\varphi_{BA}(x) \vee \varphi_{\neg A \neg B}(x)) \vee \bigvee_{B,C \in \Xi} \varphi_{B \neg C}$$

is an $\mathsf{FO}(<)$-rewriting of $\boldsymbol{q}(x)$ over Ξ-ABoxes. Indeed, let $(\mathcal{O},\mathcal{A}) \models A(l)$ for $l \in \mathsf{tem}(\mathcal{A})$. If (i) at the beginning of the proof of Theorem 28 holds, then we have $\mathfrak{S}_{\mathcal{A}} \models \varphi_{BA}(l)$ or $\mathfrak{S}_{\mathcal{A}} \models \varphi_{\neg A \neg B}(l)$, so $\mathfrak{S}_{\mathcal{A}} \models \varphi(l)$. If (ii) holds, consider the B and C given by it. If $\boldsymbol{L}_{B \neg C}$ is $\mathsf{FO}(<)$-definable, then $\mathfrak{S}_{\mathcal{A}} \models \varphi_{B \neg C}$ and $\mathfrak{S}_{\mathcal{A}} \models \varphi(l)$ as required. If $\boldsymbol{L}_{B \neg C}$ is not $\mathsf{FO}(<)$-definable and $B, C \notin \Xi^{\forall}_{A}$, consider $B(k_1) \in \mathcal{A}$ and $C(k_2) \in \mathcal{A}$ such that $k_2 - k_1 \in \boldsymbol{L}_{B \neg C}$ given by (ii). If $k_2 - k_1 < n_1 + n_2$, then $\mathfrak{S}_{\mathcal{A}} \models \varphi_{B \neg C}$ and $\mathfrak{S}_{\mathcal{A}} \models \varphi(l)$ as required. If, on the contrary, $k_2 - k_1 \geq n_1 + n_2$, we have either $l - k_1 \in \boldsymbol{L}_{BA}$ or $k_2 - l \in \boldsymbol{L}_{\neg A \neg B}$. Then $\mathfrak{S}_{\mathcal{A}} \models \varphi_{BA}(l)$ or $\mathfrak{S}_{\mathcal{A}} \models \varphi_{\neg A \neg B}(l)$, so $\mathfrak{S}_{\mathcal{A}} \models \varphi(l)$. Finally, if either B or C is in $\Xi^{\forall}_{A}$, we have either $\mathfrak{S}_{\mathcal{A}} \models \varphi_{BA}$ or $\mathfrak{S}_{\mathcal{A}} \models \varphi_{CA}$ or $\mathfrak{S}_{\mathcal{A}} \models \varphi_{\neg A \neg B}$ or $\mathfrak{S}_{\mathcal{A}} \models \varphi_{\neg A \neg C}$, so $\mathfrak{S}_{\mathcal{A}} \models \varphi(l)$. The proof that $\mathfrak{S}_{\mathcal{A}} \models \varphi(l)$ implies $(\mathcal{O},\mathcal{A}) \models A(l)$ is similar and left to the reader. ❑

Thus, to check $\mathsf{FO}(<)$-rewritability of $\boldsymbol{q}$ and $\boldsymbol{q}(x)$, it suffices to check $\mathsf{FO}(<)$-definability, emptiness and finiteness of the languages of the form $\boldsymbol{L}_{L_1 L_2}$, for $L_1, L_2 \in \mathsf{lit}(\boldsymbol{q})$. Emptiness and finiteness can be checked in NL in the size of $\mathfrak{A}_{L_1 L_2}$. Using Stockmeyer & Meyer's (1973, Theorem 6.1), one can show that deciding $\mathsf{FO}(<)$-definability of the language of a unary NFA is coNP-complete, which gives the required upper bound. To establish coNP-hardness, for any given unary NFA $\mathfrak{A} = (Q, \{a\}, \delta, Q_0, F)$ with $Q = \{Q_0, \dots, Q_n\}$, we define an $LTL^{\bigcirc}_{core}$ ontology $\mathcal{O}_{\mathfrak{A}}$ with $\mathsf{sig}(\mathcal{O}_{\mathfrak{A}}) = Q \cup \{X, Y\}$ and the axioms $X \to \bigcirc_{F} Q_0$, $Q_i \wedge Y \to \bot$, for every $Q_i \in F$, and $Q_i \to \bigcirc_{F} Q_j$, for every transition $Q_i \to_a Q_j$ in $\mathfrak{A}$. Let A be a fresh concept name. The OMAQ $\boldsymbol{q} = (\mathcal{O}_{\mathfrak{A}}, A)$ (respectively, $\boldsymbol{q}(x) = (\mathcal{O}_{\mathfrak{A}}, A(x))$) is $\mathsf{FO}(<)$-rewritable over $\{X, Y\}$-ABoxes iff $\boldsymbol{L}(\mathfrak{A})$ is $\mathsf{FO}(<)$-definable because $(\mathcal{O},\mathcal{A}) \models A(l)$ for some $l \in \mathbb{Z}$ (respectively, $l \in \mathsf{tem}(\mathcal{A})$), for an $\{X, Y\}$-ABox $\mathcal{A}$, iff $\mathcal{A}$ is inconsistent with $\mathcal{O}_{\mathfrak{A}}$ iff there are $X(i), Y(j) \in \mathcal{A}$ with $a^{j-i-1} \in \boldsymbol{L}(\mathfrak{A})$. ❑

Our next result deals with a weaker (core) ontology language but more expressive queries.

8.2 $LTL^{\bigcirc}_{core}$ OMPEQs

Theorem 30. *Deciding $\mathsf{FO}(<)$-rewritability of Boolean and specific $LTL^{\bigcirc}_{core}$ OMPEQs over Ξ-ABoxes is Π^p_2-complete.*

Proof. By Proposition 21 (ii) and Lemma 20, it suffices to consider Boolean OMPEQs $\boldsymbol{q} = (\mathcal{O}, \varkappa)$ with a $\bot$-free $\mathcal{O}$. Also, for the same technical reasons as in Section 7.3, we can assume that $\varkappa$ takes the form $\Diamond_P \Diamond_F \varkappa'$.

We first observe that checking $\mathsf{FO}(<)$-definability of $\boldsymbol{L}_{\Xi}(\boldsymbol{q})$ can be reduced to checking $\mathsf{FO}(<)$-definability of finitely many simpler languages. For $n \geq 0$, let

$$W_{n,\Xi} = \{a_1 \dots a_k \in \Sigma^*_{\Xi} \mid |a_i| \geq 1,\ \sum_{i=1}^{k} |a_i| \leq n\}.$$

With each $\mathcal{B} = a_1 \dots a_k \in W_{|\varkappa|,\Xi}$ we associate the languages

$$\boldsymbol{L}^1_{\mathcal{B}} = \boldsymbol{L}((\emptyset^* a_1) \dots (\emptyset^* a_k) \emptyset^*) \quad \text{and} \quad \boldsymbol{L}_{\mathcal{B}} = \boldsymbol{L}^1_{\mathcal{B}} \cap \boldsymbol{L}_{\Xi}(\boldsymbol{q}).$$

For $\mathcal{U} = u_1 \dots u_k$ and $\mathcal{V} = v_1 \dots v_l$ in Σ_Ξ^*, we write $\mathcal{U} \leq \mathcal{V}$ if $k = l$ and $u_i \subseteq v_i$, for all i. Let $\boldsymbol{L}_\mathcal{B}^\uparrow = \{\mathcal{V} \in \Sigma_\Xi^* \mid \exists \mathcal{U} \in \boldsymbol{L}_\mathcal{B}\, \mathcal{U} \leq \mathcal{V}\}$. We show that

$$\boldsymbol{L}_\Xi(\boldsymbol{q}) \;=\; \bigcup_{\mathcal{B} \in W_{|\varkappa|,\Xi}} \boldsymbol{L}_\mathcal{B}^\uparrow. \tag{38}$$

Let $\mathcal{A} \in \boldsymbol{L}_\Xi(\boldsymbol{q})$, and so $(\mathcal{O}, \mathcal{A}) \models \exists x\, \varkappa(x)$. Observe that, for any $\mathcal{A}$ and $j \in \mathbb{Z}$, $(\mathcal{O}, \mathcal{A}) \models \varkappa(j)$ iff $(\mathcal{O}, \mathcal{A}') \models \varkappa(j)$, for some $\mathcal{A}' \subseteq \mathcal{A}$ with $|\mathcal{A}| \leq |\varkappa|$. The latter statement is shown by induction on the construction of $\varkappa$, where the base case $\varkappa = A$ follows from the proof of Theorem 28, and left to the reader. This observation implies that $(\mathcal{O}, \mathcal{A}') \models \exists x\, \varkappa(x)$ for some Ξ-ABox $\mathcal{A}' \subseteq \mathcal{A}$ with $|\mathcal{A}'| \leq |\varkappa|$. Let $\mathcal{B}$ be the result of removing all $\emptyset$ from $\mathcal{A}'$ (viewed as a word). Clearly, $\mathcal{B} \in W_{|\varkappa|,\Xi}$ and $\mathcal{A} \in \boldsymbol{L}_\mathcal{B}^\uparrow$. The converse inclusion follows from the fact that $\mathcal{A} \in \boldsymbol{L}_\Xi(\boldsymbol{q})$ implies $\mathcal{A}' \in \boldsymbol{L}_\Xi(\boldsymbol{q})$ for any $\mathcal{A} \subseteq \mathcal{A}'$.

Lemma 31. *The language $\boldsymbol{L}_\Xi(\boldsymbol{q})$ is $\mathsf{FO}(<)$-definable iff $\boldsymbol{L}_\mathcal{B}$ is $\mathsf{FO}(<)$-definable, for every $\mathcal{B} \in W_{|\varkappa|,\Xi}$.*

Proof. ($\Rightarrow$) If $\boldsymbol{L}_\Xi(\boldsymbol{q})$ is $\mathsf{FO}(<)$-definable, then so is $\boldsymbol{L}_\mathcal{B}$ as $\boldsymbol{L}_\mathcal{B}^1$ is $\mathsf{FO}(<)$-definable.

($\Leftarrow$) Suppose $\boldsymbol{L}_\mathcal{B}$ is $\mathsf{FO}(<)$-definable for any $\mathcal{B} \in W_{|\varkappa|,\Xi}$. By (38), it suffices to prove that $\boldsymbol{L}_\mathcal{B}^\uparrow$ is $\mathsf{FO}(<)$-definable. For $0 \leq l \leq k$, let $\boldsymbol{L}_{\mathcal{B},l}^1 = \boldsymbol{L}\big((\emptyset^* a_{k-l+1}) \dots (\emptyset^* a_k)\emptyset^*\big)$. Note that $\boldsymbol{L}_{\mathcal{B},0}^1 = \boldsymbol{L}(\emptyset^*)$ and $\boldsymbol{L}_{\mathcal{B},k}^1 = \boldsymbol{L}_\mathcal{B}^1$. We prove by induction on l that, for any $\boldsymbol{L} \subseteq \boldsymbol{L}_{\mathcal{B},l}^1$, if $\boldsymbol{L}$ is $\mathsf{FO}(<)$-definable, then $\boldsymbol{L}^\uparrow$ is $\mathsf{FO}(<)$-definable. Let $l = 0$ and suppose $\boldsymbol{L}$ is $\mathsf{FO}(<)$-definable. Then $\boldsymbol{L}_{\mathcal{B},l}^1 = \boldsymbol{L}(\emptyset^*)$, and so $\boldsymbol{L}$ is a finite or cofinite subset of $\boldsymbol{L}(\emptyset^*)$. Either way the language $\boldsymbol{L}^\uparrow$ is $\mathsf{FO}(<)$-definable. Now, suppose $l > 0$ and $\boldsymbol{L} \subseteq \boldsymbol{L}_{\mathcal{B},l}^1$ is $\mathsf{FO}(<)$-definable. Let $\mathfrak{A} = (Q, \Gamma_\Xi, \delta, q_0, F)$ be a minimal DFA accepting $\boldsymbol{L}$. Let $Q_\emptyset = \{q \in Q \mid \exists i\, \delta_{\emptyset^i}(q_0) = q\}$. For $p \in Q_\emptyset$, let $\boldsymbol{L}_p$ be the language accepted by the automaton $(Q_\emptyset, \{\emptyset\}, \delta|_{Q_\emptyset}, \{q_0\}, \{p\})$ and let $\boldsymbol{L}'_p$ be the language accepted by the automaton $(Q \setminus Q_\emptyset, \Gamma_\Xi, \delta|_{Q \setminus Q_\emptyset}, \delta_{\boldsymbol{a}_{k-l+1}}(p), F)$. Clearly, $\boldsymbol{L}'_p \subseteq \boldsymbol{L}_{\mathcal{B},l-1}^1$ and both $\boldsymbol{L}_p$ and $\boldsymbol{L}'_p$ are $\mathsf{FO}(<)$-definable. Since $\boldsymbol{L}'_p \subseteq \boldsymbol{L}(\emptyset^*)$ and by IH, the languages $\boldsymbol{L}_p^\uparrow$ and $\boldsymbol{L}'^\uparrow_p$ are $\mathsf{FO}(<)$-definable, and so $\boldsymbol{L}^\uparrow = \bigcup_{p \in Q_\emptyset} (\boldsymbol{L}_p^\uparrow \cdot (\bigcup_{\boldsymbol{a} \supseteq \boldsymbol{a}_{k-l+1}} \{\boldsymbol{a}\}) \cdot \boldsymbol{L}'^\uparrow_p)$ is $\mathsf{FO}(<)$-definable as well. ❑

Now we give a criterion of checking $\mathsf{FO}(<)$-definability of $\boldsymbol{L}_w$ (cf. Theorem 24).

