Ontology-Mediated Query Answering over Temporal Data: A Survey

Alessandro Artale[1], Roman Kontchakov[2], Alisa Kovtunova[1], Vladislav Ryzhikov[1], Frank Wolter[3], and Michael Zakharyaschev[2]

1 **KRDB Research Centre, Free University of Bozen-Bolzano, Italy**
{artale,alisa.kovtunova,ryzhikov}@inf.unibz.it
2 **Department of Computer Science and Information Systems, Birkbeck, University of London, UK**
{roman,michael}@dcs.bbk.ac.uk
3 **Department of Computer Science, University of Liverpool, UK**
wolter@liverpool.ac.uk

Abstract

We discuss the use of various temporal knowledge representation formalisms for ontology-mediated query answering over temporal data. In particular, we analyse ontology and query languages based on the linear temporal logic *LTL*, the multi-dimensional Halpern-Shoham interval temporal logic $\mathcal{HS}_n$, as well as the metric temporal logic *MTL*. Our main focus is on the data complexity of answering temporal ontology-mediated queries and their rewritability into standard first-order and datalog queries.

1998 ACM Subject Classification I.2.4 Knowledge Representation Formalisms and Methods.

Keywords and phrases Description Logic, Temporal Logic, Ontology Mediated Query Answering, Data Complexity.

Digital Object Identifier 10.4230/LIPIcs.TIME.2017.1

1 Introduction

This paper is a survey of recent developments in applying temporal logics for ontology-mediated query answering over temporal data.

Ontology-based data access (OBDA) [73] has recently become one of the most successful applications of description logics (DLs). The chief aim of OBDA is to facilitate access to possibly heterogeneous, distributed and incomplete data for non-IT-expert users. To illustrate, suppose that such a user wants to query some data sources $\mathcal{D}$. Under the OBDA paradigm, the user does not have to know the schemas of $\mathcal{D}$ (that is, how the data is organised). Instead, the user is given an ontology $\mathcal{O}$ describing the domain of their interest in familiar and standard terms that can be used directly to formulate the desired queries $\boldsymbol{q}(\boldsymbol{x})$ in, say, the query language SPARQL, possibly with the help of a graphical tool. The OBDA system relies on a (GAV) mapping $\mathcal{M}$, relating the terms in $\mathcal{O}$ with the schemas of $\mathcal{D}$ (and produced by an IT expert), to find tuples $\boldsymbol{a}$ from $\mathcal{D}$ such that $\mathcal{O}, \mathcal{M}(\mathcal{D}) \models \boldsymbol{q}(\boldsymbol{a})$, where $\mathcal{M}(\mathcal{D})$ is the result of applying $\mathcal{M}$ to $\mathcal{D}$. Depending on the language of the *ontology-mediated query* (OMQ) $\boldsymbol{Q} = (\mathcal{O}, \boldsymbol{q}(\boldsymbol{x}))$, this can sometimes be done by rewriting $\boldsymbol{Q}$ to a first-order (FO) or datalog query $\boldsymbol{q}'(\boldsymbol{x})$ that can be executed over any given data instance $\mathcal{D}$ directly by conventional data management systems. For example, FO-rewritings always exist if $\mathcal{O}$ is an *OWL 2 QL* ontology (based on the *DL-Lite* family of DLs) and $\boldsymbol{q}(\boldsymbol{x})$ is a conjunctive query (CQ) [37, 6], while datalog rewritings can be constructed for OMQs with ontologies in

OWL 2 EL (based on the $\mathcal{EL}$ family of DLs) and CQs [78]. For recent applications of OBDA, the reader can consult [5, 22, 47, 36, 38, 77, 82].

The W3C standard ontology languages *OWL 2 QL* and *OWL 2 EL* mentioned above were designed to represent knowledge about static domains and are not suitable when the data and the terms the user is interested in are essentially temporal. Suppose, for example, that the data comprises sensor readings from some industrial installations, say, gas turbines, or from weather stations across a country, and that the user—a service engineer or, respectively, a meteorologist—is interested in detecting events such as

- *active power trip*, which happens when the active power of a turbine was above 1.5MW for a period of at least 10 seconds, maximum 3 seconds after which there was a period of at least one minute where the active power was below 0.15MW; or
- *blizzard*, which happens when severe snowstorms with low temperatures and strong winds last for at least three hours.

To be able to represent these concepts, an ontology language clearly requires various temporal constructs that have been studied in the context of temporal representation and reasoning [44, 45, 41].

Combinations of DLs with temporal formalisms have been widely investigated since the pioneering work of Schmiedel [81] and Schild [80] in the early 1990s; we refer the reader to [45, 21, 7, 62] for surveys and [71, 10, 51, 52, 50, 14] for more recent developments. However, the main reasoning task targeted in this line of research was concept satisfiability rather than query answering and the general aim was to probe various combinations of temporal and DL constructs that ensure decidability of concept satisfiability with acceptable combined complexity.

In the context of answering OMQs, our main concern is their FO- or datalog-rewritability, and the data complexity of query evaluation, where the given OMQ is regarded to be fixed while the data varies. Thus, in this survey we focus on temporal data modelling and algorithmic properties of OMQ answering and do not discuss in any detail advances in temporal DLs not related to query answering. The plan for this paper is as follows. We distinguish three temporal data models and the corresponding languages for ontologies and queries: the discrete point-based approach where time is discrete and each fact comes with a time-point in which it holds true, the more general interval-based approach where facts are stamped with the interval in which they are true, and finally, a model based on a dense flow of time where the focus is on modelling and querying metric temporal properties. In Sections 2–4, we discuss the state of the art in point-based ontology-mediated query answering. The languages considered range from the full two-sorted FOL to time-centric languages based of *LTL*, domain-centric languages based on DLs, and combinations of both. In Section 5, we consider ontology-mediated query answering over interval-based models focussing on Halpern and Shoham's modal logic for time intervals. In Section 6, we discuss dense time and how a combination of datalog and metric operators can be used to model metric knowledge and support ontology-mediated querying in this case. We close in Section 7 with a discussion of practical issues in temporal ontology-mediated querying and a recent implementation.

2 Point-Based Temporal Ontology-Mediated Querying

Suppose we have a database on submission, acceptance and publication of papers in the area of computer science collected from various sources on the web and elsewhere. For instance,

the database may contain the facts

$$\mathsf{underSubmissionTo}(a, \mathrm{JACM}, \mathrm{Feb2016}), \qquad \mathsf{UnderSubmission}(b, \mathrm{Jan2016}),$$
$$\mathsf{acceptedIn}(c, \mathrm{JACM}, \mathrm{July2016}), \qquad \mathsf{Published}(c, \mathrm{Oct2016}),$$
$$\mathsf{authorOf}(\mathrm{Bob}, c, \mathrm{May2014})$$

stating that paper a was under submission to JACM in February 2016; paper b was under submission in January 2016 (to an unknown journal); paper c, authored by Bob in May 2014, was recorded as accepted by JACM in July 2016, and published (in some venue) in October 2016. Observe that the predicates in the snippet above have a timestamp as their last argument (e.g., Oct2016) and either one or two domain arguments (e.g., a, JACM). Following the description logic (DL) tradition, we call predicates with one domain argument *concepts* (e.g., Published) and predicates with two domain arguments *roles* (e.g., authorOf). A finite set of timestamped facts such as the snippet above is called a temporal ABox. In general, a *temporal ABox*, denoted $\mathcal{A}$, consists of assertions of the form

$$A_k(a_i, n), \qquad P_k(a_i, a_j, n),$$

where A_k is a concept name, P_k a role name, a_i and a_j individual names, and $n \in \mathbb{Z}$ a timestamp.

Now, we introduce models for temporal ABoxes. Let $T \subseteq \mathbb{Z}$ be a (possibly infinite) interval. A *T-interpretation* $\mathcal{I}$ is a structure

$$\mathcal{I} = \big((T, <),\ \Delta^{\mathcal{I}},\ P_1^{\mathcal{I}}, P_2^{\mathcal{I}}, \dots, A_1^{\mathcal{I}}, A_2^{\mathcal{I}} \dots, a_1^{\mathcal{I}}, a_2^{\mathcal{I}}, \dots\big)$$

such that $<$ is the standard linear order on $\mathbb{Z}$ restricted to T, $\Delta^{\mathcal{I}} \neq \emptyset$ is the interpretation domain, $P_k^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}} \times T$ and $A_k^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}} \times T$, for each k, and $a_i^{\mathcal{I}} \in \Delta^{\mathcal{I}}$, for each i (we assume rigid, or time-independent, interpretation of individual names) and $a_i^{\mathcal{I}} \neq a_j^{\mathcal{I}}$ whenever $i \neq j$ (thus, we make the *unique name assumption*). Note that the domain $\Delta^{\mathcal{I}}$ is time-independent. Time-dependent domains can be modelled using an 'existence predicate'; we refer the reader to [45, 44, 31] and references therein for a discussion of relevant domain assumptions in the literature on modal and temporal logic. For a temporal ABox $\mathcal{A}$, we say that a T-interpretation $\mathcal{I}$ *satisfies* $\mathcal{A}$ or that $\mathcal{I}$ is a *model* of $\mathcal{A}$ if T contains all timestamps n that occur in $\mathcal{A}$ and

$$(a_i^{\mathcal{I}}, n) \in A_k^{\mathcal{I}}, \ \text{ for all } A_k(a_i, n) \in \mathcal{A}, \quad \text{and} \quad (a_i^{\mathcal{I}}, a_j^{\mathcal{I}}, n) \in P_k^{\mathcal{I}}, \ \text{ for all } P_k(a_i, a_j, n) \in \mathcal{A}.$$

Let $\min \mathcal{A}$ and $\max \mathcal{A}$ be the minimal and, respectively, maximal integers occurring in $\mathcal{A}$. We assume without loss of generality that $\min \mathcal{A} = 0$. In what follows, we shall mostly be working with $\mathbb{Z}$-interpretations satisfying $\mathcal{A}$ (called *$\mathbb{Z}$-models of $\mathcal{A}$*), $\mathbb{N}$-interpretations satisfying $\mathcal{A}$ (called *$\mathbb{N}$-models of $\mathcal{A}$*) and $[\min \mathcal{A}, \max \mathcal{A}]$-interpretations satisfying $\mathcal{A}$ (called *ABox-fitting models of $\mathcal{A}$*).

The models $\mathcal{I}$ of a temporal ABox $\mathcal{A}$ reflect the *open-world assumption* underpinning ontology-mediated query answering: rather than assuming that the ABox contains all relevant domain individuals and time points, one admits additional domain individuals and time points that might be required to satisfy domain knowledge. Thus, $\mathbb{Z}$-models reflect the common sense view of time as being infinite in the past and the future. $\mathbb{N}$-models and ABox-fitting models reflect a more pragmatic approach and assume that the time points not used as timestamps (or are before/after any timestamped data) are irrelevant for querying the data.

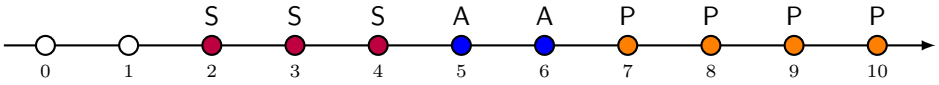

Figure 1 A typical timeline for a publication in Example 1: S, A and P stand for Submitted, Accepted and Published, respectively.

We next introduce the ontology and query languages that have been proposed for ontology-mediated querying of point-based temporal data. Most of these languages can be regarded as fragments of the *two-sorted first-order language* 2-FOL(<) [84] constructed from atoms $A_k(x,t)$, $P_k(x,y,t)$, $t_1 < t_2$, and $t_1 = t_2$, where A_k is a concept name, P_k a role name, x and y are *domain variables* ranging over the interpretation domain $\Delta^{\mathcal{I}}$, and t, t_1 and t_2 are *temporal variables* ranging over the time instants in T. For any 2-FOL(<)-formula φ, any T-interpretation $\mathcal{I}$, and any assignments $\mathfrak{d}$ of elements of $\Delta^{\mathcal{I}}$ to the domain variables and $\mathfrak{t}$ of elements of T to the temporal variables, we define the *truth-relation* $\mathcal{I} \models^{\mathfrak{d},\mathfrak{t}} \varphi$ by induction as follows:

$$
\begin{array}{rlrl}
\mathcal{I} \models^{\mathfrak{d},\mathfrak{t}} A_k(x,t) & \text{iff } (\mathfrak{d}(x),\mathfrak{t}(t)) \in A_k^{\mathcal{I}}, & \mathcal{I} \models^{\mathfrak{d},\mathfrak{t}} \top, & \\
\mathcal{I} \models^{\mathfrak{d},\mathfrak{t}} P_k(x,y,t) & \text{iff } (\mathfrak{d}(x),\mathfrak{d}(y),\mathfrak{t}(t)) \in P_k^{\mathcal{I}}, & \mathcal{I} \not\models^{\mathfrak{d},\mathfrak{t}} \bot, & \\
\mathcal{I} \models^{\mathfrak{d},\mathfrak{t}} t_1 < t_2 & \text{iff } \mathfrak{t}(t_1) < \mathfrak{t}(t_2), & \mathcal{I} \models^{\mathfrak{d},\mathfrak{t}} \neg\varphi & \text{iff } \mathcal{I} \not\models^{\mathfrak{d},\mathfrak{t}} \varphi, \\
\mathcal{I} \models^{\mathfrak{d},\mathfrak{t}} t_1 = t_2 & \text{iff } \mathfrak{t}(t_1) = \mathfrak{t}(t_2), & \mathcal{I} \models^{\mathfrak{d},\mathfrak{t}} \varphi_1 \wedge \varphi_2 & \text{iff } \mathcal{I} \models^{\mathfrak{d},\mathfrak{t}} \varphi_1 \text{ and } \mathcal{I} \models^{\mathfrak{d},\mathfrak{t}} \varphi_2, \\
\mathcal{I} \models^{\mathfrak{d},\mathfrak{t}} \forall x\, \varphi & \text{iff } \mathcal{I} \models^{\mathfrak{d}',\mathfrak{t}} \varphi, \text{ for all } \mathfrak{d}' \text{ that differ from } \mathfrak{d} \text{ only on } x, & & \\
\mathcal{I} \models^{\mathfrak{d},\mathfrak{t}} \forall t\, \varphi & \text{iff } \mathcal{I} \models^{\mathfrak{d},\mathfrak{t}'} \varphi, \text{ for all } \mathfrak{t}' \text{ that differ from } \mathfrak{t} \text{ only on } t; & &
\end{array}
$$

other first-order connectives and quantifiers such as $\rightarrow$, $\leftrightarrow$, $\exists$ are defined in the standard way. By an *ontology*, $\mathcal{O}$, we mean a set of 2-FOL(<)-sentences. We say that $\mathcal{I}$ *satisfies* an ontology $\mathcal{O}$ or $\mathcal{I}$ is a *model* of $\mathcal{O}$ if $\mathcal{I} \models \varphi$, for each $\varphi \in \mathcal{O}$ (since the ontology contains only sentences, the assignments are irrelevant).

▶ **Example 1.** Consider a simple temporal ontology about research papers (as above) with role names publishedIn, acceptedIn and underSubmissionTo. We state that the domains of the three roles are mutually disjoint using axioms such as

$$\forall t \forall x \forall y_1 \forall y_2 \left(\mathsf{publishedIn}(x,y_1,t) \wedge \mathsf{acceptedIn}(x,y_2,t) \rightarrow \bot\right). \tag{1}$$

Basic temporal dependencies can be formulated as follows:

$$\forall t \forall x \forall y \left(\mathsf{publishedIn}(x,y,t) \rightarrow \forall s \left((s > t) \rightarrow \mathsf{publishedIn}(x,y,s)\right)\right), \tag{2}$$

$$\begin{aligned} \forall t \forall x \forall y \big(&\mathsf{publishedIn}(x,y,t) \rightarrow \exists s \left((s < t) \wedge \mathsf{acceptedIn}(x,y,s) \wedge\right. \\ &\left.\exists s' ((s < s') \wedge \mathsf{publishedIn}(x,y,s') \wedge \neg \exists s'' \left((s < s'') \wedge (s'' < s')\right))\right)\big), \end{aligned} \tag{3}$$

$$\begin{aligned} \forall t \forall x \forall y \big(&\exists s' \left((s' < t) \wedge \mathsf{acceptedIn}(x,y,s')\right) \wedge \\ &\exists s'' \left((t < s'') \wedge \mathsf{acceptedIn}(x,y,s'')\right) \rightarrow \mathsf{acceptedIn}(x,y,t)\big), \end{aligned} \tag{4}$$

an analogue of (3) for acceptedIn and underSubmissionTo and the convexity axiom (4) for underSubmissionTo. The temporal ABox does not always use these role names, but rather integrates information from various data sources. For example, for a paper to be published it is necessary and sufficient that it is published in some venue (even if the publication venue is unknown). So, we use concept names Published, Accepted and UnderSubmission to refer to all published, accepted and submitted papers, respectively: e.g.,

$$\forall t \forall x \left(\mathsf{Published}(x,t) \leftrightarrow \exists y\, \mathsf{publishedIn}(x,y,t)\right) \tag{5}$$

and similarly for acceptedIn and underSubmissionTo. It follows from these axioms, in particular, that Submitted, Accepted and Published form consecutive intervals as depicted in Fig. 1.

A 2-FOL($<$) *ontology-mediated query* (OMQ) is a pair $\boldsymbol{Q}(\boldsymbol{x},\boldsymbol{t}) = (\mathcal{O}, \boldsymbol{q}(\boldsymbol{x},\boldsymbol{t}))$, where $\mathcal{O}$ is an ontology and $\boldsymbol{q}(\boldsymbol{x},\boldsymbol{t})$ a 2-FOL($<$)-formula with free domain variables $\boldsymbol{x}$ and free temporal variables $\boldsymbol{t}$. We call $\boldsymbol{q}(\boldsymbol{x},\boldsymbol{t})$ a *query* and $\boldsymbol{x}$, $\boldsymbol{t}$ its *answer variables*. Given a temporal ABox $\mathcal{A}$, a model $\mathcal{I}$ of $\mathcal{A}$, a tuple $\boldsymbol{a}$ of individual names in $\mathcal{A}$ of the same length as $\boldsymbol{x}$, and a tuple $\boldsymbol{n}$ of time points in $\mathcal{A}$ of the same length as $\boldsymbol{t}$, we write $\mathcal{I} \models \boldsymbol{q}(\boldsymbol{a},\boldsymbol{n})$ if $\mathcal{I} \models^{\mathfrak{d},\mathfrak{t}} \boldsymbol{q}(\boldsymbol{x},\boldsymbol{t})$, for the assignments $\mathfrak{d}\colon \boldsymbol{x} \mapsto \boldsymbol{a}$ and $\mathfrak{t}\colon \boldsymbol{t} \mapsto \boldsymbol{n}$. Let $T \in \{\mathbb{Z}, \mathbb{N}, [\min \mathcal{A}, \max \mathcal{A}]\}$. We say that the tuple $(\boldsymbol{a},\boldsymbol{n})$ is a *certain answer to* $\boldsymbol{Q} = (\mathcal{O}, \boldsymbol{q}(\boldsymbol{x},\boldsymbol{t}))$ *over* $\mathcal{A}$ *and* T and write $\mathcal{O}, \mathcal{A} \models_T \boldsymbol{q}(\boldsymbol{a},\boldsymbol{n})$ if

$\mathcal{I} \models \boldsymbol{q}(\boldsymbol{a},\boldsymbol{n})$ for all T-models $\mathcal{I}$ of $\mathcal{O}$ and $\mathcal{A}$.

▶ **Example 2.** In the context of Example 1, we now assume that the unit of time is one month. Then we can formulate the following queries.