Lemma 32. *Let $w = a_1 \dots a_k \in W_{|\varkappa|,\Xi}$. Then $\boldsymbol{L}_w$ is not $\mathsf{FO}(<)$-definable iff there are words $\mathcal{A} = (\emptyset^{i_1} a_1) \dots (\emptyset^{i_{l-1}} a_{l-1})\emptyset^{i_l}$, $\mathcal{D} = (\emptyset^{i'_l} a_l)(\emptyset^{i_{l+1}} a_{l+1}) \dots (\emptyset^{i_k} a_k)\emptyset^{i_{k+1}}$, $\mathcal{B} = \emptyset^n$ and $k \geq 2$ such that (i) and (ii) from Theorem 24 hold. Moreover, we can find $\mathcal{A}, \mathcal{B}, \mathcal{D}$ and k such that $|\mathcal{A}|, |\mathcal{B}|, |\mathcal{D}|, k \leq 2^{O(|\boldsymbol{q}|)}$.*

Proof. We only outline modifications needed to the proof of Theorem 24 to obtain this result and the specific form of $\mathcal{A}$, $\mathcal{B}$ and $\mathcal{D}$. Consider the automaton $\mathfrak{A}$ defined in the proof of Theorem 24 and denote by $\mathfrak{A}_j$, for $1 \leq j \leq k+1$, a copy of $\mathfrak{A}$ restricted to the alphabet $\emptyset$. We construct an automaton $\mathfrak{A}_w$ by taking a disjoint union of all the $\mathfrak{A}_j$ and adding a transition $q \to_{a_j} q'$, for $1 \leq j \leq k$, from q in $\mathfrak{A}_j$ to $q' \in \mathfrak{A}_{j+1}$ for each pair (q, q') such that $\mathfrak{A}$ contains an a_j-transition from the original of q to the original of q'. The initial state of $\mathfrak{A}_w$ is q_{-1} from $\mathfrak{A}_1$ and the final states are those in $\mathfrak{A}_{k+1}$. It is straightforward to see that $\boldsymbol{L}_w = \boldsymbol{L}(\mathfrak{A}_w)$. The proof of Theorem 24 works for $\mathfrak{A}_w$ in place of $\mathfrak{A}$. That $\mathcal{B}$ consists of $\emptyset$ only follows from the fact that non-trivial cycles in $\mathfrak{A}_w$ can only be with $\emptyset$-symbols. ❑

We observe that (binary encoding of) $\mathcal{AB}^k\mathcal{D}$ and $\mathcal{AB}^{k+1}\mathcal{D}$ in the lemma above can be guessed and stored in polynomial time. Thus, it remains to show that conditions (i) and (ii) from Theorem 24 can be checked by an NP-oracle. The (more or less standard) proof of the following lemma is given in Appendix A.6.

Lemma 33. *Given $a_1, \dots, a_l \in \Sigma_\Xi$ with $|a_i| = 1$, for $1 \leq i \leq l$, binary numbers $i_1, \dots, i_{l+1}, j$, a $\bot$-free $LTL^{\bigcirc}_{core}$-ontology $\mathcal{O}$ and a positive existential temporal concept $\varkappa$, checking whether $\mathcal{C}_{\mathcal{O},\mathcal{A}} \models \varkappa(j)$ for $\mathcal{A} = \emptyset^{i_1} a_1 \dots \emptyset^{i_l} a_l \emptyset^{i_{l+1}}$ can be done in* NP.

We can now complete the proof of the upper bound. By Lemmas 31 and 32, $\boldsymbol{L}_\Xi(\boldsymbol{q})$ is not $\mathsf{FO}(<)$-definable iff there exist $a_1 \dots a_k \in W_{|\varkappa|,\Xi}$, $\mathcal{A} = (\emptyset^{i_1} a_1) \dots (\emptyset^{i_{l-1}} a_{l-1})\emptyset^{i_l}$, $\mathcal{D} = (\emptyset^{i'_l} a_l)(\emptyset^{i_{l+1}} a_{l+1}) \dots (\emptyset^{i_k} a_k)\emptyset^{i_{k+1}}$, $\mathcal{B} = \emptyset^n$, $k \geq 2$, such that $|\mathcal{A}|, |\mathcal{B}|, |\mathcal{D}|, k \leq 2^{O(|\boldsymbol{q}|)}$, (i) $\neg\varkappa \in \boldsymbol{\tau}_{\mathcal{O},\mathcal{AB}^k\mathcal{D}}(|\mathcal{A}| - 1) = \boldsymbol{\tau}_{\mathcal{O},\mathcal{AB}^k\mathcal{D}}(|\mathcal{AB}^k| - 1)$; and (ii) $\varkappa \in \boldsymbol{\tau}_{\mathcal{O},\mathcal{AB}^{k+1}\mathcal{D}}(|\mathcal{AB}| - 1) = \boldsymbol{\tau}_{\mathcal{O},\mathcal{AB}^{k+1}\mathcal{D}}(|\mathcal{AB}^{k+1}| - 1)$. We can check non-$\mathsf{FO}(<)$-definability of $\boldsymbol{L}_\Xi(\boldsymbol{q})$ by guessing the required $\mathcal{A}$, $\mathcal{B}$, $\mathcal{D}$, k and the four types involved in the conditions (i) and (ii). That the four types are indeed correct can be checked in polynomial time by using the NP-oracle provided by Lemma 33.

The proof of the matching lower bound is by reduction of $\forall\exists$CNF—the satisfiability problem for fully quantified Boolean formulas in CNF with the prefix $\forall\exists$—which is known to be Π^p_2-complete (e.g., Arora & Barak, 2009). By Lemma 20 and Proposition 21 (ii), we can only consider specific OMQs. Suppose we are given a closed QBF $\varphi = \forall X_1 \dots \forall X_n \exists Y_1 \dots \exists Y_m\, \psi = \forall \boldsymbol{x} \exists \boldsymbol{y}\, \psi$ with a CNF ψ. Define an $LTL^{\bigcirc}_{core}$ OMPEQ $\boldsymbol{q}_\varphi(x) = (\mathcal{O}_\varphi, \varkappa_\varphi(x))$ and Ξ such that $\boldsymbol{q}_\varphi(x)$ is $\mathsf{FO}(<)$-rewritable over Σ-ABoxes iff $\forall \boldsymbol{x} \exists \boldsymbol{y}\, \psi$ is true. Let Ξ consist of the atomic concepts A, B, A^j_i, for $1 \leq i \leq m$, $0 \leq j \leq p_i - 1$, where p_i is the i-th prime number, X^0_k, X^1_k, for $1 \leq k \leq n$, Y^0_i, Y^1_i, for $1 \leq i \leq m$. The ontology $\mathcal{O}_\varphi$ has the following axioms for all such i and k:

$$
\begin{array}{lllll}
A \to A^0_i, & A^j_i \to \bigcirc_F A_i^{(j+1) \bmod p_i}, & \text{for } 0 \leq j \leq p_i - 1, & & \\
A^0_i \to Y^0_i, & A^1_i \to Y^1_i, & X^0_k \to \bigcirc_F X^0_k, & X^1_k \to \bigcirc_F X^1_k, & B \to \bigcirc_F \bigcirc_F B.
\end{array}
$$

The size of $\mathcal{O}_\varphi$ is polynomial in $n + m$. Let ψ' be the result of replacing all X_i in ψ with X^1_i, all $\neg X_i$ with X^0_i, and similarly for the Y_i. We set

$$
\varkappa_\varphi = A \wedge \bigwedge_{i=0}^{n} (X^0_i \vee X^1_i) \wedge (\bigcirc_P B \vee \Diamond_F \psi').
$$

To show that $\boldsymbol{q}_\varphi(x)$ is as required, suppose $\forall \boldsymbol{x} \exists \boldsymbol{y}\, \psi$ is true. Let $(\mathcal{O}_\varphi, \mathcal{A}) \models \varkappa_\varphi(t)$, for some $\mathcal{A}$ and t. Then $A(t) \in \mathcal{A}$ and $(\mathcal{O}_\varphi, \mathcal{A}) \models \bigwedge_{i=0}^n (X^0_i \vee X^1_i)(t)$. So, for any i, there is $s \leq t$ with $X^0_i(s) \in \mathcal{A}$ or $X^1_i(s) \in \mathcal{A}$. Let $\mathfrak{a}_1 \colon \{X_1, \dots, X_n\} \to \{0, 1\}$ be such that $(\mathcal{O}_\varphi, \mathcal{A}) \models X_i^{\mathfrak{a}_1(X_i)}(s)$ for all $s > t$ and i. Take an assignment $\mathfrak{a}_2 \colon \{Y_1, \dots, Y_m\} \to \{0, 1\}$ that makes ψ true. There is a number $r > 0$ such that $r = \mathfrak{a}_2(Y_i) \pmod{p_i}$ for all i. Then $(\mathcal{O}_\varphi, \mathcal{A}) \models Y_i^{\mathfrak{a}_2(i)}(t + r)$, $(\mathcal{O}_\varphi, \mathcal{A}) \models \psi'(t + r)$, and so $(\mathcal{O}_\varphi, \mathcal{A}) \models \Diamond_F \psi'(t)$. Thus, the sentence

$$
A(x) \wedge \bigwedge_{i=0}^{n} \exists s_i \left((s_i \leqslant x) \wedge (X^0_i(s_i) \vee X^1_i(s_i)) \right)
$$

is an $\mathsf{FO}(<)$-rewriting of $\boldsymbol{q}_\varphi(x)$ over Ξ-ABoxes.

If $\forall \boldsymbol{x} \exists \boldsymbol{y}\, \psi$ is false, there is an assignment $\mathfrak{a}\colon \{X_1, \dots, X_n\} \to \{0,1\}$ such that ψ is false under any assignment of the Y_i. Suppose $\boldsymbol{L}_\Xi(\boldsymbol{q}_\varphi(x))$ over Γ_Ξ is $\mathsf{FO}(<)$-definable. Let $a_\mathfrak{a} = \{A\} \cup \bigcup_{i=1}^n \{X_i^{\mathfrak{a}(X_i)}\} \in \Sigma_\Xi$. Consider $\mathcal{A}_l = \{B(0)\} \cup \bigcup_{Z \in a_\mathfrak{a}} \{Z(l)\}$ for some $l > 0$. Observe that, since $(\mathcal{O}_\varphi, \mathcal{A}) \not\models \Diamond_F \psi'(l)$, l is a certain answer to $\boldsymbol{q}_\varphi(x)$ over $\mathcal{A}_l$ iff $(\mathcal{O}_\varphi, \mathcal{A}) \models B(l-1)$. It follows that $\boldsymbol{L}(\{B\}(\emptyset\emptyset)^* a'_\mathfrak{a}) = \boldsymbol{L}_\Xi(\boldsymbol{q}_\varphi(x)) \cap \boldsymbol{L}(\{B\}\emptyset^* a'_\mathfrak{a})$ (recall that $a' \in \Gamma_\Xi$ for each $a \in \Sigma_\Xi$). Clearly, $\boldsymbol{L}(\{B\}\emptyset^* a'_\mathfrak{a})$ is $\mathsf{FO}(<)$-definable, and so $\boldsymbol{L}(\{B\}(\emptyset\emptyset)^* a'_\mathfrak{a})$ is $\mathsf{FO}(<)$-definable, which is not the case (Straubing, 1994, Theorem IV.2.1). ❑

8.3 $\boldsymbol{LTL}_{core}^{\bigcirc}$ OMPQs

If we increase the expressive power of $LTL_{core}^{\bigcirc}$ OMPEQs $\boldsymbol{q} = (\mathcal{O}, \varkappa)$ by allowing $\Box$-operators in $\varkappa$, the problem of deciding $\mathsf{FO}(<)$-rewritability becomes PSPACE-complete, as established by Theorem 35 below. The upper bound follows from Theorem 25 and the next observation showing that, even though core disjointness constraints $C_1 \wedge C_2 \to \bot$ may have IDB concepts C_1 and C_2, there is always an equivalent linear $LTL_{horn}^{\bigcirc}$ ontology.

Proposition 34. *For any Boolean (specific) $LTL_{core}^{\bigcirc}$ OMPQ and any signature Ξ, one can construct in polynomial time a Ξ-equivalent Boolean (specific) linear $LTL_{horn}^{\bigcirc}$ OMPQ.*

Proof. We only consider Boolean OMQs $\boldsymbol{q} = (\mathcal{O}, \varkappa)$ as the case of specific ones is similar. First, for each atom A in $\mathcal{O}$, we introduce a fresh atom $\bar{A}$ and, for each axiom $C_1 \to C_2$ in $\mathcal{O}$, we add to $\mathcal{O}$ the axiom $\bar{C}_2 \to \bar{C}_1$, where $\bar{C}$ is the result of replacing A in C by $\bar{A}$; we also replace each axiom $C_1 \wedge C_2 \to \bot$ in $\mathcal{O}$ with $C_1 \to \bar{C}_2$. Then we rename each atom A ($\bar{A}$) in $\boldsymbol{q} = (\mathcal{O}, \varkappa)$ to A' (respectively, $\bar{A}'$), for a fresh A' (respectively, $\bar{A}'$). Denote by C' the temporal concept obtained by replacing A by A' in C. Finally, we add the axioms $A \to A'$ and $A \wedge \bar{A}' \to \bot$ to $\mathcal{O}$, for $A \in \Xi$, denoting the result by $\boldsymbol{q}' = (\mathcal{O}', \varkappa')$. It is easy to see that $\boldsymbol{q}'$ is linear because all the IDB atoms of $\mathcal{O}'$ are of the form A' or $\bar{A}'$. For example, let $\mathcal{O} = \{\bigcirc_P A \to B,\, \bigcirc_F D \to C,\, C \wedge B \to \bot\}$ and $\Xi = \{A, B, D\}$. Then $\mathcal{O}'$ contains the axioms $\bigcirc_P A' \to B'$, $\bar{B}' \to \bigcirc_P \bar{A}'$, $\bigcirc_F D' \to C'$, $\bar{C}' \to \bigcirc_F \bar{D}'$, $C' \to \bar{B}'$ together with $X \to X'$ and $X \wedge \bar{X}' \to \bot$, for each $X \in \Xi$. Clearly, $\boldsymbol{q}$ and $\boldsymbol{q}'$ are Ξ-equivalent. For example, over $\mathcal{A} = \{A(0), D(2)\}$, both $\boldsymbol{q}$ and $\boldsymbol{q}'$ return yes as both $(\mathcal{O}, \mathcal{A})$ and $(\mathcal{O}', \mathcal{A})$ are inconsistent. ❑

Theorem 35. *Deciding $\mathsf{FO}(<)$-rewritability of Boolean and specific $LTL_{core}^{\bigcirc}$ OMPQs is* PSPACE*-complete*

Proof. The upper bound follows from Proposition 34 and Theorem 25. To prove the lower one, we reduce the PSPACE-complete problem of deciding the emptiness of the intersection of a set of DFAs (Kozen, 1977) to OMPQ rewritability. Let $\mathfrak{A}_1, \dots, \mathfrak{A}_n$ with $\mathfrak{A}_i = (Q_i, \Sigma, \delta^i, q_0^i, F_i)$ be a sequence of DFAs that do not accept the empty word, have a common input alphabet, and disjoint sets $Q_i = \{q_0^i, \dots, q_{j_i}^i\}$ of states.