- Find all accepted papers and their acceptance dates such that the paper was under submission for at least a year:

$$\boldsymbol{q}(x,t) = \mathsf{Accepted}(x,t) \land \mathsf{UnderSubmission}(x,t-1) \land \mathsf{UnderSubmission}(x,t-13). \quad (6)$$

  Since the flow of time is discrete, any formula of the form $P(\boldsymbol{x},t-1)$ is simply an abbreviation for $\exists t' \big[P(\boldsymbol{x},t') \land (t' < t) \land \neg \exists t'' \big((t' < t'') \land (t'' < t)\big)\big]$; $\mathsf{UnderSubmission}(x,t-13)$ can be defined similarly.
- Papers that were published within two months after acceptance but had been under submission for three years:

$$\boldsymbol{q}(x,t) = \exists s \big((s < t) \land \mathsf{Accepted}(x,s) \land \mathsf{UnderSubmission}(x,s-1) \land \mathsf{Published}(x,s+2) \land \mathsf{UnderSubmission}(x,s-37)\big). \quad (7)$$

- Authors of papers that were submitted more than two years ago but have not been accepted yet:

$$\boldsymbol{q}(x,t) = \exists y \big(\mathsf{authorOf}(x,y,t) \land \mathsf{UnderSubmission}(y,t-24) \land \mathsf{UnderSubmission}(y,t)\big). \quad (8)$$

  Recall that UnderSubmission is disjoint with Accepted and can only occur before the paper is eventually accepted.

Note that 2-FOL($<$)-formulas as we defined them do not use individual constants. This assumption is for simplicity only; it is straightforward to extend the syntax and semantics of temporal ontologies and queries to 2-FOL($<$) with individual constants.

Given two fragments $\mathcal{L}$ and $\mathcal{Q}$ of 2-FOL($<$), we denote by $(\mathcal{L},\mathcal{Q})$ the *class of ontology-mediated queries* $\boldsymbol{Q}(\boldsymbol{x},\boldsymbol{t}) = (\mathcal{O}, \boldsymbol{q}(\boldsymbol{x},\boldsymbol{t}))$ such that $\mathcal{O}$ is formulated in $\mathcal{L}$ and $\boldsymbol{q}(\boldsymbol{x},\boldsymbol{t})$ in $\mathcal{Q}$. Let T be any of $\mathbb{Z}$, $\mathbb{N}$ or $[\min \mathcal{A}, \max \mathcal{A}]$. By $(\mathcal{L},\mathcal{Q})$*-OMQ evaluation over* T we understand the problem of deciding, for a given $(\mathcal{L},\mathcal{Q})$-OMQ $\boldsymbol{Q}(\boldsymbol{x},\boldsymbol{t}) = (\mathcal{O}, \boldsymbol{q}(\boldsymbol{x},\boldsymbol{t}))$, a temporal ABox $\mathcal{A}$ and tuples $\boldsymbol{a}$ and $\boldsymbol{n}$ in $\mathcal{A}$ of the same length as $\boldsymbol{x}$ and $\boldsymbol{t}$, whether $\mathcal{O}, \mathcal{A} \models_T \boldsymbol{q}(\boldsymbol{a},\boldsymbol{n})$. The *combined complexity* of $(\mathcal{L},\mathcal{Q})$-OMQ evaluation over T is defined as the computational complexity of the above problem. As the queries and ontologies are mostly much smaller than the ABox $\mathcal{A}$, combined complexity is often misleading as a measure of the resources needed for query evaluation [85]. An alternative and often more appropriate complexity measure is the *data complexity* of $(\mathcal{L},\mathcal{Q})$-OMQ evaluation over T, that is the complexity of deciding, for *fixed* $\mathcal{O}$ in $\mathcal{L}$ and $\boldsymbol{q}(\boldsymbol{x},\boldsymbol{t})$ in $\mathcal{Q}$, whether $\mathcal{O}, \mathcal{A} \models_T \boldsymbol{q}(\boldsymbol{a},\boldsymbol{n})$ for any given ABox $\mathcal{A}$ and tuples $\boldsymbol{a}$ and $\boldsymbol{n}$.

The data complexity of OMQ evaluation is closely related to the equivalent rewritability of OMQs into standard query languages. With any temporal ABox $\mathcal{A}$ we associate a $[\min \mathcal{A}, \max \mathcal{A}]$-interpretation

$$\mathcal{I}_\mathcal{A} = \big(([\min \mathcal{A}, \max \mathcal{A}], <), \Delta^{\mathcal{I}_\mathcal{A}}, P_1^{\mathcal{I}_\mathcal{A}}, P_2^{\mathcal{I}_\mathcal{A}}, \dots, A_1^{\mathcal{I}_\mathcal{A}}, A_2^{\mathcal{I}_\mathcal{A}}, \dots, a_1^{\mathcal{I}_\mathcal{A}}, a_2^{\mathcal{I}_\mathcal{A}}, \dots\big),$$

where $\Delta^{\mathcal{I}_\mathcal{A}}$ is the set of individual names in $\mathcal{A}$, $a_i^{\mathcal{I}_\mathcal{A}} = a_i$ for all i, and

$$A_k^{\mathcal{I}_\mathcal{A}} = \{(a_i, n) \mid A_k(a_i, n) \in \mathcal{A}\} \quad \text{and} \quad P_k^{\mathcal{I}_\mathcal{A}} = \{(a_i, a_j, n) \mid P_k(a_i, a_j, n) \in \mathcal{A}\}, \quad \text{for all } k.$$

Now, let $\mathcal{Q}'$ be any query language over $[\min \mathcal{A}, \max \mathcal{A}]$-interpretations, for example, 2-FOL($<$) itself, a fragment of 2-FOL($<$) or even its extension. We say that $(\mathcal{L}, \mathcal{Q})$-OMQs are *$\mathcal{Q}'$-rewritable over T* if, for every OMQ $(\mathcal{O}, \boldsymbol{q}(\boldsymbol{x}, \boldsymbol{t}))$ in $(\mathcal{L}, \mathcal{Q})$, there exists $\boldsymbol{q}'(\boldsymbol{x}, \boldsymbol{t})$ in $\mathcal{Q}'$ such that, for every temporal ABox $\mathcal{A}$ that has a common model with $\mathcal{O}$, the following equivalence holds for all tuples $\boldsymbol{a}$ and $\boldsymbol{n}$ in $\mathcal{A}$ of appropriate length:

$$\mathcal{O}, \mathcal{A} \models_T \boldsymbol{q}(\boldsymbol{a}, \boldsymbol{n}) \qquad \text{iff} \qquad \mathcal{I}_\mathcal{A} \models \boldsymbol{q}'(\boldsymbol{a}, \boldsymbol{n}).$$

If $\mathcal{Q}'$ is 2-FOL($<$) over T, then $\mathcal{I}_\mathcal{A} \models \boldsymbol{q}'(\boldsymbol{a}, \boldsymbol{n})$ is the standard database query evaluation problem for temporal ABoxes and 2-FOL($<$) queries, which is known to be PSPACE-complete for combined complexity; see, e.g., [61]; if, however, one fixes the query and thus considers the data complexity, then this problem is in AC^0, the class of languages computable by bounded-depth polynomial-size circuits with unary NOT-gates and unbounded fan-in AND- and OR-gates.

Of course, the OMQ evaluation problem for the full 2-FOL($<$) is undecidable, and it is one of the main problems of temporal ontology-mediated query answering to design useful ontology and query languages for which the query evaluation problem is decidable or, even better, feasible in practice. The latter requirement is typically interpreted as being at least in PTIME in data complexity, but to query very large data and employ existing query engines PTIME query evaluation often is not sufficient, and one aims at rewritability into first-order logic (AC^0 data complexity). In the following two sections, we discuss a few known approaches to this problem.

3 Queries Mediated by Domain- or Time-centric Ontologies

An important way of obtaining temporal ontology languages from 2-FOL is simply omitting (non-trivial) quantification over one of its two sorts. Thus, intuitively, if we disallow all but a single outermost universal quantifier for a temporal variable, then we obtain a 'domain-centric' ontology language, in which one can define a time-independent model of the domain; and if we disallow all but a sequence of outermost universal quantifiers for domain variables, then we obtain a 'time-centric' ontology language using which one can define a propositional temporal model. Both approaches have been investigated, and, in the rest of the section, we shall provide a summary of the obtained results. If a model representing both temporal and domain knowledge is needed, we have to carefully define the interaction between the domain and time quantifiers.

3.1 Domain-Centric Ontology Languages

It is straightforward to restrict 2-FOL($<$) in such a way that it only defines non-temporal properties: such an ontology would consist of 2-FOL($<$) sentences $\forall t\, \varphi(t)$, where φ does not contain quantifiers over temporal variables. Of course, query evaluation is still undecidable,

and so further restrictions of its expressive power are needed. In standard, non-temporal, ontology-mediated query answering, description logics are the most popular fragments of first-order logic used to define ontologies. Here, we introduce three basic families of DLs that have been important in the context of OMQ answering, and from which many others can be derived in a straightforward way. Namely, we introduce the basic expressive DL $\mathcal{ALC}$ [19] and the lightweight DLs *DL-Lite* [37, 6] and $\mathcal{EL}$ [18]. In $\mathcal{ALC}$, *concepts* C are constructed using the grammar

$$C \quad ::= \quad \top \quad | \quad A_k \quad | \quad \neg C \quad | \quad C_1 \sqcap C_2 \quad | \quad \exists P_k.C.$$

An $\mathcal{ALC}$ *TBox* (*ontology*) is a finite set of *concept inclusions* $C_1 \sqsubseteq C_2$, where C_1 and C_2 are $\mathcal{ALC}$ concepts. Concepts in the fragment $\mathcal{EL}$ of $\mathcal{ALC}$ are $\mathcal{ALC}$ concepts without occurrences of negation $\neg$. An $\mathcal{EL}$ TBox is a finite set of concept inclusions $C_1 \sqsubseteq C_2$, where C_1 and C_2 are $\mathcal{EL}$ concepts. In *DL-Lite*, *basic concepts* B and *roles* R are constructed using the grammar

$$\begin{aligned} B \quad &::= \quad \top \quad | \quad A_k \quad | \quad \exists R.\top, \\ R \quad &::= \quad P_k \quad | \quad P_k^-. \end{aligned}$$