Let ∇_i be the set of atoms $N_{q,a,r}^i$, for $q, r \in Q_i$, $a \in \Sigma$, such that $\delta_a^i(q) = r$. Consider the ontology $\mathcal{O}$ with atomic concepts $\{X, Y, B\} \cup \bigcup_{1 \le i \le n} \nabla_i$ and the following axioms, for

$1 \leq i, l \leq n$, $q, r, s, t \in Q_i$, $q', r' \in Q_l$, $a, b \in \Sigma$:

(1) $N^i_{q,a,r} \wedge N^l_{q',b,r'} \to \bot$, if either $a \neq b$, or $i = l$ and $(q, r) \neq (q', r')$;

(2) $N^i_{q,a,r} \wedge \bigcirc_F N^i_{s,b,t} \to \bot$, if $r \neq s$;

(3) $X \wedge \bigcirc_F N^i_{q,a,r} \to \bot$, if $q \neq q^i_0$;

(4) $N^i_{q,a,r} \wedge \bigcirc_F Y \to \bot$, if $r \notin F_i$;

(5) $X \wedge \bigcirc_F Y \to \bot$;

(6) $Y \to \bigcirc_F Y$;

(7) $B \to \bigcirc_F \bigcirc_F B$.

Let

$$\varkappa \;=\; \bigcirc_P B \wedge X \wedge \Box_F \Big(\big(\bigwedge_{1 \leq i \leq n} \bigvee_{\delta^i_a(q)=r} N^i_{q,a,r} \big) \vee Y \Big).$$

We claim that the OMPQs $\boldsymbol{q}(x) = (\mathcal{O}, \varkappa(x))$ and $\boldsymbol{q} = (\mathcal{O}, \varkappa)$ are $\mathsf{FO}(<)$-rewritable over Ξ-ABoxes, for $\Xi = \mathsf{sig}(\boldsymbol{q})$, iff $\bigcap_{1 \leq i \leq n} \boldsymbol{L}(\mathfrak{A}_i) = \emptyset$.

($\Leftarrow$) If $\bigcap_{1 \leq i \leq n} \boldsymbol{L}(\mathfrak{A}_i) = \emptyset$, then, for any Ξ-ABox $\mathcal{A}$ and $k \in \mathsf{tem}(\mathcal{A})$, we have $\mathcal{O}, \mathcal{A} \models \varkappa(k)$ iff $\mathcal{A}$ is inconsistent with $\mathcal{O}$ because the formula $X \wedge \Box_F \Big(\big(\bigwedge_{1 \leq i \leq n} \bigvee_{\delta^i_a(q)=r} N^i_{q,a,r} \big) \vee Y \Big)$ cannot be true at any place in a consistent ABox. Let φ be the disjunction of the formulas $\exists x\, (C(x) \wedge D(x))$, for all axioms $C \wedge D \to \bot$ of the form (1), the formulas $\exists x\, (C(x) \wedge D(x+1))$, for all axioms $C \wedge \bigcirc_F D \to \bot$ of the forms (2) and (3), and $\exists x, y\, ((y < x+2) \wedge Y(y) \wedge C(x))$, for all axioms $C \wedge \bigcirc_F Y \to \bot$ of the forms (4) and (5). Then $(x = x) \wedge \varphi$ is an $\mathsf{FO}(<)$-rewriting of $\boldsymbol{q}(x)$, and φ is an $\mathsf{FO}(<)$-rewriting of $\boldsymbol{q}$.

($\Rightarrow$) Let $w = a_1 \dots a_k \in \bigcap_{1 \leq i \leq n} \boldsymbol{L}(\mathfrak{A}_i)$, $q(i,j) = \delta^i_{a_1 \dots a_j}(q^i_0)$, $\boldsymbol{n}_j = \bigcup_i \{N^i_{q(i,j-1),a_j,q(i,j)}\}$; let $\boldsymbol{L}_1 = \boldsymbol{L}(\{B\}(\emptyset)^*\{X\}\boldsymbol{n}_1 \dots \boldsymbol{n}_k\{Y\})$, and let $\boldsymbol{L}'_1 = \boldsymbol{L}(\{B\}(\emptyset)^*\{X'\}\boldsymbol{n}_1 \dots \boldsymbol{n}_k\{Y\})$. Clearly, $\boldsymbol{L}_1$ and $\boldsymbol{L}'_1$ are $\mathsf{FO}(<)$-definable. If $\boldsymbol{L}_\Xi(\boldsymbol{q})$ is $\mathsf{FO}(<)$-definable, then so is $\boldsymbol{L}_2 = \boldsymbol{L}_1 \cap \boldsymbol{L}_\Xi(\boldsymbol{q})$. However, $\boldsymbol{L}_2 = \boldsymbol{L}(\{B\}(\emptyset\emptyset)^*\{X\}\boldsymbol{n}_1 \dots \boldsymbol{n}_k\{Y\})$ is not $\mathsf{FO}(<)$-definable. Similarly, $\boldsymbol{L}'_2 = \boldsymbol{L}'_1 \cap \boldsymbol{L}_\Xi(\boldsymbol{q}(x))$ is not $\mathsf{FO}(<)$-definable. So $\boldsymbol{L}_\Xi(\boldsymbol{q})$ and $\boldsymbol{L}_\Xi(\boldsymbol{q}(x))$ are not $\mathsf{FO}(<)$-definable. ❑

9. Conclusions

The problems we investigate in this article originate in the area of ontology-based access to temporal data. Classical atemporal ontology-based data access (OBDA), which over the past 15 years has become one of the most impressive applications of Description Logics and Semantic Technologies, is based on the idea of rewriting ontology-mediated queries (OMQs) into query languages supported by conventional database management systems (DBMSs). For relational data, standard target languages for rewritings are SQL—that is, essentially FO-formulas—and datalog, which allows recursive queries over data. The idea of rewriting has led to numerous and profound results that either uniformly classify OMQs according to their FO- and datalog-rewritability or establish the computational complexity of recognising FO- and datalog-rewritability of OMQs in expressive languages and design practical decision and rewriting algorithms. In classical database theory, FO- and linear-datalog-rewritability of datalog queries has been an active research area since the 1980s.

Unfortunately, those results and developed techniques are not applicable to OMQs over temporal data, where the timestamps are linearly ordered by the precedence relation $<$ and

OMQs may contain temporal constructs. First, as well known, the interaction between temporal and description logic operators tends to dramatically increase the complexity of OMQ answering, which makes the uniform classification of OMQs according to their rewritability type much harder. Some initial steps in this direction have been made by Borgwardt et al. (2019), Artale et al. (2022). Second, even without the description logic constructs, pure one-dimensional temporal OMQs give rise to the complexity classes and target languages for FO-rewritings that have not occurred in the OBDA context so far. For instance, any $\mathit{LTL}_{bool}^{\Box\bigcirc}$ OMQ is rewritable into $\mathsf{FO}(<, \mathsf{RPR})$—a class not appearing in the classical (atemporal) OBDA literature—that essentially requires recursion, which is weaker than linear datalog recursion but still not expressible in SQL.

In this article, our concern is determining the optimal rewritability type for OMQs given in linear temporal logic *LTL*. In fact, we argue in the introduction that such OMQs provide an adequate formalism for querying sensor log data from various parts of complex equipment where there is no relevant interaction between those parts, and the results of measurements are qualitatively graded as, e.g., high, medium, low, etc. Our starting point is establishing a close connection between rewritability of *LTL* OMQs and definability of regular languages by means of $\mathsf{FO}(<)$-formulas possibly containing extra predicates and constructs. The computational complexity and definability of regular languages have been investigated since the late 1980s. The relevant FO-languages identified are $\mathsf{FO}(<)$, $\mathsf{FO}(<, \equiv)$, $\mathsf{FO}(<, \mathsf{MOD})$ and $\mathsf{FO}(\mathsf{RPR})$, the first two of which are in AC^0 for data complexity, the third is in ACC^0 and the last one in NC^1. In practice, $\mathsf{FO}(<, \mathsf{MOD})$-rewritable OMQs can be implemented in SQL using the `count` operator, while $\mathsf{FO}(<, \equiv)$-rewritable ones do not need it. It is also known that recognising $\mathsf{FO}(<)$-definability of regular languages given by a DFA is PSpace-complete; recognising definability by $\mathsf{FO}(<, \equiv)$ and $\mathsf{FO}(<, \mathsf{MOD})$ formulas is known to be decidable, but the exact complexity has so far remained open.

The main technical results we obtain here are threefold. First, we settle the open problems just mentioned by proving that deciding $\mathsf{FO}(<)$-, $\mathsf{FO}(<, \equiv)$- and $\mathsf{FO}(<, \mathsf{MOD})$-definability of regular languages given by a DFA, NFA or 2NFA is PSpace-complete. Second, we show that deciding $\mathsf{FO}(<)$-, $\mathsf{FO}(<, \equiv)$- and $\mathsf{FO}(<, \mathsf{MOD})$-rewritability of *LTL* OMQs is ExpSpace-complete. And finally, we identify a number of natural and practically important OMQ classes for which these problems are PSpace-, Π_2^p- or coNP-complete; these results could lead to feasible algorithms to be used in temporal OBDA systems.

While this article makes steps towards the non-uniform approach to temporal OBDA, many interesting and challenging problems remain open. We discuss some of them below.

1. Our results on linear Horn, core and Krom OMQs are only established for ontologies with $\bigcirc_F$ and $\bigcirc_P$. Some of the techniques used in the proofs do not go through in the presence of $\Box_F$ and $\Box_P$, and so it would be interesting to see if the same complexity results hold for the fragments with all of these operators. One could also consider adding the operators 'since' and 'until' to ontologies and/or queries in *LTL* OMQs. General results, such as Theorem 16, will not be affected by this, but it is an open question for the fragments mentioned above. Finally, we could not establish the complexity of deciding $\mathsf{FO}(<, \mathsf{MOD})$-rewritability of linear $\mathit{LTL}_{horn}^{\bigcirc}$ OMPQs. It is likely to be PSpace, but we did not manage to prove an appropriate criterion in the spirit of Theorems 24 and 26.

2. In this article, we consider queries with at most one answer variable. More expressive query languages based on monadic first-order logic MFO(<) and allowing multiple answer variables have been suggested by Artale et al. (2021). It would be interesting to understand the impact of replacing *LTL* queries with MFO(<) queries in *LTL* OMQs on their FO-rewritability properties.

3. Another prominent temporal KR formalism that has great potential as an ontology and query language for temporal OBDA is metric temporal logic *MTL*, which was originally introduced for modelling and reasoning about real-time systems (Koymans, 1990; Alur & Henzinger, 1993). Each operator in *MTL* is indexed by a temporal interval over which the operator works: for example, $\Diamond_{(0,1.5]}A$ is true at t iff A holds at some t' with $0 < t' - t \leq 1.5$. The interpretation domain is dense $\mathbb{R}$ or $\mathbb{Q}$ under the continuous semantics and the active domain of the data instance under the pointwise semantics (Ouaknine & Worrell, 2008). *MTL* is more expressive and succinct than *LTL* and is also suitable in scenarios where sensors report their measurements asynchronously. In the context of OBDA, *MTL* has recently been investigated by Brandt et al. (2018), Ryzhikov et al. (2019), Walega et al. (2020b), Tena Cucala et al. (2021), Wang et al. (2022). Target rewriting languages for *MTL* OMQs include FO(DTC), FO(TC) with (deterministic) transitive closure, and datalog(FO), which correspond to the complexity classes L, NL and P, respectively. At present, the problem of recognising the data complexity and optimal rewritability type of *MTL* OMQs is wide open.

4. In OBDA practice, we are concerned not only with the fact of FO-rewritability of a given OMQ but also with the size and shape of the rewriting to be executed by a DBMS (e.g., Bienvenu et al., 2018, 2017). The experiments with a few real-world use cases reported by Brandt et al. (2018, 2019) indicate that temporal OMQs with a non-recursive ontology are scalable and efficient. But we are not aware of any theoretical results on the succinctness of FO-rewritings for temporal OMQs.

5. Extending the results obtained above for 1D *LTL* OMQs to various 2D combinations of *LTL* with description logics (such as *DL-Lite*, $\mathcal{EL}$ or $\mathcal{ALC}$), Schema.org or datalog could be especially challenging due to the interaction between the temporal and domain dimensions. In the Horn case, one might try to use a variant of the automata-theoretic approach developed by Lutz and Sabellek (2017, 2019).

6. Finally, from the application point of view, it is important to identify real-world use-cases for temporal OBDA, relevant classes of OMQ, and then develop OMQ rewriting and optimisation algorithms for those classes. Some work in this direction has recently been done for both *MTL* and *LTL* (Wang et al., 2022; Brandt et al., 2018; Tahrat et al., 2020). Although the results of this paper suggest algorithms that can identify the best rewritability class (and so the most efficient database query language) for a given OMQ, implementing and optimising such algorithms is a serious challenge. Furthermore, the algorithms mentioned above need to be incorporated into a user-friendly OBDA system such as Ontop (Rodriguez-Muro et al., 2013; Xiao et al., 2020).

Acknowledgments

This work was supported by the EPSRC U.K. grant EP/S032282 for the project '$quant^{MD}$: Ontology-Based Management for Many-Dimensional Quantitative Data'. We are grateful to the referees of this article for their careful reading, valuable comments and suggestions.

Appendix A.

A.1 Proof of Theorem 5 (*ii*)

Theorem 5 (*ii*)**.** *Let $\boldsymbol{q} = (\mathcal{O}, \varkappa)$ be a Boolean and $\boldsymbol{q}(x) = (\mathcal{O}, \varkappa(x))$ a specific OMQ. Then, for any $\mathcal{L} \in \{\mathsf{FO}(<), \mathsf{FO}(<,\equiv), \mathsf{FO}(<,\mathsf{MOD})\}$ and $\Xi \subseteq \mathsf{sig}(\boldsymbol{q})$, the OMQ $\boldsymbol{q}$ is $\mathcal{L}$-rewritable over Ξ-ABoxes iff $\boldsymbol{L}_\Xi(\boldsymbol{q})$ is $\mathcal{L}$-definable; similarly, $\boldsymbol{q}(x)$ is $\mathcal{L}$-rewritable over Ξ-ABoxes iff $\boldsymbol{L}_\Xi(\boldsymbol{q}(x))$ is $\mathcal{L}$-definable.*

Proof. For any $A \in \Xi$ and any $a \in \Sigma_\Xi$, we set

$$\chi_A(y) = \bigvee_{A \in a \in \Sigma_\Xi} a(y), \qquad \chi_a(y) = \bigwedge_{A \in a} A(y) \wedge \bigwedge_{A \notin a} \neg A(y),$$

where $a(y)$ is a unary predicate associated with each $a \in \Sigma_\Xi$. For any Ξ-ABox $\mathcal{A}$ and any $n \in \mathsf{tem}(\mathcal{A})$, we have $\mathfrak{S}_\mathcal{A} \models A(n)$ iff $\mathfrak{S}_{w_\mathcal{A}} \models \chi_A(n)$, and $\mathfrak{S}_{w_\mathcal{A}} \models a(n)$ iff $\mathfrak{S}_\mathcal{A} \models \chi_a(n)$. Thus, we obtain an $\mathcal{L}$-sentence defining $\boldsymbol{L}_\Xi(\boldsymbol{q})$ by taking an $\mathcal{L}$-rewriting of $\boldsymbol{q}$ and replacing all atoms $A(y)$ in it with $\chi_A(y)$. Conversely, we obtain an $\mathcal{L}$-rewriting of $\boldsymbol{q}$ by taking an $\mathcal{L}$-sentence defining $\boldsymbol{L}_\Xi(\boldsymbol{q})$ and replacing all $a(y)$ in it with $\chi_a(y)$.