A *DL-Lite*$^{\mathcal{H}}_{core}$ TBox is a finite set of concept and role inclusions of the form

$$\begin{aligned} &B_1 \sqsubseteq B_2, && B_1 \sqcap B_2 \sqsubseteq \bot, \\ &R_1 \sqsubseteq R_2, && R_1 \sqcap R_2 \sqsubseteq \bot, \end{aligned}$$

where B_1 and B_2 are basic concepts and R_1 and R_2 are roles. Concept and role inclusions of the second type are also called *disjointness axioms*. In *DL-Lite*$^{\mathcal{H}}_{horn}$, one can also form intersections of basic concepts:

$$B_1 \sqcap \cdots \sqcap B_k \sqsubseteq B, \qquad\qquad B_1 \sqcap \cdots \sqcap B_k \sqsubseteq \bot;$$

in *DL-Lite*$^{\mathcal{H}}_{krom}$, one can use negation (but still any concept inclusion contains only two concepts): that is, concept inclusions are of the form $B_1 \sqsubseteq B_2$, $B_1 \sqcap B_2 \sqsubseteq \bot$ and $\top \sqsubseteq B_1 \sqcup B_2$; finally, in *DL-Lite*$^{\mathcal{H}}_{bool}$ one can use both conjunction and negation resulting in concept inclusions of the form $D_1 \sqsubseteq D_2$, where the D_i are defined using the rule

$$D \quad ::= \quad B \quad | \quad \neg D \quad | \quad D_1 \sqcap D_2.$$

(We will assume without loss of generality that the concept inclusions in *DL-Lite*$^{\mathcal{H}}_{bool}$ are given in normal form: $B_1 \sqcap \cdots \sqcap B_k \sqsubseteq B'_1 \sqcup \cdots \sqcup B'_n$; as usual, we assume that the empty union is $\bot$ and the empty intersection is $\top$). All of the above languages in the DL-Lite family contain role inclusions, and the fragment of *DL-Lite*$^{\mathcal{H}}_{\boldsymbol{c}}$ without role inclusions is denoted by *DL-Lite*$_{\boldsymbol{c}}$. Two concept (or role) inclusions $C_1 \sqsubseteq C_2$ and $C_2 \sqsubseteq C_1$ are often abbreviated as $C_1 \equiv C_2$ and are called a concept (respectively, role) *equivalence axiom*.

The DLs introduced above can be regarded as fragments of first-order logic (with a single sort). In the temporal setting, every (basic) concept C can be translated (using the so-called *standard translation*) to a 2-FOL($<$)-formula $C^\sharp(x,t)$ with one free domain variable x and a single temporal variable t, and every role R to a 2-FOL($<$)-formula $R^\sharp(x,y,t)$ with two domain variables x, y and a single temporal variable t:

$$\begin{aligned} (A_k)^\sharp(x,t) &= A_k(x,t), & (\neg C)^\sharp(x,t) &= \neg C^\sharp(x,t), \\ (P_k)^\sharp(x,y,t) &= P_k(x,y,t), & (C_1 \sqcap C_2)^\sharp(x,t) &= C_1^\sharp(x,t) \wedge C_2^\sharp(x,t), \\ (P_k^-)^\sharp(x,y,t) &= P_k(y,x,t), & (\exists R.C)^\sharp(x,t) &= \exists y \left(R^\sharp(x,y,t) \wedge C^\sharp(y,t)\right). \end{aligned}$$

Table 1 Complexity of the satisfiability problem over $\mathbb{N}$.

language	combined complexity		
	no rigid symbols	rigid concepts only	rigid roles & concepts
$\mathcal{ALC}$-LTL [20]	ExpTime	NExpTime	2ExpTime
—"— global GCIs [20]	ExpTime	ExpTime	2ExpTime
$\mathcal{EL}$-LTL [28]	PSpace	NExpTime	NExpTime
—"— global GCIs [28]	PSpace	PSpace	PSpace

Every concept inclusion $C_1 \sqsubseteq C_2$ is then translated as $\forall t \forall x \left(C_1^\sharp(x,t) \to C_2^\sharp(x,t)\right)$ and every role inclusion $R_1 \sqsubseteq R_2$ as $\forall t \forall x \forall y \left(R_1^\sharp(x,y,t) \to R_2^\sharp(x,y,t)\right)$. Thus, we obtain a first important type of temporal ontologies by demanding that its (essentially atemporal) concept and role inclusions hold true at *every time point*. More formally, given a T-interpretation $\mathcal{I}$ and $n \in T$, we can define an *n-slice* $\mathcal{I}(n)$ of $\mathcal{I}$ by taking the standard Tarski-style interpretation for the respective DL:

$$\mathcal{I}(n) \;=\; \big(\Delta^{\mathcal{I}}, P_1^{\mathcal{I}(n)}, P_2^{\mathcal{I}(n)}, \dots, A_1^{\mathcal{I}(n)}, A_2^{\mathcal{I}(n)}, \dots, a_1^{\mathcal{I}(n)}, a_2^{\mathcal{I}(n)}, \dots\big),$$

where $a_i^{\mathcal{I}(n)} = a_i^{\mathcal{I}}$, for all i, and

$$P_k^{\mathcal{I}(n)} = \{(u,v) \mid (u,v,n) \in P_k^{\mathcal{I}}\} \quad \text{and} \quad A_k^{\mathcal{I}(n)} = \{u \mid (u,n) \in A_k^{\mathcal{I}}\}, \quad \text{for all } k.$$

It follows that $\mathcal{I}$ is a T-model of an ontology $\mathcal{O}$ iff each of the concept and role inclusions in the ontology is satisfied in each of the slices $\mathcal{I}(n)$, for $n \in T$.

▶ **Example 3.** In a domain-centric ontology language in the context of Example 1, we can express (1) by using a concept disjointness axiom and (5) using a concept equivalence axiom:

$$\exists \mathsf{publishedIn}.\top \sqcap \exists \mathsf{acceptedIn}.\top \sqsubseteq \bot, \tag{1'}$$

$$\mathsf{Published} \equiv \exists \mathsf{publishedIn}.\top. \tag{5'}$$

It is to be noted that, in the languages just introduced, one cannot represent or reason about any dependencies between the interpretations $\mathcal{I}(n)$ and $\mathcal{I}(m)$ for distinct time points n and m. Examples of such dependencies are sentences (2)–(4) in Example 1. The extension of any ontology language $\mathcal{L}$ with the option to say that a concept name A is time-independent (that is, $A^{\mathcal{I}(n)} = A^{\mathcal{I}(m)}$ for all time points n, m) is called $\mathcal{L}$ *with rigid concepts*. The extension of $\mathcal{L}$ with *rigid roles* is defined analogously. In Example 1, $\mathsf{authorOf}$ could be a rigid role. Of course, if the language has rigid roles, then rigid concepts can be 'simulated' by considering domains of rigid roles: if role R is rigid, then the equivalence axiom $C \equiv \exists R$ ensures that concept C is also rigid.

Baader et al. [20] proposed domain-centric languages. They introduced *$\mathcal{ALC}$-LTL* as the language of $\mathcal{ALC}$ axioms (concept inclusions, or GCI as they are often called in description logic) and ABox assertions with Boolean connectives and temporal operators applied to them. For example, the temporalised axiom $\Diamond_F \Box_F (\mathsf{USCitizen} \sqsubseteq \exists \mathsf{insuredBy}.\mathsf{Insurer})$ says that there is a future time point, from which on every US citizen will always have a health insurance. It turns out that without rigid symbols the two components—the domain and the time—have very little interaction, and so in order to check whether a given formula φ in *$\mathcal{ALC}$-LTL* is satisfiable, one can check whether (1) the *propositional abstraction* of φ (the result of

replacing DL axioms with propositional variables) is satisfiable and (2) the satisfying model yields consistent sets of DL axioms. As a result, the complexity is usually the maximum of the complexities of the two components; see Table 1. Rigid concepts and/or rigid roles make the interaction stronger and require additional global guessing/bookkeeping but the propositional abstraction technique is still applicable. The second set of results in Table 1 refers to the fragment of $\mathcal{ALC}$-*LTL* in which Boolean connectives and temporal operators can be applied only to ABox assertions but $\mathcal{ALC}$ axioms hold globally in all models (in precisely the same way as we defined in the standard translation above). Such a restriction dramatically reduces the complexity for the logic $\mathcal{EL}$-*LTL* [28].

Note also that the Semantic Web community has developed a variety of extensions of RDF/S and OWL with validity time [64, 74, 48]. The focus of this direction of research is on representing and querying timestamped RDF triples or OWL axioms.

3.2 Time-Centric OMQs

One of the main differences between description logics and first-order logic is that the former do not use individual variables. Instead, description logic constructors such as existential restrictions express certain quantifier patterns. The situation in reasoning about time is similar: instead of representing explicitly the temporal precedence relation $<$ using individual variables, one employs temporal operators encoding certain natural language patterns. A very well studied language based on temporal operators is *linear-time temporal logic* (*LTL*) [72]. In contrast to the description logics introduced above, which are much weaker than first-order logic, *LTL* with operators $\mathcal{S}$ ('since') and $\mathcal{U}$ ('until') has exactly the same expressive power as the corresponding *time-fragment* of 2-FOL($<$) (Kamp's theorem); see, e.g., [44, 75]. We now introduce the *LTL* extensions of the concept and role grammars defined above:

$$\begin{array}{lcl} D & ::= & C \mid \bigcirc_P D \mid \bigcirc_F D \mid D_1 \,\mathcal{S}\, D_2 \mid D_1 \,\mathcal{U}\, D_2, \\ S & ::= & R \mid \bigcirc_P S \mid \bigcirc_F S \mid S_1 \,\mathcal{S}\, S_2 \mid S_1 \,\mathcal{U}\, D_2. \end{array}$$

We will also use common abbreviations: for example, $\Diamond_P D = \top \,\mathcal{S}\, D$ ('sometime in the past' for concepts) and $\Box_F S = \neg(\top \,\mathcal{U}\, \neg S)$ ('always in the future' for roles). The standard translation of concepts can be extended to temporalised concepts as follows:

$$\begin{aligned} (\bigcirc_P D)^\sharp(x,t) &= D^\sharp(x,t-1), \\ (\bigcirc_F D)^\sharp(x,t) &= D^\sharp(x,t+1), \\ (D_1 \,\mathcal{S}\, D_2)^\sharp(x,t) &= \exists t_1 \big((t_1 < t) \;\wedge\; D_2^\sharp(x,t_1) \;\wedge\; \forall t_2 \big((t_1 < t_2) \wedge (t_2 < t) \to D_1^\sharp(x,t_2)\big)\big), \\ (D_1 \,\mathcal{U}\, D_2)^\sharp(x,t) &= \exists t_1 \big((t < t_1) \;\wedge\; D_2^\sharp(x,t_1) \;\wedge\; \forall t_2 \big((t < t_2) \wedge (t_2 < t_1) \to D_1^\sharp(x,t_2)\big)\big). \end{aligned}$$

Temporalised roles are translated into 2-FOL($<$) similarly (but two domain variables are used rather than one). Recall that $(t-1)$ is a shortcut for t' that satisfies the condition $(t' < t) \wedge \neg\exists t'' \big((t' < t'') \wedge (t'' < t)\big)$. So, under the strict interpretation of $\mathcal{U}$ and $\mathcal{S}$, the temporal operators $\bigcirc_F$ ('next time') and $\bigcirc_P$ ('previous time') could be equivalently defined as $\bigcirc_F D = \bot \,\mathcal{U}\, D$ and $\bigcirc_P D = \bot \,\mathcal{S}\, D$, respectively.