Consider next $\boldsymbol{q}(x)$. Let $\varphi(x)$ be an $\mathcal{L}$-rewriting of $\boldsymbol{q}(x)$ and let $\varphi'(x)$ be the result of replacing atoms $A(y)$ in $\varphi(x)$ with $\chi'_A(y) = \bigvee_{A \in a \in \Gamma_\Xi} a(y)$. Given an ABox $\mathcal{A}$ and $i \in \mathsf{tem}(\mathcal{A})$, we have $\mathfrak{S}_\mathcal{A} \models \varphi(i)$ iff $\mathfrak{S}_{w_{\mathcal{A},i}} \models \varphi'(i)$. A word $w = a_0 \dots a_n \in \Gamma_\Xi^*$ is in $\boldsymbol{L}_\Xi(\boldsymbol{q}(x))$ iff (a) there is i such that $a_i \in \Sigma'_\Xi$, (b) $a_j \in \Sigma_\Xi$ for all $j \neq i$, and (c) $\mathfrak{S}_w \models \varphi'(i)$. Therefore, for the sentence

$$\varphi'' = \exists x \left(\varphi'(x) \wedge \forall y \left[\left((y = x) \to \bigvee_{a' \in \Sigma'_\Xi} a'(y)\right) \wedge \left((y \neq x) \to \bigvee_{a \in \Sigma_\Xi} a(y)\right)\right]\right)$$

and a word $w \in \Gamma_\Xi^*$, we have $\mathfrak{S}_w \models \varphi''$ iff $w = w_{\mathcal{A},i}$ for some $\mathcal{A}$ and i such that $\mathfrak{S}_\mathcal{A} \models \varphi(i)$. It follows that φ'' defines $\boldsymbol{L}_\Xi(\boldsymbol{q}(x))$.

Now, let ψ be an $\mathcal{L}$-sentence defining $\boldsymbol{L}_\Xi(\boldsymbol{q}(x))$ and let $\psi'(x)$ be the result of replacing atoms $a(y)$ in φ, for $a \in \Sigma_\Xi$, with $a(y) \wedge (x \neq y)$ and atoms $a'(y)$, for $a' \in \Sigma'_\Xi$, with $a(y) \wedge (x = y)$. For $w = a_0 \dots a_n \in \Sigma_\Xi^*$, we have $\mathfrak{S}_w \models \psi'(i)$ iff $\mathfrak{S}_{w_i} \models \psi$, where w_i is w with a_i replaced by a'_i. Let $\psi''(x)$ be the result of replacing $a(y)$ in $\psi'(x)$ with $\chi_a(y)$. Then, for any $\mathcal{A}$ and $i \in \mathsf{tem}(\mathcal{A})$, we have $\mathfrak{S}_\mathcal{A} \models \psi''(i)$ iff $\mathfrak{S}_{w_\mathcal{A}} \models \psi'(i)$ iff $\mathfrak{S}_{w_{\mathcal{A},i}} \models \psi$, and so $\psi''(x)$ is a rewriting of $\boldsymbol{q}$. ❑

A.2 Proof of Lemma 7

Lemma 7. *Suppose $\mathcal{L} \in \{\mathsf{FO}(<), \mathsf{FO}(<,\equiv), \mathsf{FO}(<,\mathsf{MOD})\}$ and Σ, Γ and Δ are alphabets such that $\Sigma \cup \{x, y\} \subseteq \Gamma \subseteq \Delta$, for some $x, y \notin \Sigma$. Then a regular language $\boldsymbol{L}$ over Σ is $\mathcal{L}$-definable iff the regular language*

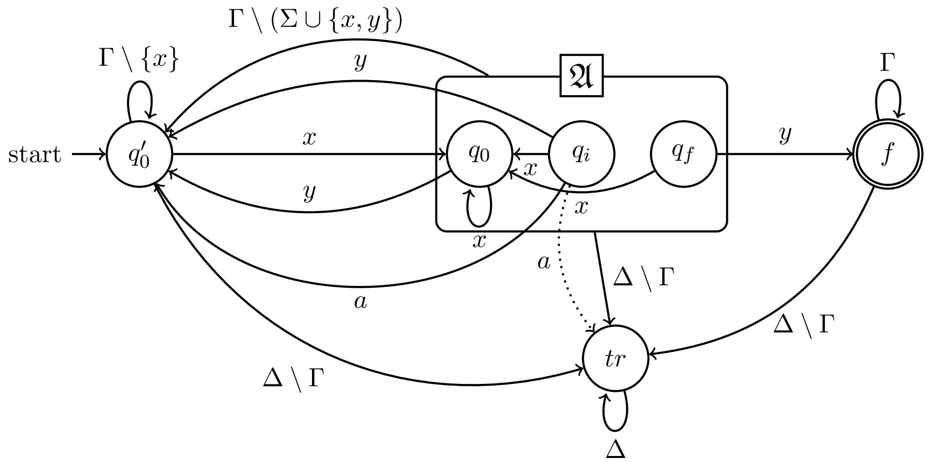
$$\boldsymbol{L}' = \{w_1 x w y w_2 \mid w \in \boldsymbol{L},\ w_1, w_2 \in \Gamma^*\}$$

is $\mathcal{L}$-definable over Δ.

Proof. Let $\boldsymbol{L} = \boldsymbol{L}(\mathfrak{A})$, for a minimal DFA $\mathfrak{A} = (Q, \Sigma, \delta, q_0, F)$. Let tr be the trash state[4] in $\mathfrak{A}$ if any. Given alphabets Γ, Δ, consider the DFA

$$\mathfrak{A}' = \big(Q \cup \{tr, q_0', f\}, \Delta, \delta', q_0', \{f\}\big),$$

where δ' consists of the following transitions: (q, a, p) for $(q, a, p) \in \delta$, $p \neq tr$, (q, a, q_0') for $(q, a, tr) \in \delta$, (q_0', a, q_0') for $a \in \Gamma \setminus \{x\}$, (q, x, q_0) for $q \in Q \cup \{q_0'\}$, (q, a, q_0') for $q \in Q$ and $a \in \Gamma \setminus (\Sigma \cup \{x, y\})$, (q, y, q_0') for $q \in Q \setminus F$, (q, y, f) for $q \in F$, (f, a, f) for $a \in \Gamma$, (q, a, tr) for q and $a \in \Delta \setminus \Gamma$, (tr, a, tr) for $a \in \Delta$. The DFA $\mathfrak{A}'$ is illustrated in the picture below, where a transition labelled by a set stands for the corresponding transitions for each element of that set, the transitions starting from the frame around $\mathfrak{A}$ represent the corresponding transitions from every state in $\mathfrak{A}$, and the transitions from states in $\mathfrak{A}$ to tr (shown as the dashed arrow in the picture) are redirected to q_0'. It is readily checked that $\boldsymbol{L}(\mathfrak{A}') = \{w_1 x w y w_2 \mid w \in \boldsymbol{L},\ w_1, w_2 \in \Gamma^*\}$.



We now show that $\boldsymbol{L}(\mathfrak{A})$ is $\mathcal{L}$-definable iff the language $\boldsymbol{L}(\mathfrak{A}')$ is $\mathcal{L}$-definable. As the argument is effectively the same for all $\mathcal{L}$, we only show it in one case.

($\Leftarrow$) If $\boldsymbol{L}(\mathfrak{A})$ is not $\mathsf{FO}(<)$-definable, then, by Theorem 6 (i), there exist a state q, a number k, and a word $u \in \Sigma^*$ such that $q \not\sim \delta_u(q)$ and $q = \delta_{u^k}(q)$. One can readily check that the same q, k and u satisfy the same condition in $\mathfrak{A}'$, and so $\boldsymbol{L}(\mathfrak{A}')$.

($\Rightarrow$) If $\boldsymbol{L}(\mathfrak{A}')$ is not $\mathsf{FO}(<)$-definable, then, by Theorem 6 (i), there exist a state q, a number k, and a word $u \in \Delta^*$ such that $q \not\sim \delta_u(q)$ and $q = \delta_{u^k}(q)$. There are no transitions leaving tr and the only transition leaving f is to tr. It follows that, when reading u^k starting from q, $\mathfrak{A}'$ can visit f or tr. Suppose it visits q_0'. As the only way of leaving q_0' not to tr is via x, the word u contains x. Let $u = u_1 x u_2$. But then, for any $p \notin \{f, tr\}$, we have $\delta_u(p) = \delta_{u_2}(q_0)$, and so all $\delta_{u^i}(q)$ are the same, which is a contradiction. Thus, $\mathfrak{A}'$ does not visit q_0'. It follows that $\delta_{u^i}(q) \in Q$ and $u \in \Sigma^*$. Then the same q, k, and u satisfy the conditions of Theorem 6 (i) for $\mathfrak{A}$, and so $\boldsymbol{L}(\mathfrak{A})$ is not $\mathsf{FO}(<)$-definable. ❑

4. A *trash state* is a state from which no accepting state is reachable. A minimal DFA can have at most one trash state.

A.3 Additional Axioms and Counters for the Proof of Theorem 16

Below are the axioms describing the transitions of the automata $\mathfrak{A}_i$. For $\mathfrak{A}_0$, we use the axioms

$$
\begin{aligned}
&[\mathbb{A}=0] \wedge T \wedge \sharp \to [(\bigcirc_F \mathbb{A})=0] \wedge \bigcirc_F Q \wedge [(\bigcirc_F \mathbb{L})=0],\\
&[\mathbb{A}=0] \wedge Q \wedge [\mathbb{L}=0] \wedge (q_1, x_1) \to [(\bigcirc_F \mathbb{A})=0] \wedge \bigcirc_F Q \wedge [(\bigcirc_F \mathbb{L})=1],\\
&\dots\\
&[\mathbb{A}=0] \wedge Q \wedge [\mathbb{L}=n-1] \wedge x_n \to [(\bigcirc_F \mathbb{A})=0] \wedge \bigcirc_F Q \wedge [(\bigcirc_F \mathbb{L})=n],\\
&[\mathbb{A}=0] \wedge Q \wedge [\mathbb{L}>n-1] \wedge [\mathbb{L}< N] \wedge \mathsf{b} \to [(\bigcirc_F \mathbb{A})=0] \wedge \bigcirc_F Q \wedge [(\bigcirc_F \mathbb{L})=\mathbb{L}+1],\\
&[\mathbb{A}=0] \wedge Q \wedge [\mathbb{L}=N] \wedge \sharp \to [(\bigcirc_F \mathbb{A})=0] \wedge \bigcirc_F P \wedge [(\bigcirc_F \mathbb{L})=0],\\
&[\mathbb{A}=0] \wedge P \wedge [\mathbb{L}=0] \wedge a \to [(\bigcirc_F \mathbb{A})=0] \wedge \bigcirc_F P_{\sharp\sharp}, \quad \text{for } a \neq (q_{acc}, \mathsf{b}), \sharp, \flat,\\
&[\mathbb{A}=0] \wedge P \wedge [\mathbb{L}=0] \wedge (q_{acc}, \mathsf{b}) \to [(\bigcirc_F \mathbb{A})=0] \wedge \bigcirc_F P \wedge [(\bigcirc_F \mathbb{L})=1],\\
&[\mathbb{A}=0] \wedge P_{\sharp\sharp} \wedge a \to [(\bigcirc_F \mathbb{A})=0] \wedge \bigcirc_F P_{\sharp\sharp}, \quad \text{for } a \neq \sharp,\\
&[\mathbb{A}=0] \wedge P_{\sharp\sharp} \wedge \sharp \to [(\bigcirc_F \mathbb{A})=0] \wedge \bigcirc_F P \wedge [(\bigcirc_F \mathbb{L})=0],\\
&[\mathbb{A}=0] \wedge P \wedge [\mathbb{L}>0] \wedge [\mathbb{L}<N] \wedge \mathsf{b} \to [(\bigcirc_F \mathbb{A})=0] \wedge \bigcirc_F P \wedge [(\bigcirc_F \mathbb{L})=\mathbb{L}+1)],\\
&[\mathbb{A}=0] \wedge P \wedge [\mathbb{L}=N] \wedge \flat \to [\mathbb{A}=0] \wedge \bigcirc_F F.
\end{aligned}
$$

For $\mathfrak{A}_i$ with $0 < i \leq N$ and $a, b, c \in \Sigma' \setminus \{\sharp, \flat\}$, we need the axioms

$$
\begin{aligned}
&[\mathbb{A}=1] \wedge [\mathbb{A}<N+1] \wedge T \wedge \sharp \to [(\bigcirc_F \mathbb{A})=\mathbb{A}] \wedge \bigcirc_F R_\sharp,\\
&[\mathbb{A}>1] \wedge [\mathbb{A}<N+1] \wedge T \wedge \sharp \to [(\bigcirc_F \mathbb{A})=\mathbb{A}] \wedge \bigcirc_F Q \wedge [(\bigcirc_F \mathbb{L})=\mathbb{A}-1],\\
&[\mathbb{A}>1] \wedge [\mathbb{A}<N+1] \wedge Q \wedge [\mathbb{L}>1] \wedge a \to [(\bigcirc_F \mathbb{A})=\mathbb{A}] \wedge \bigcirc_F Q \wedge [(\bigcirc_F \mathbb{L})=\mathbb{L}-1],\\
&[\mathbb{A}>0] \wedge [\mathbb{A}<N+1] \wedge Q \wedge [\mathbb{L}=1] \wedge a \to [(\bigcirc_F \mathbb{A})=\mathbb{A}] \wedge \bigcirc_F R_a,\\
&[\mathbb{A}>0] \wedge [\mathbb{A}<N+1] \wedge R_a \wedge b \to [(\bigcirc_F \mathbb{A})=\mathbb{A}] \wedge \bigcirc_F R_{ab},\\
&[\mathbb{A}>0] \wedge [\mathbb{A}<N] \wedge R_{ab} \wedge c \to [(\bigcirc_F \mathbb{A})=\mathbb{A}] \wedge \bigcirc_F Q_{\gamma(a,b,c)} \wedge \bigcirc_F [\mathbb{L}=\mathbb{A}+1],\\
&[\mathbb{A}=N] \wedge R_{ab} \wedge \sharp \to [(\bigcirc_F \mathbb{A})=\mathbb{A}] \wedge \bigcirc_F P_{\gamma(a,b,\sharp)} \wedge \bigcirc_F [\mathbb{L}=N-1],\\
&[\mathbb{A}>0] \wedge [\mathbb{A}<N+1] \wedge Q_a \wedge [\mathbb{L}<N] \wedge b \to [(\bigcirc_F \mathbb{A})=\mathbb{A}] \wedge \bigcirc_F Q_a \wedge [(\bigcirc_F \mathbb{L})=\mathbb{L}+1],\\
&[\mathbb{A}=1] \wedge Q_a \wedge [\mathbb{L}=N] \wedge \sharp \to [(\bigcirc_F \mathbb{A})=\mathbb{A}] \wedge \bigcirc_F P_{\sharp a}\\
&[\mathbb{A}>1] \wedge [\mathbb{A}<N+1] \wedge Q_a \wedge [\mathbb{L}=N] \wedge \sharp \to [(\bigcirc_F \mathbb{A})=\mathbb{A}] \wedge \bigcirc_F P_a \wedge [(\bigcirc_F \mathbb{L})=\mathbb{A}-1]\\
&[\mathbb{A}>1] \wedge [\mathbb{A}<N+1] \wedge P_a \wedge [\mathbb{L}>1] \wedge b \to [(\bigcirc_F \mathbb{A})=\mathbb{A}] \wedge \bigcirc_F P_a \wedge [(\bigcirc_F \mathbb{L})=\mathbb{L}-1],\\
&[\mathbb{A}>0] \wedge [\mathbb{A}<N+1] \wedge P_a \wedge [\mathbb{L}=1] \wedge b \to [(\bigcirc_F \mathbb{A})=\mathbb{A}] \wedge \bigcirc_F P_{ba}\\
&[\mathbb{A}>0] \wedge [\mathbb{A}<N+1] \wedge P_{ab} \wedge a \to [(\bigcirc_F \mathbb{A})=\mathbb{A}] \wedge \bigcirc_F R_{ab},\\
&[\mathbb{A}>0] \wedge [\mathbb{A}<N] \wedge Q_b \wedge [\mathbb{L}=N] \wedge \flat \to [(\bigcirc_F \mathbb{A})=\mathbb{A}] \wedge \bigcirc_F F,\\
&[\mathbb{A}=N] \wedge R_{ab} \wedge \flat \to [(\bigcirc_F \mathbb{A})=\mathbb{A}] \wedge \bigcirc_F F.
\end{aligned}
$$

To calculate the value of j in the construction of $\mathcal{O}_{\mathsf{MOD}}$, we use the following counters, formulas, and axioms.

For two counters $\mathbb{X}$ and $\mathbb{Y}$, set

$$[\mathbb{X}=\mathbb{Y}/2] \;=\; X^0_k \wedge \bigwedge_{l=2}^{k} \left((Y^0_l \to X^0_{l-1}) \wedge (Y^1_l \to X^1_{l-1})\right).$$

We have $\mathcal{I}, n \models [\mathbb{X} = \mathbb{Y}/2]$ iff the values x of $\mathbb{X}$ and y of $\mathbb{Y}$ at n in $\mathcal{I}$ satisfy $x = \lfloor y/2 \rfloor$. We define three new counters $\mathbb{C}^{=}_{\mathbb{XY}}$, $\mathbb{C}^{-}_{\mathbb{XY}}$, and $\mathbb{C}^{+}_{\mathbb{XY}}$, which come with the following axioms, for all $\iota_1, \iota_2, \iota_3 \in \{0, 1\}$, that should be added to the ontology:

$$
\begin{aligned}
&X_i^{\iota_1} \wedge Y_i^{\iota_2} \to (C^{=}_{\mathbb{XY}})_i^{(\iota_1+\iota_2+1) \bmod 2}, && \text{for all } i \in [1, k],\\
&X_1^{\iota_1} \wedge Y_1^{\iota_2} \to (C^{+}_{\mathbb{XY}})_1^{0}, &&\\
&X_{i-1}^{\iota_1} \wedge Y_{i-1}^{\iota_2} \wedge (C^{+}_{\mathbb{XY}})_{i-1}^{\iota_3} \to (C^{+}_{\mathbb{XY}})_i^{(\iota_1\iota_2+\iota_1\iota_3+\iota_2\iota_3) \bmod 2}, && \text{for all } i \in [2, k],\\
&X_1^{\iota_1} \wedge Y_1^{\iota_2} \to (C^{-}_{\mathbb{XY}})_1^{0}, &&\\
&X_{i-1}^{\iota_1} \wedge Y_{i-1}^{\iota_2} \wedge (C^{-}_{\mathbb{XY}})_{i-1}^{\iota_3} \to (C^{-}_{\mathbb{XY}})_i^{(\iota_1\iota_2+\iota_1\iota_3+\iota_2\iota_3+\iota_2+\iota_3) \bmod 2}, && \text{for all } i \in [2, k].
\end{aligned}
$$

Define the following formulas, where $\mathbb{W}, \mathbb{X}, \mathbb{Y}$ are some counters:

$$
\begin{aligned}
[\mathbb{X} > \mathbb{Y}] &= \bigvee_{i=1}^{k} \big(X_i^1 \wedge Y_i^0 \wedge \bigwedge_{j=i+1}^{k} (C^{=}_{\mathbb{XY}})_i^1\big),\\
[\mathbb{X} \geq \mathbb{Y}] &= [\mathbb{X} > \mathbb{Y}] \vee \bigwedge_{i=1}^{k} (C^{=}_{\mathbb{XY}})_i^1,\\
[\mathbb{W} = \mathbb{X} + \mathbb{Y}] &= \bigwedge_{i=1}^{k} \bigwedge_{\iota_{1,2,3} \in \{0,1\}} \big(X_i^{\iota_1} \wedge Y_i^{\iota_2} \wedge (C^{+}_{\mathbb{XY}})_i^{\iota_3} \to W_i^{\iota_1+\iota_2+\iota_3 \bmod 2}\big),\\
[\mathbb{W} = \mathbb{X} - \mathbb{Y}] &= \bigwedge_{i=1}^{k} \bigwedge_{\iota_{1,2,3} \in \{0,1\}} \big(X_i^{\iota_1} \wedge Y_i^{\iota_2} \wedge (C^{-}_{\mathbb{XY}})_i^{\iota_3} \to W_i^{\iota_1+\iota_2+\iota_3 \bmod 2}\big).
\end{aligned}
$$

We have $\mathcal{I}, n \models [\mathbb{X} > \mathbb{Y}]$, $\mathcal{I}, n \models [\mathbb{X} \geq \mathbb{Y}]$, $\mathcal{I}, n \models [\mathbb{W} = \mathbb{X} + \mathbb{Y}]$, or $\mathcal{I}, n \models [\mathbb{W} = \mathbb{X} - \mathbb{Y}]$ iff the values x of $\mathbb{X}$, y of $\mathbb{Y}$, and w of $\mathbb{W}$ at n in $\mathcal{I}$ satisfy, respectively, the following conditions: $x > y$, $x \geq y$, $w = x + y$ for $x + y < 2^k$, and $w = x - y$ for $x \geq y$.

In our ontology $\mathcal{O}_{\mathsf{MOD}}$, we use counters $\mathbb{U}_l$, $\mathbb{V}_l$, $\mathbb{R}_l$, $\mathbb{R}_l^{+}$, $\mathbb{R}_l^{-}$, $\mathbb{S}_l$, $\mathbb{S}_l^{-}$, $\mathbb{S}_l^{+}$, $\mathbb{D}_l$, $\mathbb{G}_l$, $\mathbb{H}_l$, for $l \in [0, \dots, 2k]$, along with some auxiliary counters $\mathbb{C}_{\mathbb{XY}}$. Intuitively, the counters with the index l hold the values of the corresponding expressions after the l-th step of the algorithm according to the table below:

$\mathbb{U}_l, \mathbb{V}_l, \mathbb{R}_l, \mathbb{S}_l$	u, v, r, s
$\mathbb{R}_l^{+}, \mathbb{S}_l^{+}$	$r + p, s + p$
$\mathbb{R}_l^{-}, \mathbb{S}_l^{-}$	$-r \bmod p, -s \bmod p$
$\mathbb{D}_l$	$\lvert u - v \rvert$
$\mathbb{G}_l$	the even number from the pair $((r - s) \bmod p)$, $((r - s) \bmod p) + p$
$\mathbb{H}_l$	the even number from the pair $((s - r) \bmod p)$, $((s - r) \bmod p) + p$

We add the following axioms (simulating the algorithm) to the ontology $\mathcal{O}_{\mathsf{MOD}}$:

$$
\begin{aligned}
&[\mathbb{A} > 0] \wedge [\mathbb{A} < p] \wedge S \wedge \natural \to [\mathbb{U}_0 = p] \wedge [\mathbb{V}_0 = \mathbb{A}] \wedge [\mathbb{R}_0 = 0] \wedge [\mathbb{S}_0 = 1],\\
&[\mathbb{U}_l > \mathbb{V}_l] \to [\mathbb{D}_l = \mathbb{U}_l - \mathbb{V}_l],\\
&[\mathbb{V}_l \geq \mathbb{U}_l] \to [\mathbb{D}_l = \mathbb{V}_l - \mathbb{U}_l],\\
&[\mathbb{R}_l^{+} = \mathbb{R}_l + \mathbb{U}_0] \wedge [\mathbb{R}_l^{-} = \mathbb{U}_0 - \mathbb{R}_l] \wedge [\mathbb{S}_l^{+} = \mathbb{S}_l + \mathbb{U}_0] \wedge [\mathbb{S}_l^{-} = \mathbb{U}_0 - \mathbb{S}_l],
\end{aligned}
$$

$$[\mathbb{R}_l \ge \mathbb{S}_l] \wedge (((R_l)^0_1 \wedge (S_l)^0_1) \vee ((R_l)^1_1 \wedge (S_l)^1_1)) \to [\mathbb{G}_l = \mathbb{R}_l - \mathbb{S}_l] \wedge [\mathbb{H}_l = \mathbb{S}^+_l + \mathbb{R}^-_l],$$
$$[\mathbb{R}_l \ge \mathbb{S}_l] \wedge (((R_l)^1_1 \wedge (S_l)^0_1) \vee ((R_l)^0_1 \wedge (S_l)^1_1)) \to [\mathbb{G}_l = \mathbb{R}_l + \mathbb{S}^-_l] \wedge [\mathbb{H}_l = \mathbb{S}^+_l - \mathbb{R}_l],$$
$$[\mathbb{S}_l > \mathbb{R}_l] \wedge (((R_l)^0_1 \wedge (S_l)^0_1) \vee ((R_l)^1_1 \wedge (S_l)^1_1)) \to [\mathbb{G}_l = \mathbb{R}^+_l + \mathbb{S}^-_l] \wedge [\mathbb{H}_l = \mathbb{S}_l - \mathbb{R}_l],$$
$$[\mathbb{S}_l > \mathbb{R}_l] \wedge (((R_l)^1_1 \wedge (S_l)^0_1) \vee ((R_l)^0_1 \wedge (S_l)^1_1)) \to [\mathbb{G}_l = \mathbb{R}^+_l - \mathbb{S}_l] \wedge [\mathbb{H}_l = \mathbb{S}_l + \mathbb{R}^-_l],$$
$$[\mathbb{V}_l > 0] \wedge (V_l)^0_1 \wedge (S_l)^0_1 \to [\mathbb{U}_{l+1} = \mathbb{U}_l] \wedge [\mathbb{V}_{l+1} = \mathbb{V}_l/2] \wedge [\mathbb{R}_{l+1} = \mathbb{R}_l] \wedge [\mathbb{S}_{l+1} = \mathbb{S}_l/2],$$
$$[\mathbb{V}_l > 0] \wedge (V_l)^0_1 \wedge (S_l)^1_1 \to [\mathbb{U}_{l+1} = \mathbb{U}_l] \wedge [\mathbb{V}_{l+1} = \mathbb{V}_l/2] \wedge [\mathbb{R}_{l+1} = \mathbb{R}_l] \wedge [\mathbb{S}_{l+1} = \mathbb{S}^+_l/2],$$
$$(V_l)^1_1 \wedge (U_l)^0_1 \wedge (R_l)^0_1 \to [\mathbb{U}_{l+1} = \mathbb{U}_l/2] \wedge [\mathbb{V}_{l+1} = \mathbb{V}_l] \wedge [\mathbb{R}_{l+1} = \mathbb{R}_l/2] \wedge [\mathbb{S}_{l+1} = \mathbb{S}_l],$$
$$(V_l)^1_1 \wedge (U_l)^0_1 \wedge (R_l)^1_1 \to [\mathbb{U}_{l+1} = \mathbb{U}_l/2] \wedge [\mathbb{V}_{l+1} = \mathbb{V}_l] \wedge [\mathbb{R}_{l+1} = \mathbb{R}^+_l/2] \wedge [\mathbb{S}_{l+1} = \mathbb{S}_l],$$
$$(V_l)^1_1 \wedge (U_l)^1_1 \wedge [\mathbb{U}_l > \mathbb{V}_l] \to [\mathbb{U}_{l+1} = \mathbb{D}_l/2] \wedge [\mathbb{V}_{l+1} = \mathbb{V}_l] \wedge [\mathbb{R}_{l+1} = \mathbb{H}_l/2] \wedge [\mathbb{S}_{l+1} = \mathbb{S}_l],$$
$$(V_l)^1_1 \wedge (U_l)^1_1 \wedge [\mathbb{V}_l \ge \mathbb{U}_l] \to [\mathbb{U}_{l+1} = \mathbb{U}_l] \wedge [\mathbb{V}_{l+1} = \mathbb{D}_l/2] \wedge [\mathbb{R}_{l+1} = \mathbb{R}_l] \wedge [\mathbb{S}_{l+1} = \mathbb{G}_j/2],$$
$$[\mathbb{V}_l = 0] \to [\mathbb{J} = \mathbb{R}^-_l].$$

A.4 Proof of Lemma 23

Lemma 23. *Let $\mathcal{A} \in \Sigma^*_\Xi$ be of the form $\emptyset^N \mathcal{B} \emptyset^N$. Then $A \in \boldsymbol{\tau}^{\mathsf{sig}(\mathcal{O})}_{\mathcal{O},\mathcal{A}}(\ell)$ iff there exists a run $(q_0, 0), \dots, (q, \ell), (q_A, i)$ of $\mathfrak{A}^\Xi_\mathcal{O}$ on $\mathcal{A}$, for all ℓ with $N \le \ell < |\mathcal{A}| - N$.*

Proof. We call a sequence $\mathfrak{D}$ of the form

$$(C^0_1 \wedge \dots \wedge C^0_{k_0} \to A_1, n_1), (C^1_1 \wedge \dots \wedge C^1_{k_1} \wedge \bigcirc^{i_1} A_1 \to A_2, n_2), \dots,$$
$$(C^m_1 \wedge \dots \wedge C^m_{k_m} \wedge \bigcirc^{i_m} A_m \to A, n_{m+1}) \quad (39)$$

a *derivation* of A from $\mathcal{O}$ and $\mathcal{A}$ if the axioms are from $\mathcal{O}$ and the numbers $n_1, \dots, n_m, n_{m+1}$ are such that $n_{j+1} = n_j + i_j$ and $\mathcal{A} \models C^j_1 \wedge \dots \wedge C^j_{k_j}(n_{j+1})$. We say that such a derivation ends at n if $n_{m+1} = n$. It is straightforward to verify that $A \in \boldsymbol{\tau}^{\mathsf{sig}(\mathcal{O})}_{\mathcal{O},\mathcal{A}}(\ell)$ iff there is a derivation of A at ℓ, for any $\ell \in \mathbb{Z}$.