▶ **Example 4.** In the context of Example 1, we can represent the 'concept' analogues of sentences (1)– (4) as follows:

$$\mathsf{Published} \sqcap \mathsf{Accepted} \sqsubseteq \bot, \tag{1''}$$

$$\mathsf{Published} \sqsubseteq \Box_F \mathsf{Published}, \tag{2''}$$

$$\mathsf{Published} \sqsubseteq \Diamond_P(\mathsf{Accepted} \sqcap \bigcirc_F \mathsf{Published}), \tag{3''}$$

$$\Diamond_P \mathsf{Accepted} \sqcap \Diamond_F \mathsf{Accepted} \sqsubseteq \mathsf{Accepted}. \tag{4''}$$

Rigid concepts and roles can be defined in the language introduced above be using inclusions of the form $C \sqsubseteq \Box_F \Box_P C$ or $C \equiv \bigcirc_F C$.

If no relational knowledge is needed for the domain and the focus is on temporal aspects (as in Example 4), then it suffices to work with ontologies that represent the behaviour of individual domain elements without formalising any interaction between them. So, in the remainder of Section 3.2, we concentrate on the concept-only ontology languages and, in Section 4, we show how these results can be extended to the full setting (under certain restrictions).

In order to present the fine-grained analysis of the complexity of OMQ evaluation, we assume that our ontologies are given in a certain normal form. More precisely, it is known that any *LTL* formula can be transformed into a polynomial-size *LTL* formula in *separated normal form* (SNF) [42] that has the same models (if restricted to the original vocabulary—the transformation requires introduction of auxiliary names, but the result is a conservative extension, which preserves the models restricted to the original vocabulary). The formulas in SNF are conjunctions of global and initial *temporal clauses* that only use the operators $\bigcirc_P$, $\bigcirc_F$, $\Box_P$ and $\Box_F$. So, we consider the *time-centric* ontology language $\mathit{LTL}_{bool}^{\Box\bigcirc}$ with concept inclusions of the form

$$L_1 \sqcap \dots \sqcap L_k \sqsubseteq L'_1 \sqcup \dots \sqcup L'_n,$$

where the L_i and L'_i are concept names possibly prefixed by unary temporal operators $\bigcirc_P$, $\bigcirc_F$, $\Box_P$ or $\Box_F$. We also define the *core*, *krom* and *horn* fragments of $\mathit{LTL}_{bool}^{\Box\bigcirc}$, where the temporal clauses are restricted to

$$\begin{array}{lll} L_1 \sqsubseteq L_2, & L_1 \sqcap L_2 \sqsubseteq \bot, & (\mathit{core}) \\ L_1 \sqsubseteq L_2, & L_1 \sqcap L_2 \sqsubseteq \bot, \quad \top \sqsubseteq L_1 \sqcup L_2, & (\mathit{krom}) \\ L_1 \sqcap \dots \sqcap L_k \sqsubseteq L, & L_1 \sqcap \dots \sqcap L_k \sqsubseteq \bot, & (\mathit{horn}) \end{array}$$

respectively, and the $\bigcirc$- and $\Box$-fragments $\mathit{LTL}_{\boldsymbol{c}}^{\bigcirc}$ and $\mathit{LTL}_{\boldsymbol{c}}^{\Box}$, where only $\bigcirc_P/\bigcirc_F$ and $\Box_P/\Box_F$ operators can be applied. It can be seen that the sub-Boolean fragments of $\mathit{LTL}_{bool}^{\Box\bigcirc}$ are in fact the concept-only counterparts of the respective fragments of *DL-Lite*. Recall that the satisfiability problem is PSPACE-complete for $\mathit{LTL}_{bool}^{\Box\bigcirc}$- and $\mathit{LTL}_{horn}^{\Box\bigcirc}$-formulas, NP-complete for $\mathit{LTL}_{krom}^{\Box\bigcirc}$- and $\mathit{LTL}_{krom}^{\Box}$-formulas, and NLOGSPACE-complete for $\mathit{LTL}_{core}^{\bigcirc}$-, $\mathit{LTL}_{krom}^{\bigcirc}$- and $\mathit{LTL}_{core}^{\Box}$-formulas [9].

Let $\mathcal{O}$ be an ontology in a time-centric language. An *atomic LTL-OMQ* is a pair of the form $(\mathcal{O}, A_k(x,t))$, where A_k is a concept name. We also consider a larger class of OMQs based on *positive temporal concepts* $\varkappa$, which are defined by the following grammar:

$$\begin{array}{rcl} \varkappa & ::= & \bot \mid \top \mid A_k \mid \varkappa_1 \sqcap \varkappa_2 \mid \varkappa_1 \sqcup \varkappa_2 \mid \\ & & \Box_P \varkappa \mid \Box_F \varkappa \mid \varkappa_1 \,\mathcal{S}\, \varkappa_2 \mid \varkappa_1 \,\mathcal{U}\, \varkappa_2. \end{array}$$

Observe that operators $\bigcirc_P$, $\bigcirc_F$, $\Diamond_P$ and $\Diamond_F$ can be used in positive temporal concepts as abbreviations. A *positive LTL-OMQ* is a pair of the form $(\mathcal{O}, \varkappa(x,t))$. It is to be noted that, unlike the ontology language, where we used a normal form, one cannot eliminate the binary temporal operators $\mathcal{S}$ and $\mathcal{U}$ (and the 'sometime in the past/future' operators $\Diamond_P/\Diamond_F$).

In the context of *LTL*-OMQs, two types of $\mathcal{Q}'$-rewritability are of interest for the target language $\mathcal{Q}'$: 2-FOL($<$) and 2-FOL($<,+$). The second language extends 2-FOL($<$) with the ternary numeric predicate PLUS that is interpreted in the two-sorted structure $\mathcal{I}_\mathcal{A}$ (defined in Section 2) as follows:

$$\mathcal{I}_\mathcal{A} \models \mathrm{PLUS}(n, n_1, n_2) \quad \text{iff} \quad n = n_1 + n_2, \quad \text{for } n, n_1, n_2 \in [\min \mathcal{A}, \max \mathcal{A}].$$

Table 2 Data complexity and rewritability of *LTL*-OMQs over $\mathbb{Z}$.

	atomic			positive		
$\boldsymbol{c}$	$\mathit{LTL}_{\boldsymbol{c}}^{\Box}$	$\mathit{LTL}_{\boldsymbol{c}}^{\bigcirc}$	$\mathit{LTL}_{\boldsymbol{c}}^{\Box\bigcirc}$	$\mathit{LTL}_{\boldsymbol{c}}^{\Box}$	$\mathit{LTL}_{\boldsymbol{c}}^{\bigcirc}$	$\mathit{LTL}_{\boldsymbol{c}}^{\Box\bigcirc}$
bool	2-FOL(<)	MSO(<)		MSO(<)	MSO(<)	
krom		2-FOL(<, +)	MSO(<)*			
horn		MSO(<)		2-FOL(<)		
core		2-FOL(<, +)	MSO(<)*		2-FOL(<, +)	MSO(<)*

*It is still open whether these can be improved to 2-FOL(<, +); all other results in the table are optimal: in particular, MSO(<) means NC^1-hardness for data complexity and so, no 2-FOL(<, +)-rewritability.

Observe that even though we can express terms such as $t+n$, for a fixed $n \in \mathbb{Z}$, in 2-FOL(<), terms of the form $t+s$ are not expressible in 2-FOL(<). Evaluation of 2-FOL(<, +)-formulas is known to be in LogTime-uniform AC^0 for data complexity [56] (recall that AC^0 is the class of languages computable by bounded-depth polynomial-size circuits with unary NOT-gates and unbounded fan-in AND- and OR-gates).

▶ **Example 5.** Let $\mathcal{O}$ be an ontology with the following two axioms:

$$\bigcirc_{\!F} A \sqsubseteq B, \qquad \bigcirc_{\!F} B \sqsubseteq A. \tag{9}$$

Consider the positive *LTL*-OMQ $\boldsymbol{Q}(x,t) = (\mathcal{O}, \bigcirc_{\!F} \bigcirc_{\!F} B(x,t))$ and ABox $\mathcal{A} = \{A(a,0), C(a,1)\}$. We have $(a^{\mathcal{I}}, 2n+1) \in B^{\mathcal{I}}$, for any $n \geq 0$ and any $\mathbb{N}$- or $\mathbb{Z}$-model $\mathcal{I}$ of $\mathcal{O}$ and $\mathcal{A}$. It follows that $(a,1)$ is the only certain answer to $\boldsymbol{Q}(x,t)$ because only 1 of all odd numbers is within the interval between $\min \mathcal{A}$ and $\max \mathcal{A}$ (note, however, that the relevant B is true at moment 3).

▶ **Example 6.** Consider now the atomic *LTL*-OMQ $\boldsymbol{Q}(x,t) = (\mathcal{O}, A(x,t))$ with the same $\mathcal{O}$ defined by (9). It is not hard to see that $(a,n) \in A^{\mathcal{I}}$ for any $\mathbb{Z}$-model $\mathcal{I}$ of $\mathcal{O}$ and a given temporal ABox $\mathcal{A}$ iff t is either at an even distance $t-s$ from some $A(a,s) \in \mathcal{A}$ or at an odd distance $t-s$ from some $B(a,s) \in \mathcal{A}$. Thus, the following formula

$$\exists s, n, k, k' \, [\big(A(x,s) \wedge \textsc{plus}(k,n,n) \wedge \textsc{plus}(t,s,k)\big) \vee \big(B(x,s) \wedge \textsc{plus}(k,n,n) \wedge \textsc{plus}(k',k,1) \wedge \textsc{plus}(t,s,k')\big)]$$

is a 2-FOL(<, +)-rewriting of $\boldsymbol{Q}(x,t)$, where, for example, $\textsc{plus}(k,n,n)$ means $k = 2n$. Note that s, n, k and all other quantified variables range between $0 = \min \mathcal{A}$ and $\max \mathcal{A}$ in any $\mathcal{I}_{\mathcal{A}}$; in particular, $t \geq s$. Finally, observe that $\boldsymbol{Q}(x,t)$ is *not* 2-FOL(<)-rewritable since properties such as 't is even' are not definable by 2-FOL(<)-formulas [61].