Let $\mathcal{A}$ be of the form $\emptyset^N \mathcal{B} \emptyset^N$. We now show that, for any ℓ with $N \le \ell < |\mathcal{A}| - N$,

$$\text{if there is a derivation of } A \text{ at } \ell, \text{ then there is a derivation of } A \text{ at } \ell$$
$$\text{such that } 0 \le n_j < |\mathcal{A}| \text{ for all } n_j \text{ in it.} \quad (40)$$

Proposition 36. *Let $\mathfrak{D}_1$, $\mathfrak{D}_2$, $\mathfrak{D}_3$ be derivations from $\mathcal{O}$ and $\mathcal{A}$ of the form:*

$$\mathfrak{D}_1 = \dots, (C_1 \wedge \dots \wedge C_k \wedge \bigcirc^i A \to A_0, n_0),$$
$$\mathfrak{D}_2 = (\bigcirc^{i_0} A_0 \to A_1, n_1), \dots, (\bigcirc^{i_{m-1}} A_{m-1} \to A_m, n_m),$$
$$\mathfrak{D}_3 = (C'_1 \wedge \dots \wedge C'_{k'} \wedge \bigcirc^i A_m \to A_{m+1}, n_{m+1}), \dots$$

If $\mathfrak{D}_1\mathfrak{D}_2\mathfrak{D}_3$ is a derivation of A at ℓ, then there is a derivation $\mathfrak{D}_1\mathfrak{D}'_2\mathfrak{D}_3$ of A at ℓ from $\mathcal{O}$ and $\mathcal{A}$ such that $\min\{n_0, n_{m+1}\} - 2M^2 \le n_j \le \max\{n_0, n_{m+1}\} + 2M^2$ *for all n_j in $\mathfrak{D}'_2$.*

Proof. Suppose $n_{m+1} > n_0$ (the opposite case is analogous). Let j be the earliest number in $\mathfrak{D}_2$ such that

– either $n_j = n_{m+1}$ and $n_{j+k} = n_{m+1}$ for some $k \ge 0$,

– or $n_j = n_0$ and $n_{j+k} = n_0$ for some $k \geq 0$.

If there is no such j, then Proposition 36 holds with $\mathfrak{D}'_2 = \mathfrak{D}_2$. Suppose the former case holds for the earliest j. Let $\mathfrak{D}_2 = \mathfrak{D}_4\mathfrak{D}_5\mathfrak{D}_6$, where $\mathfrak{D}_5$ is the subsequence of $\mathfrak{D}_2$ between j (not inclusive) and $j+k$. Consider any quadruple $((A_{j'}, n_{j'}), (A_{j''}, n_{j''}), (A_{k''}, n_{k''}), (A_{k'}, n_{k'}))$ in $\mathfrak{D}_5$ with $j' \leq j'' \leq k'' \leq k'$, $n_{j'} = n_{k'}$, $n_{j''} = n_{k''}$, $A_{j'} = A_{j''}$ and $A_{k'} = A_{k''}$. Clearly, $\mathfrak{D}_1(\mathfrak{D}_4\mathfrak{D}'_5\mathfrak{D}_6)\mathfrak{D}_3$ is also a derivation A at ℓ from $\mathcal{O}$ and $\mathcal{A}$, where

$$\begin{aligned}\mathfrak{D}'_5 = (\bigcirc^{i_j} A_j \to A_{j+1}, n_{j+1}), \dots, (\bigcirc^{i_{j'-1}} A_{j'-1} \to A_{j'}, n_{j'}),&\\ (\bigcirc^{i_{j''}} A_{j''} \to A_{j''+1}, n_{j''+1} - d), \dots&\\ (\bigcirc^{i_{k''-1}} A_{k''-1} \to A_{k''}, n_{k''} - d), (\bigcirc^{i_{k'}} A_{k'} \to A_{k'+1}, n_{k'+1}), \dots,&\\ (\bigcirc^{i_{j+k-1}} A_{j+k-1} \to A_{j+k}, n_{j+k})&\end{aligned}$$

and $d = n_{j''} - n_{j'}$. After recursively applying to $\mathfrak{D}_5$ the transformation above for each quadruple $((A_{j'}, n_{j'}), (A_{j''}, n_{j''}), (A_{k''}, n_{k''}), (A_{k'}, n_{k'}))$, we obtain $\mathfrak{D}'_5$. It is easy to check that there exist no $n_1 \neq n_2$ and atoms A, B such that $(\bigcirc^{i_1} A_1 \to A, n_1), \dots, (\bigcirc^{i_2} A_2 \to B, n_1)$ and $(\bigcirc^{i_3} A_3 \to A, n_2), \dots, (\bigcirc^{i_4} A_4 \to B, n_2)$ are in $\mathfrak{D}'_5$. Therefore, $|n_{j'} - n_{m+1}| \leq 2M^2$ for all numbers $n_{j'}$ in $\mathfrak{D}'_5$. If the latter case holds for the earliest j, we can transform the subsequence $\mathfrak{D}_5$ of $\mathfrak{D}_2$ between j (not inclusive) and $j+k$ into the subsequence $\mathfrak{D}'_5$ with all numbers $|n_{j'} - n_0| \leq 2M^2$. Then we find j in $\mathfrak{D}_6$ satisfying one of the two cases above and transform $\mathfrak{D}_6$ analogously. We proceed until there are no more j satisfying either of the two cases and the result $\mathfrak{D}'_2$ of the transformation is as required by the proposition. ❑

To show (40), consider a derivation $\mathfrak{D}$ of A at ℓ, for $N \leq \ell < |\mathcal{A}| - N$, with the numbers n_j. Take the first n_j such that $n_j \geq |\mathcal{B}| + M$ or $n_j < 2M^2$. Suppose the former is the case. Since $\mathcal{A}_i = \emptyset$ for $|\emptyset^N \mathcal{B}| \leq i < |\mathcal{A}|$, there are $n_{j'}$, for $j' < j$, such that $2M^2 \leq n_{j'} < |\mathcal{B}| + M$ and a (sub)sequence $(\bigcirc^{i_{j'}} A_{j'} \to A_{j'+1}, n_{j'+1}), \dots, (\bigcirc^{i_{j-1}} A_{j-1} \to A_j, n_j)$ is in $\mathfrak{D}$. We expand this subsequence by taking all $(\bigcirc^{i_j} A_j \to A_{j+1}, n_j), \dots, (\bigcirc^{i_{j''-1}} A_{j''-1} \to A_{j''}, n_{j''})$, such that j'' is the first after j such that $n_{j''} = n_{j'}$. Let $\mathfrak{D} = \mathfrak{D}_1\mathfrak{D}_2\mathfrak{D}_3$, where $\mathfrak{D}_2$ is the expanded sequence above. By applying Proposition 36, we obtain a derivation $\mathfrak{D}_1\mathfrak{D}'_2\mathfrak{D}_3$ of A at ℓ, where all numbers n_j in $\mathfrak{D}_1\mathfrak{D}'_2$ are such that $2M^2 \leq n_j \leq n_{j'} + 2M^2 < |\mathcal{A}|$. In case $n_j < 2M^2$, we analogously obtain a derivation of A at ℓ, where all numbers n_j in $\mathfrak{D}_1\mathfrak{D}'_2$ are such that $0 \leq n_{j'} - 2M^2 \leq n_j < |\mathcal{B}| + M$. By continuing to apply Proposition 36 to $\mathfrak{D}_3$ the required number of times, we obtain a derivation of A at ℓ satisfying (40).

This completes the proof of Lemma 23 as, clearly, for any ℓ with $N \leq \ell < |\mathcal{A}| - N$, there is a run $(q_0, 0), \dots, (q, \ell), (q_A, i)$ of $\mathfrak{A}^{\Xi}_{\mathcal{O}}$ on $\mathcal{A}$ iff there is a derivation of A at ℓ such that $0 \leq n_j < |\mathcal{A}|$ for all n_j in it. ❑

A.5 Proof of Theorem 26

Theorem 26. *Let $\boldsymbol{q} = (\mathcal{O}, \varkappa)$ be an OMPQ with a $\bot$-free $LTL^{\Box\bigcirc}_{horn}$-ontology $\mathcal{O}$. Then $\boldsymbol{q}$ is not* $\mathsf{FO}(<, \equiv)$*-rewritable over Ξ-Aboxes iff there are $\mathcal{A}, \mathcal{B}, \mathcal{D} \in \Sigma^*_\Xi$ and $k \geq 2$ such that (i) and (ii) from Theorem 24 hold and there are $\mathcal{U}, \mathcal{V} \in \Sigma^*_\Xi$ such that $\mathcal{B} = \mathcal{V}\mathcal{U}$, $|\mathcal{U}| = |\mathcal{V}|$,*

(iii) $\boldsymbol{\tau}_{\mathcal{O}, \mathcal{A}\mathcal{B}^k\mathcal{D}}(|\mathcal{A}\mathcal{B}^i| - 1) = \boldsymbol{\tau}_{\mathcal{O}, \mathcal{A}\mathcal{B}^k\mathcal{D}}(|\mathcal{A}\mathcal{B}^i\mathcal{V}| - 1)$*, for all $i < k$, and*

(iv) $\boldsymbol{\tau}_{\mathcal{O}, \mathcal{A}\mathcal{B}^{k+1}\mathcal{D}}(|\mathcal{A}\mathcal{B}^i| - 1) = \boldsymbol{\tau}_{\mathcal{O}, \mathcal{A}\mathcal{B}^{k+1}\mathcal{D}}(|\mathcal{A}\mathcal{B}^i\mathcal{V}| - 1)$*, for all i with $1 \leq i \leq k$.*

Proof. Consider the DFA $\mathfrak{A} = (Q, \Sigma, \delta, q_{-1}, F)$ from the proof of Theorem 24 such that $\boldsymbol{L}_\Xi(\boldsymbol{q}) = \boldsymbol{L}(\mathfrak{A})$. ($\Rightarrow$) Suppose $\boldsymbol{q}$ is not $\mathsf{FO}(<,\equiv)$-rewritable. By Theorem 6 (*ii*), there exist $\mathcal{A}, \mathcal{V}, \mathcal{U}, \mathcal{D} \in \Sigma_\Xi^*$ with $|\mathcal{U}| = |\mathcal{V}|$ and $k \geq 2$ such that

$$q_{-1} \Rightarrow_{\mathcal{A}} q_0 \Rightarrow_{\mathcal{V}} q_0 \Rightarrow_{\mathcal{U}} q_1 \Rightarrow_{\mathcal{V}} q_1 \Rightarrow_{\mathcal{U}} \cdots \Rightarrow_{\mathcal{U}} q_{k-1} \Rightarrow_{\mathcal{V}} q_{k-1} \Rightarrow_{\mathcal{U}} q_0,$$

$q_0 \Rightarrow_{\mathcal{D}} r_0$, $q_1 \Rightarrow_{\mathcal{D}} r_1$ for some $q_0, \dots, q_{k-1}, r_0, r_1 \in Q$ with $r_0 \notin F$ and $r_1 \in F$. That (*i*) and (*ii*) are satisfied for $\mathcal{B} = \mathcal{V}\mathcal{U}$ is shown as in the proof of Theorem 24. Then (*iii*) and (*iv*) easily follow from (36).