▶ **Example 7.** Next, instead of just checking whether the distance is even or odd, we devise an ontology that checks whether the number of certain symbols in a given interval is even or odd. More precisely, consider the atomic *LTL*-OMQ $\boldsymbol{Q}(x,t) = (\mathcal{O}, B_0(x,t))$, where $\mathcal{O}$ consists of concept inclusions

$$\bigcirc_{\!F} B_k \sqcap A_0 \sqsubseteq B_k \qquad \text{and} \qquad \bigcirc_{\!F} B_k \sqcap A_1 \sqsubseteq B_{1-k}, \qquad \text{for } \ k = 0, 1.$$

Informally, each occurrence of A_0 in the ABox keeps the same subscript k in B_k and each occurrence of A_1 flips the subscript over by replacing B_0 with B_1 and the other way round. So, for any word $\boldsymbol{e} = (e_0, \dots, e_{n-1}) \in \{0,1\}^n$, let $\mathcal{A}_{\boldsymbol{e}} = \{ B_0(a,n) \} \cup \{ A_{e_i}(a,i) \mid 0 \leq i < n \}$.

It is not hard to check that $(a, 0)$ is a certain answer to $\boldsymbol{Q}(x, t)$ over $\mathcal{A}_{\boldsymbol{e}}$ iff the number of 1s in $\boldsymbol{e}$ is even (PARITY). As PARITY is not in AC^0 [43], $\boldsymbol{Q}(x, t)$ is not 2-FOL-rewritable even if *arbitrary* numeric predicates (not only PLUS) are allowed in rewritings.

On the other hand, PARITY is a regular language, and so belongs to $\mathrm{NC}^1 \supsetneq \mathrm{AC}^0$, the class of languages recognisable by logarithmic-depth circuits with unary NOT-gates and fan-in two AND- and OR-gates. Recall also that (*i*) regular languages coincide with those definable by *monadic second-order* (MSO) formulas built from atoms of the form $A(t)$ and $t < t'$ using the Booleans, first-order quantifiers $\forall t$ and $\exists t$, and second-order quantifiers $\forall A$ and $\exists A$ [35], and that (*ii*) MSO($<$)-formulas can encode the semantics of propositional temporal logic see, e.g., [46]. Thus, all $\mathit{LTL}_{\mathit{bool}}^{\Box\bigcirc}$ OMQs are MSO($<$)-rewritable, and so answering such OMQs is in NC^1 for data complexity.[1] On the other hand, in many cases we can construct 2-FOL($<, +$)- or even 2-FOL($<$)-rewritings; for details on the results, see Table 2 [8].

3.3 Query Answering with Domain-Centric Ontologies

Even without a temporal dimension, first-order logic is too expressive for effective ontology-mediated query answering. Instead, research has been focussed on ontologies in description logic and on queries in small fragments of FO. Most popular are *conjunctive queries* (CQs), defined by the grammar

$$\varphi \quad ::= \quad A_k(x) \quad | \quad P_k(x_1, x_2) \quad | \quad \varphi_1 \wedge \varphi_2 \quad | \quad \exists x\, \varphi,$$

and disjunctions of CQs called *unions of conjunctive queries* (UCQs). Thus, CQs are (equivalent to) conjunctions of atoms in which some variables are existentially quantified. We briefly discuss basic results on the rewritability and data complexity of ontology-mediated queries in the atemporal classes of OMQs $(\mathit{DL\text{-}Lite}_{\mathit{horn}}^{\mathcal{H}}, \mathrm{CQ})$, $(\mathcal{EL}, \mathrm{CQ})$, and $(\mathcal{ALC}, \mathrm{CQ})$. We assume for simplicity that ABoxes consist of facts without time stamps and interpret the description logics in the corresponding single sorted interpretations. Then OMQs in $(\mathit{DL\text{-}Lite}_{\mathit{horn}}^{\mathcal{H}}, \mathrm{CQ})$ are always rewritable into UCQs, a fact which was the main motivation for the introduction of the *DL-Lite* family [37, 6].

▶ **Example 8.** For $\mathcal{O} = \{\, \exists \mathsf{publishedIn}.\top \sqsubseteq \mathsf{Published} \,\}$ and the CQ $\boldsymbol{q}(x) = \mathsf{Published}(x)$, a rewriting of $(\mathcal{O}, \boldsymbol{q}(x))$ is given by $\boldsymbol{q}'(x) = \exists y\, \mathsf{publishedIn}(x, y) \vee \mathsf{Published}(x)$. Intuitively, $\boldsymbol{q}'(x)$ is the disjunction over all possible 'reasons' (according to $\mathcal{O}$) for x to be published.

It follows that answering OMQs from $(\mathit{DL\text{-}Lite}_{\mathit{horn}}^{\mathcal{H}}, \mathrm{CQ})$ is in AC^0 fot data complexity. In contrast, not all OMQs in $(\mathcal{EL}, \mathrm{CQ})$ are rewritable into UCQs or even first-order logic.

▶ **Example 9.** For $\mathcal{O} = \{\, \exists \mathsf{refersTo}.\mathsf{Publication} \sqsubseteq \mathsf{Publication} \,\}$ and CQ $\boldsymbol{q}(x) = \mathsf{Publication}(x)$, one can readily see that $\mathcal{O}, \mathcal{A} \models \boldsymbol{q}(a)$ iff there is a path from a to some individual b such that $\mathsf{Publication}(b) \in \mathcal{A}$ along the $\mathsf{refersTo}$-relation in $\mathcal{A}$. Since reachability cannot be expressed in first-order logic, there is no rewriting of $(\mathcal{O}, \boldsymbol{q}(x))$ in first-order logic.

It can be shown, however, that every OMQ in $(\mathcal{EL}, \mathrm{CQ})$ is rewritable into a datalog program, and so CQ evaluation is in PTIME for data complexity [78]. For OMQs in $(\mathcal{ALC}, \mathrm{CQ})$, the situation is even worse: in this case, OMQ answering can be CONP-hard for data complexity [55]. Bienvenu et al. [24] give a partial classification of OMQs in $(\mathcal{ALC}, \mathrm{CQ})$ into those in PTIME and those that are CONP-hard for data complexity.

[1] By NC^1 we mean the uniform NC^1, which coincides with ALOGTIME.

We now return to querying temporal data under the assumption that the ontology is domain-centric. Querying in this framework has mainly been investigated in the context of ontology-based monitoring of dynamic systems [15, 13, 17]. Suppose that timestamped data is collected while monitoring a system. The data collected at each time point n forms a sequence $(\mathcal{A}(i))_{0\leq i\leq n}$ of ABoxes $\mathcal{A}(i)$ such that every $\mathcal{A}(i)$ contains assertions of the form $A_k(a,i)$ and $P_k(a,b,i)$. Thus, the temporal ABox $\mathcal{A}$ is given as $\mathcal{A}=\bigcup_{0\leq i\leq n}\mathcal{A}(i)$, where $\min\mathcal{A}=0$ and $\max\mathcal{A}=n$ is the current time point. Ontology-mediated queries are used to detect whether an event of interest has occured in $\mathcal{A}$ up to the time point n. The following example illustrates this scenario.

▶ **Example 10.** Suppose that a temporal ABox $\mathcal{A}$ maintained by a journal editor contains data about the submission, reviewing, acceptance and publication of articles. Thus, similarly to the temporal ABox introduced above it contains assertions stating whether an article is under submission, has been accepted, has been published, and so on. A monitoring query of interest might be query (8) from Example 2: find the authors of papers that were submitted more than two years ago but have not been accepted yet.

To query temporal data under domain-centric ontologies, CQs have been extended to temporalised CQs in which LTL operators can be applied to CQs. Thus, queries in *LTL*-CQ are defined by the following grammar:

$$\psi \quad ::= \quad \varphi \quad | \quad \neg\psi \quad | \quad \psi_1\wedge\psi_2 \quad | \quad \psi_1\,\mathcal{S}\,\psi_2 \quad | \quad \psi_1\,\mathcal{U}\,\psi_2,$$

where φ is a CQ. Note that disjunction and the temporal connectives $\bigcirc_P$, $\bigcirc_F$, $\Box_P$, $\Box_F$, $\Diamond_P$, and $\Diamond_F$ can be used as abbreviations in *LTL*-CQs. Thus, *LTL*-CQ extends the set of queries in *LTL*-OMQs from Section 3.2 by admitting negation and applying *LTL* connectives to CQs rather than atomic queries; however, negation can only be applied to a formula all of whose free variables are answer variables of the query. Observe that *LTL*-CQs do not contain temporal variables. To evaluate an *LTL*-CQ, the user chooses a time point n for evaluation, typically the last time point of the temporal ABox representing the dynamic system to be monitored. Formally, *LTL*-CQs ψ can be translated into 2-FOL($<$) formulas with a single temporal variable t and any number of domain variables as follows (we only give the translation for CQs, the extension to general *LTL*-CQs is defined in the same way as the extension of $\cdot^\sharp$ to temporalised concepts in Section 3.2):

$$(A_k(x))^\flat = A_k(x,t), \qquad (P_k(x,y))^\flat = P_k(x,y,t),$$
$$(\varphi_1\wedge\varphi_2)^\flat = \varphi_1^\flat\wedge\varphi_2^\flat, \qquad (\exists x\,\varphi)^\flat = \exists x\,\varphi^\flat.$$

Now, given a DL ontology $\mathcal{O}$, a temporal ABox $\mathcal{A}$, a time point $n\in[\min\mathcal{A},\max\mathcal{A}]$, an *LTL*-CQ $\psi(\boldsymbol{x})$, a tuple $\boldsymbol{a}$ in $\mathcal{A}$, and $T\in\{\mathbb{Z},\mathbb{N},[\min\mathcal{A},\max\mathcal{A}]\}$, one is interested in whether $(\boldsymbol{a},n)$ is a certain answer to $(\mathcal{O},\psi^\flat(\boldsymbol{x},t))$ over $\mathcal{A}$ and T, or $\mathcal{O},\mathcal{A}\models_T\psi^\flat(\boldsymbol{a},n)$ in symbols.