($\Leftarrow$) Suppose (*i*)–(*iv*) hold and set $\mathcal{E}(i_0, \dots, i_j) = \mathcal{V}^{i_0}\mathcal{U} \dots \mathcal{V}^{i_j}\mathcal{U}$. Let $\mathcal{F}_{j'}(i_0, \dots, i_j)$ be the prefix of $\mathcal{E}(i_0, \dots, i_j)$ of the form $\mathcal{V}^{i_0}\mathcal{U} \dots \mathcal{V}^{i_{j'}-1}\mathcal{U}\mathcal{V}^{i_{j'}}$, for $j' \leq j$. By the properties of the canonical models, we then obtain the following, for any $0 \leq n \leq m$ and any $0 \leq \ell < k$:

(a) $\boldsymbol{\tau}_{\mathcal{O},\mathcal{A}\mathcal{E}(i_0,\dots,i_{km+k-1})\mathcal{D}}(|\mathcal{A}\mathcal{F}_{kn+\ell}(i_0, \dots, i_{km+k-1})| - 1) = \boldsymbol{\tau}_{\mathcal{O},\mathcal{A}\mathcal{B}^k\mathcal{D}}(|\mathcal{A}\mathcal{B}^\ell| - 1),$,

(b) $\boldsymbol{\tau}_{\mathcal{O},\mathcal{A}\mathcal{E}(i_0,\dots,i_{km+k-1},i_0)\mathcal{D}}(|\mathcal{A}\mathcal{F}_{kn+\ell+1}(i_0, \dots, i_{km+k-1}, i_0)| - 1) = \boldsymbol{\tau}_{\mathcal{O},\mathcal{A}\mathcal{B}^{k+1}\mathcal{D}}(|\mathcal{A}\mathcal{B}^{\ell+1}| - 1).$

The rest of the proof relies on the following observation:

Proposition 37. *Let $\mathfrak{A}$ be a DFA with a set of states Q, $|Q| \geq 3$, over an alphabet Σ. Then, for any $q \in Q$ and $w \in \Sigma^*$, there exists q' such that $q \Rightarrow_{w^{|Q|!-1}} q' \Rightarrow_{w^{|Q|!}} q'$.*

Take the DFA $\mathfrak{A}$ from the proof of Theorem 24, assume without loss of generality that $|Q| \geq 3$, and, for $m \geq 0$, consider the sequence

$$q_{-1} \Rightarrow_{\mathcal{A}\mathcal{V}^{|Q|!-1}} q_0 \Rightarrow_{\mathcal{V}^{|Q|!}} q_0' \Rightarrow_{\mathcal{U}} q_0'' \Rightarrow_{\mathcal{V}^{|Q|!-1}} q_1 \Rightarrow_{\mathcal{V}^{|Q|!}} q_1' \Rightarrow_{\mathcal{U}} q_1'' \Rightarrow_{\mathcal{V}^{|Q|!-1}} \dots$$
$$q_{km+k-1} \Rightarrow_{\mathcal{V}^{|Q|!}} q_{km+k-1}' \Rightarrow_{\mathcal{U}} q_{km+k}.$$

By Proposition 37, $q_i = q_i'$ for $0 \leq i < km + k$. By taking an appropriate m, as in the proof of Lemma 24, we can find i and j such that

$$q_{-1} \Rightarrow_{\mathcal{A}\mathcal{V}^{|Q|!-1}(\mathcal{U}\mathcal{V}^{|Q|!-1})^{ik}} r_0 \Rightarrow_{\mathcal{U}\mathcal{V}^{|Q|!-1}} r_1 \Rightarrow_{\mathcal{U}\mathcal{V}^{|Q|!-1}} \cdots \Rightarrow_{\mathcal{U}\mathcal{V}^{|Q|!-1}} r_{jk+k-1} \Rightarrow_{\mathcal{U}\mathcal{V}^{|Q|!-1}} r_0$$

and $r_\ell \Rightarrow_{\mathcal{V}^{|Q|!}} r_\ell$, for $0 \leq \ell < jk + k$. It can be readily shown using (a) and (b) that $q_0' \notin F$ and $q_1' \in F$ for such q_0' and q_1' that $r_0 \Rightarrow_{\mathcal{D}} q_0'$ and $r_1 \Rightarrow_{\mathcal{D}} q_1'$. Now, we have found a state r_0 in $\mathfrak{A}$ that satisfies the condition of Theorem 6 (*ii*) with $u = \mathcal{U}\mathcal{V}^{|Q|!-1}$ and $v = \mathcal{V}^{|Q|!}$. Therefore, $\boldsymbol{q}$ is not $\mathsf{FO}(<,\equiv)$-rewritable. ❑

A.6 Proof of Lemma 33

Lemma 33. *Given $a_1, \dots, a_l \in \Sigma_\Xi$ with $|a_i| = 1$, for $1 \leq i \leq l$, binary numbers $i_1, \dots, i_{l+1}, j$, a $\bot$-free $LTL^{\bigcirc}_{core}$-ontology $\mathcal{O}$ and a positive existential temporal concept $\varkappa$, checking whether $\mathcal{C}_{\mathcal{O},\mathcal{A}} \models \varkappa(j)$ for $\mathcal{A} = \emptyset^{i_1}a_1 \dots \emptyset^{i_l}a_l\emptyset^{i_{l+1}}$ can be done in* NP.

Proof. We first show that, for any ABox $\mathcal{A}$, we have $\mathcal{C}_{\mathcal{O},\mathcal{A}} \models \varkappa(j)$ iff there exist numbers n, n' and k, k' with $0 < n, k \leq |\{\Diamond\varkappa' \in \mathsf{sub}(\varkappa)\}| + 3$, $0 \leq n' < n$, $0 \leq k' < k$, a set of numbers $\{j_{-k}, \dots, j_{-1}, j_0, j_1, \dots j_n\} \subseteq \mathbb{Z}$, and types $\boldsymbol{\tau}_{-k}, \dots, \boldsymbol{\tau}_{-1}, \boldsymbol{\tau}_0, \boldsymbol{\tau}_1, \dots \boldsymbol{\tau}_n$ for $(\mathcal{O}, \varkappa)$ such that:

– $j_i < j_{i+1}$, for all i with $-k \leq i < n$, and $j_0 = j$;

- $j_{i+1} - j_i \leq 2^{O(|\boldsymbol{q}|)}$ if $j_i > \max \mathcal{A}$ or $j_{i+1} < 0$;
- $\boldsymbol{\tau}^{\mathsf{sig}(\mathcal{O})}_{\mathcal{O},\mathcal{A}}(j_i) \subseteq \boldsymbol{\tau}_i$, for all i with $-k \leq i < n$, and $\varkappa \in \boldsymbol{\tau}_0$;
- $\boldsymbol{\tau}_n = \boldsymbol{\tau}_{n'}$ and $\boldsymbol{\tau}_{-k} = \boldsymbol{\tau}_{-k'}$;
- for all $i < n'$ and $\Diamond_F \varkappa' \in \mathsf{sub}(\varkappa)$, $\Diamond_F \varkappa' \in \boldsymbol{\tau}_i$ implies $\varkappa' \in \boldsymbol{\tau}_{i'}$ for some $i' \in (i, n]$;
- for all $i \in [n', n]$ and $\Diamond_F \varkappa' \in \mathsf{sub}(\varkappa)$, $\Diamond_F \varkappa' \in \boldsymbol{\tau}_i$ implies $\varkappa' \in \boldsymbol{\tau}_{i'}$ for some $i' \in [n', n]$;
- for all $i < n'$ and $\Diamond_F \varkappa' \in \mathsf{sub}(\varkappa)$, $\varkappa' \in \boldsymbol{\tau}_i$ implies $\Diamond_F \varkappa' \in \boldsymbol{\tau}_{i'}$ for all $i' < i$;
- for all $i \in [n', n]$ and $\Diamond_F \varkappa' \in \mathsf{sub}(\varkappa)$, $\varkappa' \in \boldsymbol{\tau}_i$ implies $\Diamond_F \varkappa' \in \boldsymbol{\tau}_{i'}$ for all $i' \in [n', n]$,

and similarly for $\Diamond_P \varkappa'$ formulas.

($\Rightarrow$) Suppose $(\mathcal{O}, \mathcal{A}) \models \varkappa(j)$, so $\varkappa \in \boldsymbol{\tau}_{\mathcal{O},\mathcal{A}}(j)$. Let Φ be the set of $\Diamond \varkappa' \in \boldsymbol{\tau}_{\mathcal{O},\mathcal{A}}(j)$ for which there exist (unique) $j_{\varkappa'}$ satisfying $\neg\Diamond \varkappa', \varkappa' \in \boldsymbol{\tau}_{\mathcal{O},\mathcal{A}}(j_{\varkappa'})$. Let $\{j_{\varkappa'} \mid \Diamond \varkappa' \in \Phi\} \cup \{j\} = \{j_{-k'}, \dots, j_{-1}, j_0, j_1, \dots, j_{n'}\}$ such that $j_0 = j$ $j_{-k'} < j_{-k'+1} < \dots < j_{n'-1} < j_{n'}$. We take the smallest numbers $j_{n'+1}$ and j'' exceeding $\max \mathcal{A}$ for which $\boldsymbol{\tau}_{\mathcal{O},\mathcal{A}}(j_{n'+1}) = \boldsymbol{\tau}_{\mathcal{O},\mathcal{A}}(j'')$ and $j'' > j_{n'+1}$. Let Ψ be the set of all $\Diamond_F \varkappa' \in \boldsymbol{\tau}_{\mathcal{O},\mathcal{A}}(j_{n'+1})$. For each $\Diamond_F \varkappa' \in \Psi$, we take the smallest $j_{\varkappa'} \in (j_{n'+1}, j'']$ with $\varkappa' \in \boldsymbol{\tau}_{\mathcal{O},\mathcal{A}}(j_{\varkappa'})$. Let $\{j_{\varkappa'} \mid \Diamond_F \varkappa' \in \Psi\} = \{j_{n'+2}, \dots, j_{n-1}\}$. Finally, we set $j_n = j''$ (for the appropriate n). The selection of k and $j_{-k}, \dots, j_{-k'-1}$ is analogous and left to the reader. We take $\boldsymbol{\tau}_i = \boldsymbol{\tau}_{\mathcal{O},\mathcal{A}}(j_i)$, for $i \in [-k, n]$. Using the periodicity property of the canonical models (e.g., Artale et al., 2021, Lemma 22), one can check that the required conditions are satisfied.

($\Leftarrow$) Suppose there are n, m, n', m', j_i and $\boldsymbol{\tau}_i$ satisfying the conditions above. It is easy to check by induction on the construction of $\varkappa'$ that $\varkappa' \in \boldsymbol{\tau}_i$ implies $\varkappa' \in \boldsymbol{\tau}_{\mathcal{O},\mathcal{A}}(j_i)$ for all $\varkappa' \in \mathsf{sub}(\varkappa)$. As $\varkappa \in \boldsymbol{\tau}_0$, it follows that $(\mathcal{O}, \mathcal{A}) \models \varkappa(j)$.

It is now easy to provide the required NP algorithm. Indeed, we first guess the required binary numbers j_i and types (recall that $j_{i+1} - j_i \leq 2^{O(|\mathcal{O}|+|\varkappa|)}$ if $j_i > \max \mathcal{A}$ or $j_{i+1} < 0$). The list of conditions above can be checked in polynomial time. In particular, $\boldsymbol{\tau}^{\mathsf{sig}(\mathcal{O})}_{\mathcal{O},\mathcal{A}}(j_i) \subseteq \boldsymbol{\tau}_i$ for $\mathcal{A} = \emptyset^{i_1} a_1 \dots \emptyset^{i_l} a_l \emptyset^{i_{l+1}}$ can be checked in polynomial time using arithmetic progressions (e.g., Artale et al., 2021, Theorem 14). ❑

References

Abiteboul, S., Hull, R., & Vianu, V. (1995). *Foundations of Databases*. Addison-Wesley.

Afrati, F. N., & Papadimitriou, C. H. (1993). The parallel complexity of simple logic programs. *J. ACM*, *40*(4), 891–916.

Alur, R., & Henzinger, T. A. (1993). Real-time logics: Complexity and expressiveness. *Inf. Comput.*, *104*(1), 35–77.

Arora, S., & Barak, B. (2009). *Computational Complexity: A Modern Approach*. Cambridge University Press, New York, NY, USA.

Artale, A., Kontchakov, R., Ryzhikov, V., & Zakharyaschev, M. (2013). The complexity of clausal fragments of LTL. In *Proc. of the 19th Int. Conf. on Logic for Programming, Artificial Intelligence and Reasoning, LPAR 2013*, Vol. 8312 of *Lecture Notes in Computer Science*, pp. 35–52. Springer.

Artale, A., Calvanese, D., Kontchakov, R., & Zakharyaschev, M. (2009). The DL-Lite family and relations. *J. Artif. Intell. Res.*, *36*, 1–69.

Artale, A., Kontchakov, R., Kovtunova, A., Ryzhikov, V., Wolter, F., & Zakharyaschev, M. (2015). First-order rewritability of temporal ontology-mediated queries. In *Proc. of the 24th Int. Joint Conference on Artificial Intelligence, IJCAI'15*, pp. 2706–2712.

Artale, A., Kontchakov, R., Kovtunova, A., Ryzhikov, V., Wolter, F., & Zakharyaschev, M. (2017). Ontology-mediated query answering over temporal data: A survey (invited talk). In Schewe, S., Schneider, T., & Wijsen, J. (Eds.), *24th International Symposium on Temporal Representation and Reasoning, TIME 2017, October 16-18, 2017, Mons, Belgium*, Vol. 90 of *LIPIcs*, pp. 1:1–1:37. Schloss Dagstuhl - Leibniz-Zentrum für Informatik.

Artale, A., Kontchakov, R., Kovtunova, A., Ryzhikov, V., Wolter, F., & Zakharyaschev, M. (2021). First-order rewritability of ontology-mediated queries in linear temporal logic. *Artif. Intell.*, *299*, 103536.

Artale, A., Kontchakov, R., Kovtunova, A., Ryzhikov, V., Wolter, F., & Zakharyaschev, M. (2022). First-order rewritability and complexity of two-dimensional temporal ontology-mediated queries. *J. Artif. Intell. Res.*, *75*, 1223–1291.

Baget, J.-F., Leclère, M., Mugnier, M.-L., & Salvat, E. (2011). On rules with existential variables: Walking the decidability line. *Artif. Intell.*, *175*(9–10), 1620–1654.

Barrington, D. A. M. (1989). Bounded-width polynomial-size branching programs recognize exactly those languages in NC^1. *J. Comput. Syst. Sci.*, *38*(1), 150–164.

Barrington, D. A. M., Compton, K. J., Straubing, H., & Thérien, D. (1992). Regular languages in NC^1. *J. Comput. Syst. Sci.*, *44*(3), 478–499.

Barrington, D. A. M., & Thérien, D. (1988). Finite monoids and the fine structure of NC^1. *J. ACM*, *35*(4), 941–952.

Beaudry, M., McKenzie, P., & Thérien, D. (1992). The membership problem in aperiodic transformation monoids. *J. ACM*, *39*(3), 599–616.

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

Bennett, M., Martin, G., O'Bryant, K., & Rechnitzer, A. (2018). Explicit bounds for primes in arithmetic progressions. *Illinois Journal of Mathematics*, *62*(1–4), 427–532.

Bernátsky, L. (1997). Regular expression star-freeness is PSPACE-complete. *Acta Cybern.*, *13*(1), 1–21.

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

Bienvenu, M., Kikot, S., Kontchakov, R., Podolskii, V. V., Ryzhikov, V., & Zakharyaschev, M. (2017). The complexity of ontology-based data access with OWL 2 QL and bounded treewidth queries. In *Proc. of the 36th ACM SIGMOD-SIGACT-SIGAI Symposium on Principles of Database Systems, PODS 2017*, pp. 201–216. ACM.

Bienvenu, M., Kikot, S., Kontchakov, R., Podolskii, V. V., & Zakharyaschev, M. (2018). Ontology-mediated queries: Combined complexity and succinctness of rewritings via circuit complexity. *J. ACM*, *65*(5), 28:1–28:51.

Borgwardt, S., Forkel, W., & Kovtunova, A. (2019). Finding new diamonds: Temporal minimal-world query answering over sparse aboxes. In Fodor, P., Montali, M., Calvanese, D., & Roman, D. (Eds.), *Rules and Reasoning - Third International Joint Conference, RuleML+RR 2019, Bolzano, Italy, September 16-19, 2019, Proceedings*, Vol. 11784 of *Lecture Notes in Computer Science*, pp. 3–18. Springer.

Bourhis, P., & Lutz, C. (2016). Containment in monadic disjunctive datalog, MMSNP, and expressive description logics. In Baral, C., Delgrande, J. P., & Wolter, F. (Eds.), *Principles of Knowledge Representation and Reasoning: Proceedings of the Fifteenth International Conference, KR 2016, Cape Town, South Africa, April 25-29, 2016*, pp. 207–216. AAAI Press.

Brandt, S., Calvanese, D., Kalayci, E. G., Kontchakov, R., Mörzinger, B., Ryzhikov, V., Xiao, G., & Zakharyaschev, M. (2019). Two-dimensional rule language for querying sensor log data: A framework and use cases. In Gamper, J., Pinchinat, S., & Sciavicco, G. (Eds.), *26th International Symposium on Temporal Representation and Reasoning, TIME 2019, October 16-19, 2019, Málaga, Spain*, Vol. 147 of *LIPIcs*, pp. 7:1–7:15. Schloss Dagstuhl - Leibniz-Zentrum für Informatik.

Brandt, S., Kalayci, E. G., Ryzhikov, V., Xiao, G., & Zakharyaschev, M. (2018). Querying log data with metric temporal logic. *J. Artif. Intell. Res.*, *62*, 829–877.

Calì, A., Gottlob, G., & Pieris, A. (2012). Towards more expressive ontology languages: The query answering problem. *Artif. Intell.*, *193*, 87–128.

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

Carton, O., & Dartois, L. (2015). Aperiodic Two-way Transducers and FO-Transductions. In Kreutzer, S. (Ed.), *24th EACSL Annual Conference on Computer Science Logic (CSL 2015)*, Vol. 41 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pp. 160–174, Dagstuhl, Germany. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik.

Cho, S., & Huynh, D. T. (1991). Finite-automaton aperiodicity is PSPACE-complete. *Theor. Comp. Sci.*, *88*(1), 99–116.

Civili, C., & Rosati, R. (2012). A broad class of first-order rewritable tuple-generating dependencies. In *Proc. of the 2nd Int. Datalog 2.0 Workshop*, Vol. 7494 of *Lecture Notes in Computer Science*, pp. 68–80. Springer.

Compton, K. J., & Laflamme, C. (1990). An algebra and a logic for NC^1. *Inf. Comput.*, *87*(1/2), 240–262.

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

Demri, S., Goranko, V., & Lange, M. (2016). *Temporal Logics in Computer Science*. Cambridge Tracts in Theoretical Computer Science. Cambridge University Press.

Feier, C., Kuusisto, A., & Lutz, C. (2019). Rewritability in monadic disjunctive datalog, MMSNP, and expressive description logics. *Log. Methods Comput. Sci.*, *15*(2).

Fisher, M., Dixon, C., & Peim, M. (2001). Clausal temporal resolution. *ACM Trans. Comput. Logic*, *2*(1), 12–56.

Fleischer, L., & Kufleitner, M. (2018). The intersection problem for finite monoids. In Niedermeier, R., & Vallée, B. (Eds.), *Proc. STACS 2018*, Vol. 96 of *LIPIcs*, pp. 30:1–30:14, Dagstuhl, Germany. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik.

Furst, M. L., Saxe, J. B., & Sipser, M. (1984). Parity, circuits, and the polynomial-time hierarchy. *Mathematical Systems Theory*, *17*(1), 13–27.

Gabbay, D., Kurucz, A., Wolter, F., & Zakharyaschev, M. (2003). *Many-Dimensional Modal Logics: Theory and Applications*, Vol. 148 of *Studies in Logic*. Elsevier.

Gerasimova, O., Kikot, S., Kurucz, A., Podolskii, V. V., & Zakharyaschev, M. (2020). A data complexity and rewritability tetrachotomy of ontology-mediated queries with a covering axiom. In Calvanese, D., Erdem, E., & Thielscher, M. (Eds.), *Proceedings of the 17th International Conference on Principles of Knowledge Representation and Reasoning, KR 2020, Rhodes, Greece, September 12-18, 2020*, pp. 403–413.

Gutiérrez-Basulto, V., & Jung, J. C. (2017). Combining DL-Lite$^{\mathrm{N}}_{\mathrm{bool}}$ with branching time: A gentle marriage. In Sierra, C. (Ed.), *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI 2017, Melbourne, Australia, August 19-25, 2017*, pp. 1074–1080. ijcai.org.

Hillebrand, G. G., Kanellakis, P. C., Mairson, H. G., & Vardi, M. Y. (1995). Undecidable boundedness problems for datalog programs. *J. Log. Program.*, *25*(2), 163–190.

Hodges, W. (1993). *Model theory*, Vol. 42 of *Encyclopedia of mathematics and its applications*. Cambridge University Press.

Jukna, S. (2012). *Boolean Function Complexity - Advances and Frontiers*, Vol. 27 of *Algorithms and combinatorics*. Springer.

Kaminski, M., Nenov, Y., & Cuenca Grau, B. (2016). Datalog rewritability of disjunctive datalog programs and non-Horn ontologies. *Artif. Intell.*, *236*, 90–118.

Kamp, H. W. (1968). *Tense Logic and the Theory of Linear Order*. PhD thesis, Computer Science Department, University of California at Los Angeles, USA.

Kaplan, G., & Levy, D. (2010). Solvability of finite groups via conditions on products of 2-elements and odd p-elements. *Bulletin of the Australian Mathematical Society*, *82*(2), 265–273.

Kikot, S., Kurucz, A., Podolskii, V. V., & Zakharyaschev, M. (2021). Deciding boundedness of monadic sirups. In Libkin, L., Pichler, R., & Guagliardo, P. (Eds.), *PODS'21: Proceedings of the 40th ACM SIGMOD-SIGACT-SIGAI Symposium on Principles of Database Systems, Virtual Event, China, June 20-25, 2021*, pp. 370–387. ACM.

King, O. H. (2005). The subgroup structure of finite classical groups in terms of geometric configurations. In Webb, B. S. (Ed.), *Surveys in Combinatorics, 2005 [invited lectures from the Twentieth British Combinatorial Conference, Durham, UK, July 2005]*, Vol. 327 of *London Mathematical Society Lecture Note Series*, pp. 29–56. Cambridge University Press.

Knuth, D. E. (1998). *The art of computer programming, Volume II: Seminumerical Algorithms, 3rd Edition*. Addison-Wesley.

Koymans, R. (1990). Specifying real-time properties with metric temporal logic. *Real-Time Systems*, *2*(4), 255–299.

Kozen, D. (1977). Lower bounds for natural proof systems. In *18th Annual Symposium on Foundations of Computer Science (SFCS 1977)*, pp. 254–266.

Libkin, L. (2004). *Elements Of Finite Model Theory*. Springer.

Lutz, C., Wolter, F., & Zakharyaschev, M. (2008). Temporal description logics: A survey. In *Proc. of the 15th Int. Symposium on Temporal Representation and Reasoning (TIME 2008)*, pp. 3–14.

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

Lutz, C., & Sabellek, L. (2019). A complete classification of the complexity and rewritability of ontology-mediated queries based on the description logic EL. *CoRR*, *abs/1904.12533*.

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

Marcinkowski, J. (1999). Achilles, turtle, and undecidable boundedness problems for small DATALOG programs. *SIAM J. Comput.*, *29*(1), 231–257.

McNaughton, R., & Papert, S. (1971). *Counter-free automata*. The MIT Press.

Ouaknine, J., & Worrell, J. (2008). Some recent results in metric temporal logic. In *Formal Modeling and Analysis of Timed Systems, 6th International Conference, FORMATS 2008, Saint Malo, France, September 15-17, 2008. Proceedings*, pp. 1–13.

Poggi, A., Lembo, D., Calvanese, D., De Giacomo, G., Lenzerini, M., & Rosati, R. (2008). Linking data to ontologies. *J. Data Semant.*, *X*, 133–173.

Rabinovich, A. (2014). A proof of Kamp's theorem. *Logical Methods in Computer Science*, *10*(1).

Rodriguez-Muro, M., Kontchakov, R., & Zakharyaschev, M. (2013). Ontology-based data access: Ontop of databases. In Alani, H., Kagal, L., Fokoue, A., Groth, P. T., Biemann, C., Parreira, J. X., Aroyo, L., Noy, N. F., Welty, C., & Janowicz, K. (Eds.), *The Semantic Web - ISWC 2013 - 12th International Semantic Web Conference, Sydney, NSW, Australia, October 21-25, 2013, Proceedings, Part I*, Vol. 8218 of *Lecture Notes in Computer Science*, pp. 558–573. Springer.

Rotman, J. J. (1999). *An introduction to the theory of groups.* Springer-Verlag, New York.

Ryzhikov, V., Walega, P. A., & Zakharyaschev, M. (2019). Data complexity and rewritability of ontology-mediated queries in metric temporal logic under the event-based semantics. In Kraus, S. (Ed.), *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI 2019, Macao, China, August 10-16, 2019*, pp. 1851–1857. ijcai.org.

Schützenberger, M. P. (1965). On finite monoids having only trivial subgroups. *Inf. Control.*, *8*(2), 190–194.

Shepherdson, J. C. (1959). The reduction of two-way automata to one-way automata. *IBM J. of Research and Development*, *3*(2), 198–200.

Stern, J. (1985). Complexity of some problems from the theory of automata. *Inf. Control.*, *66*(3), 163–176.

Stockmeyer, L. J., & Meyer, A. R. (1973). Word problems requiring exponential time: Preliminary report. In Aho, A. V., Borodin, A., Constable, R. L., Floyd, R. W., Harrison, M. A., Karp, R. M., & Strong, H. R. (Eds.), *Proceedings of the 5th Annual ACM Symposium on Theory of Computing, April 30 - May 2, 1973, Austin, Texas, USA*, pp. 1–9. ACM.

Straubing, H. (1994). *Finite Automata, Formal Logic, and Circuit Complexity.* Birkhauser Verlag.

Tahrat, S., Braun, G. A., Artale, A., Gario, M., & Ozaki, A. (2020). Automated reasoning in temporal DL-Lite (extended abstract). In Borgwardt, S., & Meyer, T. (Eds.), *Proceedings of the 33rd International Workshop on Description Logics (DL 2020) co-located with the 17th International Conference on Principles of Knowledge Representation and Reasoning (KR 2020), Online Event [Rhodes, Greece], September 12th to 14th, 2020*, Vol. 2663 of *CEUR Workshop Proceedings.* CEUR-WS.org.

Tena Cucala, D. J., Walega, P. A., Cuenca Grau, B., & Kostylev, E. V. (2021). Stratified negation in datalog with metric temporal operators. In *Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Thirty-Third Conference on Innovative Applications of Artificial Intelligence, IAAI 2021, The Eleventh Symposium on Educational Advances in Artificial Intelligence, EAAI 2021, Virtual Event, February 2-9, 2021*, pp. 6488–6495. AAAI Press.

Thompson, J. G. (1968). Nonsolvable finite groups all of whose local subgroups are solvable. *Bull. Amer. Math. Soc.*, *74*(3), 383–437.

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

van der Meyden, R. (2000). Predicate boundedness of linear monadic datalog is in PSPACE. *Int. J. Found. Comput. Sci.*, *11*(4), 591–612.

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

Vardi, M. Y. (1989). A note on the reduction of two-way automata to one-way atuomata. *Inf. Process. Lett.*, *30*(5), 261–264.

Vardi, M. Y. (2007). Automata-theoretic techniques for temporal reasoning. In Blackburn, P., van Benthem, J. F. A. K., & Wolter, F. (Eds.), *Handbook of Modal Logic*, Vol. 3 of *Studies in logic and practical reasoning*, pp. 971–989. North-Holland.

Vardi, M. Y., & Wolper, P. (1986). An automata-theoretic approach to automatic program verification (preliminary report). In *Proc. of the Symposium on Logic in Computer Science (LICS'86)*, pp. 332–344.

Walega, P. A., Cuenca Grau, B., Kaminski, M., & Kostylev, E. V. (2020a). Datalogmtl over the integer timeline. In Calvanese, D., Erdem, E., & Thielscher, M. (Eds.), *Proceedings of the 17th International Conference on Principles of Knowledge Representation and Reasoning, KR 2020, Rhodes, Greece, September 12-18, 2020*, pp. 768–777.

Walega, P. A., Cuenca Grau, B., Kaminski, M., & Kostylev, E. V. (2020b). Tractable fragments of datalog with metric temporal operators. In Bessiere, C. (Ed.), *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI 2020*, pp. 1919–1925. ijcai.org.

Wang, D., Hu, P., Walega, P. A., & Cuenca Grau, B. (2022). Meteor: Practical reasoning in datalog with metric temporal operators. *CoRR*, *abs/2201.04596*.

Xiao, G., Calvanese, D., Kontchakov, R., Lembo, D., Poggi, A., Rosati, R., & Zakharyaschev, M. (2018). Ontology-based data access: A survey. In Lang, J. (Ed.), *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI 2018, July 13-19, 2018, Stockholm, Sweden.*, pp. 5511–5519. ijcai.org.

Xiao, G., Ding, L., Cogrel, B., & Calvanese, D. (2019). Virtual knowledge graphs: An overview of systems and use cases. *Data Intell.*, *1*(3), 201–223.

Xiao, G., Lanti, D., Kontchakov, R., Komla-Ebri, S., Kalayci, E. G., Ding, L., Corman, J., Cogrel, B., Calvanese, D., & Botoeva, E. (2020). The virtual knowledge graph system Ontop. In Pan, J. Z., Tamma, V. A. M., d'Amato, C., Janowicz, K., Fu, B., Polleres, A., Seneviratne, O., & Kagal, L. (Eds.), *The Semantic Web - ISWC 2020 - 19th International Semantic Web Conference, Athens, Greece, November 2-6, 2020, Proceedings, Part II*, Vol. 12507 of *Lecture Notes in Computer Science*, pp. 259–277. Springer.