▶ **Example 11.** Query (8) cannot be expressed in *LTL*-CQ. Its natural formalisation using temporal operators is the following

$$\boldsymbol{q}(x)=\exists y\big(\mathsf{authorOf}(x,y)\wedge\mathsf{UnderSubmission}(y)\wedge\bigcirc_P^{24}\mathsf{UnderSubmission}(y)\big),$$

but the quantifier $\exists y$ is applied to a temporalised formula which is not allowed in *LTL*-CQ. By regarding y as an answer variable and considering instead

$$\boldsymbol{q}(x,y)=\mathsf{authorOf}(x,y)\wedge\mathsf{UnderSubmission}(y)\wedge\bigcirc_P^{24}\mathsf{UnderSubmission}(y),$$

Table 3 Combined and data complexity of *LTL*-CQ answering with various DLs over $\mathbb{N}$.

DL	combined/data complexity		
	no rigid	rigid concepts	rigid concepts & roles
DL-Lite$^{|\mathcal{H}}_{core|horn}$ [27]	PSpace/NC^1		
DL-Lite$_{krom|bool}$ [27]	ExpTime/coNP	coNExpTime/coNP	2ExpTime/in ExpTime
DL-Lite$^{\mathcal{H}}_{krom|bool}$ [27]	2ExpTime/coNP		2ExpTime/in ExpTime
$\mathcal{EL}$ [28]	PSpace/PTime	PSpace/coNP	coNExpTime/coNP
$\mathcal{ALC}$ [15]	ExpTime/coNP	coNExpTime/coNP	2ExpTime/in ExpTime

one obtains an *LTL*-CQ. Query (6) from Example 2 can be formulated as an *LTL*-CQ as follows:

$$\boldsymbol{q}(x) = \mathsf{Accepted}(x) \land \bigcirc_P \mathsf{UnderSubmission}(x) \land \bigcirc_P^{13} \mathsf{UnderSubmission}(x).$$

Table 3 summarises the known results [27, 28, 15] on the data and combined complexity of *LTL*-CQ evaluation mediated by domain-centric DL ontologies for $\mathbb{N}$-models (thus, for the evaluation problem $\mathcal{O}, \mathcal{A} \models_{\mathbb{N}} \psi^b(\boldsymbol{a}, n)$). It is not difficult to show the same upper bounds for ABox-fitting models, and we conjecture that the same lower bounds hold for ABox-fitting models as well. We also conjecture that the same results hold for $\mathbb{Z}$-models. The proofs generalise the propositional abstraction method employed in the analysis of the complexity of the satisfiability problem for $\mathcal{ALC}$-*LTL* ontologies. In fact, since *LTL*-CQs are closed under negation, the upper bounds in Table 1 can be proved by a straightforward reduction using the upper bounds for combined complexity in Table 3. It is of interest to observe that even for basic *DL-Lite* dialects, and without rigid concepts and roles one does not obtain FO-rewritability (because the problem is NC^1-hard), which is caused by negation in *LTL*-CQs. In contrast, query evaluation for $\mathcal{EL}$ without right concept and roles is still in PTime in data complexity.

The complexity landscape presented in Table 3, has been further extended to more expressive description logics, in particular, containing subroles and transitive roles: the results for those cases are essentially the same as for $\mathcal{ALC}$ [16, 17].

The rewritability properties of OMQs using *LTL*-CQs are investigated by Borgwardt et al. [25, 26], where the focus is on query evaluation for ABox-fitting models. If no negation is present in an *LTL*-CQs $\boldsymbol{q}(\boldsymbol{x})$ and the ontology $\mathcal{O}$ is in *DL-Lite*$_{core}$ without rigid symbols, then *LTL*-CQ rewriting $\boldsymbol{q}'(\boldsymbol{x})$ of $(\mathcal{O}, \boldsymbol{q}(\boldsymbol{x}))$ can be obtained by simply replacing any non-temporal CQ $\varphi(\boldsymbol{x})$ in $\boldsymbol{q}(\boldsymbol{x})$ by the UCQ-rewriting of the non-temporal OMQ $(\mathcal{O}, \varphi(\boldsymbol{x}))$. A general transfer theory is developed [26] with the aim of showing that for a large class of domain-centric ontology languages rewritability (as well as combined rewritability [57]) is preserved under moving from non-temporal queries (such as CQs) to temporalised queries (such as *LTL*-CQs without negation).

Another major concern of research on the use of *LTL*-CQs in monitoring applications is the question whether it is possible to avoid storing the whole sequence $\mathcal{A}_0, \dots, \mathcal{A}_n$ to compute the certain answers to a given *LTL*-CQ at time-point n but instead keeping only a tail $\mathcal{A}_{n-b}, \dots, \mathcal{A}_n$ of the data. A variety of results in this direction have been obtained [26]. One approach is based on a classical separation result stating that, for every *LTL*-formula, there exists an *LTL*-formula without past-operators, which is equivalent to the original formula at the time-point 0 in $\mathbb{N}$-models [44].

Another approach is to add temporal connectives to the query language while keeping a standard atemporal ontology language [68, 69]. In the *streaming data* scenario, the relevant slices of the temporal data (i.e., the finite data history in the form of a sequence of ABoxes to be considered by the query) are specified with a *window* operator using a *sliding* parameter that determines the rate at which snapshots of the data are taken, and a *width* parameter that fixes the size on the window/history. This approach is realised in the Stream-Temporal Query Language STARQL [70]. Soylu et al. [83] have shown how the evaluation of STARQL queries is possible using standard SQL engines and report on the performance.

4 Combinations

In many cases, neither a domain nor a time-centric ontology language suffices, but some combination of them is needed. Designing combinations with good computational properties is notoriously difficult as the two-dimensional structure of temporal data makes it rather straightforward to encode the behaviour of Turing machines for even seemingly inexpressive languages. Thus, straightforward language combinations are often undecidable. In fact, only under rather intricate restrictions decidability is preserved [45, 54]. In this survey, our main concern is not decidability, but much stronger conditions such as tractability of OMQ evaluation and rewritability into 2-FOL. It should thus be clear that the interaction between temporal and DL constructors has to be pretty much restricted to obtain algorithmically well behaved combinations. In this section, we discuss three recent approaches to address this problem.

4.1 A 2-FOL(<)-Rewritable Temporal Extension of DL-Lite

Artale et al. [12] begin with the observation that, for any ontology $\mathcal{O}$, if one wants all OMQs based on $\mathcal{O}$ to be 2-FOL(<)-rewritable, then one has to ensure that $\mathcal{O}$ is materialisable (in the sense that, for any temporal ABox $\mathcal{A}$ consistent with $\mathcal{O}$, there exists a model $\mathcal{I}$ of $\mathcal{A}$ and $\mathcal{O}$ that gives exactly the certain answers to any OMQ with $\mathcal{O}$). Equivalently, one requires that no disjunction of CQs is entailed if none of it disjuncts is entailed. This excludes the use of the temporal operators $\Diamond_F$ and $\Diamond_P$ on the right-hand side of concept inclusions, as illustrated by the following example.

▶ **Example 12.** Let $\mathcal{O} = \{A \sqsubseteq \Diamond_F B\}$. Consider the two-sorted CQs

$$\begin{aligned} \boldsymbol{q}_1(x,t) &= \exists t' \big((t < t') \land C(x,t') \land B(x,t')\big), \\ \boldsymbol{q}_2(x,t) &= \exists t' \big((t < t') \land C(x,t') \land \exists t'' \big((t' < t'') \land B(x,t'')\big)\big). \end{aligned}$$

For $\mathcal{A} = \{A(a,0), C(a,1), D(a,2)\}$, either B occurs together with C (for example, at moment 1), or B occurs after the moment 1, and so $\mathcal{O}, \mathcal{A} \models_{\mathbb{Z}} \boldsymbol{q}_1(a,0) \lor \boldsymbol{q}_2(a,0)$ but $\mathcal{O}, \mathcal{A} \not\models_{\mathbb{Z}} \boldsymbol{q}_i(a,0)$ for $i = 1, 2$. One can use an encoding of 2+2-SAT [79] to show that there is a two-sorted CQ $\boldsymbol{q}$ such that evaluating $(\mathcal{O}, \boldsymbol{q})$ over $\mathbb{Z}$-models is in fact coNP-hard for data complexity (and thus, there is no rewriting).

The following combination of *DL-Lite* and *LTL* is then suggested so that it avoids non-materialisability by not admitting any temporal operators $\Diamond_P$ and $\Diamond_F$ on the right-hand side of concept or role inclusions. *Basic concepts* B, *temporalised concepts* C, *roles* R and *temporalised roles* S are defined by the following grammar:

$$\begin{aligned} B &::= A_i \mid \exists R.\top, & C &::= B \mid C_1 \sqcap C_2 \mid \Diamond_F C \mid \Diamond_P C, \\ R &::= P_i \mid P_i^-, & S &::= R \mid S_1 \sqcap S_2 \mid \Diamond_F S \mid \Diamond_P S; \end{aligned}$$

note that $\exists R.\top$ can only contain a basic role because temporalised roles can contain $\Diamond_P$ and $\Diamond_F$ (which are not allowed to occur on the right-hand side of concept inclusions). The *concept* and *role inclusions* in *DL-Lite*$_{horn}^{\text{lhs}\Diamond}$ are of the form

$$C \sqsubseteq B, \qquad S \sqsubseteq R.$$

A *DL-Lite*$_{horn}^{\text{lhs}\Diamond}$ ontology is a finite set of inclusions in *DL-Lite*$_{horn}^{\text{lhs}\Diamond}$. The following ontology illustrates expressiveness of the language.

▶ **Example 13.** In the context of Example 1, *DL-Lite*$_{horn}^{\text{lhs}\Diamond}$ can represent all 2-FOL(<)-sentences except (3):

$$\exists \mathsf{publishedIn}.\top \sqcap \exists \mathsf{acceptedIn}.\top \sqsubseteq \bot, \tag{1'}$$
$$\Diamond_P \mathsf{publishedIn} \sqsubseteq \mathsf{publishedIn}, \tag{2'}$$
$$\Diamond_P \mathsf{acceptedIn} \sqcap \Diamond_F \mathsf{acceptedIn} \sqsubseteq \mathsf{acceptedIn}. \tag{4'}$$

Note that (2′) and (4′) are role inclusions expressing convexity (also known as existential rigidity) of publishedIn and acceptedIn, respectively. We can also say that authorOf is a rigid role: $\Diamond_P \Diamond_F \mathsf{authorOf} \sqsubseteq \mathsf{authorOf}$.

As the query language we take the obvious extension of single-sorted CQs to two-sorted CQs, 2-CQ(<), defined by the following grammar:

$$\varphi \quad ::= \quad A_k(x,t) \quad | \quad P_k(x_1,x_2,t) \quad | \quad (t_1 < t_2) \quad | \quad (t_1 = t_2) \quad | \quad \varphi_1 \wedge \varphi_2 \quad | \quad \exists x\, \varphi \quad | \quad \exists t\, \varphi.$$

The query language 2-CQ(<) is rather expressive allowing an arbitrary nesting of domain and temporal quantifiers as illustrated by the following example.

▶ **Example 14.** Assuming that authorOf is rigid and using the fact that UnderSubmission is convex, query (8) can now be expressed as follows (cf. Example 11):

$$\boldsymbol{q}(x,t) \;=\; \exists y \big(\mathsf{authorOf}(x,y,t) \wedge \exists t_1 \exists t_2 \ldots \exists t_{24} \big((t_{24} < t_{23}) \wedge \cdots \wedge (t_2 < t_1) \wedge (t_1 < t) \wedge \mathsf{UnderSubmission}(y,t_{24})\big) \wedge \mathsf{UnderSubmission}(y,t)\big),$$

On the other hand, unlike *LTL*-CQ, 2-CQ(<) does not allow the $\bigcirc_P, \bigcirc_F$ operators, and it is not known whether the addition of these operators to 2-CQ(<) will preserve rewritibility.

Using the fact that ontologies in *DL-Lite*$_{horn}^{\text{lhs}\Diamond}$ are materialisable, one can show that OMQs with *DL-Lite*$_{horn}^{\text{lhs}\Diamond}$ ontologies and two-sorted CQs are 2-FOL(<)-rewritable over $\mathbb{Z}$-models.

▶ **Example 15.** A 2-FOL(<)-rewriting for OMQ $(\{(4')\}, \mathsf{acceptedIn}(x,y,t))$ over $\mathbb{Z}$ is

$$\mathsf{acceptedIn}(x,y,t) \vee \big[\exists t' \big((t' < t) \wedge \mathsf{acceptedIn}(x,y,t')\big) \wedge \exists t' \big((t' > t) \wedge \mathsf{acceptedIn}(x,y,t')\big)\big].$$

4.2 Towards a Classification for Temporal DL-Lite

A more systematic investigation into the data complexity and rewritability of OMQs based on temporal *DL-Lite* was launched by Artale et al. [8]; see also [59]. The considered languages are based on the time-centric ontology languages introduced in Section 3.2. Thus, in contrast to *DL-Lite*$_{horn}^{\text{lhs}\Diamond}$, the operators $\Diamond_P$ and $\Diamond_F$ do not occur explicitly in ontologies but, instead,

the basic temporal operators are $\bigcirc_P$, $\bigcirc_F$, $\Box_P$, and $\Box_F$. Now, temporal operators can occur both on the left- and right-hand side of concept and role inclusions. Formally, *basic concepts* B, *temporalised concepts* C, *roles* R and *temporalised roles* S are defined by the following grammar:

$$B ::= A_i \mid \exists R.\top, \qquad C ::= B \mid \Box_F C \mid \Box_P C \mid \bigcirc_F C \mid \bigcirc_P C$$
$$R ::= P_i \mid P_i^-, \qquad S ::= R \mid \Box_F S \mid \Box_P S \mid \bigcirc_F S \mid \bigcirc_P S.$$

Concept and *role inclusions* in normal form are as follows:

$$C_1 \sqcap \dots \sqcap C_k \sqsubseteq C'_1 \sqcup \dots \sqcup C_n \qquad \text{and} \qquad S_1 \sqcap \dots \sqcap S_k \sqsubseteq S'_1 \sqcup \dots \sqcup S'_n.$$

The next example shows how concept and role inclusions in *DL-Lite*$_{horn}^{\mathrm{lhs}\Diamond}$ can be expressed using the operators $\Box_P$ and $\Box_F$.

▶ **Example 16.** Role inclusion (2′) from Example 13 can equivalently be expressed using $\Box_F$:

$$\mathsf{publishedIn} \sqsubseteq \Box_F \mathsf{publishedIn}. \tag{2″}$$

Note that $\Diamond_P$ on the left-hand side is replaced by $\Box_F$ on the right-hand side. To express (4′), however, fresh role names $\mathsf{acceptedInF}$ and $\mathsf{acceptedInP}$ are required, and the following three role inclusions are, in fact, a model conservative extension of (4′):

$$\mathsf{acceptedIn} \sqsubseteq \Box_F \mathsf{acceptedInF}, \tag{4″$_1$}$$
$$\mathsf{acceptedIn} \sqsubseteq \Box_P \mathsf{acceptedInP}, \tag{4″$_2$}$$
$$\mathsf{acceptedInF} \sqcap \mathsf{acceptedInP} \sqsubseteq \mathsf{acceptedIn}. \tag{4″$_3$}$$

It is not difficult to generalise this argument to arbitrary concept and role inclusions in *DL-Lite*$_{horn}^{\mathrm{lhs}\Diamond}$.

We classify ontologies depending on the shape of their inclusions and the temporal operators in them similarly to the fragments of $\mathit{LTL}_{bool}^{\Box\bigcirc}$ in Section 3.2. For $\boldsymbol{c} \in \{\mathit{bool}, \mathit{horn}, \mathit{krom}, \mathit{core}\}$ and $\boldsymbol{o} \in \{\Box, \bigcirc, \Box\bigcirc\}$, we denote by *DL-Lite*$_{\boldsymbol{c}}^{\boldsymbol{o}}$ the ontology language whose (concept and role) inclusions have the shape specified by $\boldsymbol{c}$ (for example, the *core* fragments only contain inclusions and disjointness axioms between temporalised concepts/roles, whereas $\boldsymbol{c} = \mathit{horn}$ allows, in addition, intersection $\sqcap$ to be applied to concepts/roles) and only use the (future and past) operators indicated in $\boldsymbol{o}$ (for example, $\boldsymbol{o} = \Box$ means that only $\Box_F$ and $\Box_P$ can be used).

The main ingredients of the query language are *positive temporal concepts* $\varkappa$ and *positive temporal roles* ϱ given by the grammars

$$\varkappa ::= \top \mid A_k \mid \exists R.\varkappa \mid \varkappa_1 \sqcap \varkappa_2 \mid \varkappa_1 \sqcup \varkappa_2 \mid \boldsymbol{op}_1\, \varkappa \mid \varkappa_1\, \boldsymbol{op}_2\, \varkappa_2,$$
$$\varrho ::= S \mid \varrho_1 \sqcap \varrho_2 \mid \varrho_1 \sqcup \varrho_2 \mid \boldsymbol{op}_1\, \varrho \mid \varrho_1\, \boldsymbol{op}_2\, \varrho_2,$$

where $\boldsymbol{op}_1 \in \{\bigcirc_F, \Diamond_F, \Box_F, \bigcirc_P, \Diamond_P, \Box_P\}$ and $\boldsymbol{op}_2 \in \{\mathcal{U}, \mathcal{S}\}$. Note that we can only use non-temporalised roles in $\exists R.\varkappa$. A *DL-Lite*$_{\boldsymbol{c}}^{\boldsymbol{o}}$ *positive OMQ* is a pair of the form $\boldsymbol{Q}(x,t) = (\mathcal{O}, \varkappa(x,t))$ or $\boldsymbol{Q}(x,y,t) = (\mathcal{O}, \varrho(x,y,t))$, where $\mathcal{O}$ is a *DL-Lite*$_{\boldsymbol{c}}^{\boldsymbol{o}}$ ontology, $\varkappa$ is a positive temporal concept and ϱ a positive temporal role (which can use all temporal operators, not necessarily only those in $\boldsymbol{o}$). If $\varkappa$ and ϱ are concept and role names, we refer to $\boldsymbol{Q}$ as an *atomic OMQ*.

Most of the data complexity and rewitability results reported in Table 4 are obtained by extending the constructions from *LTL*-OMQs. A surprising result here is that answering

Table 4 Data complexity and rewritability of positive OMQs over $\mathbb{Z}$.

	$\textit{DL-Lite}_{\boldsymbol{c}}^{\Box}$	$\textit{DL-Lite}_{\boldsymbol{c}}^{\bigcirc}$	$\textit{DL-Lite}_{\boldsymbol{c}}^{\Box\bigcirc}$
bool and *krom*		coNP-hard	
horn	NC^1-hard	NC^1-hard	
horn with monotone RIs	2-FOL(<)		
core	2-FOL(<)	2-FOL(<, +)	?

positive OMQ with $\textit{DL-Lite}_{horn}^{\Box}$ ontologies turns out to be NC^1-hard (in contrast to $\textit{LTL}_{horn}^{\Box}$, which is 2-FOL(<)-rewritable). The class of $\textit{DL-Lite}_{horn}^{\Box}$ ontologies with *monotone role inclusions* (the precise definition of which is too elaborate for this survey) includes, in particular, all $\textit{DL-Lite}_{horn}^{\Box}$ ontologies whose role inclusions contain no $\Box_P$ and $\Box_F$ operators on the left-hand side. As demonstrated in Example 16, such ontologies are sufficient for encoding the language $\textit{DL-Lite}_{horn}^{\mathrm{lhs}\Diamond}$ from Section 4.1. It is still an open problem whether OMQs with $\textit{DL-Lite}_{core}^{\Box\bigcirc}$ are 2-FOL(<, +)-rewritable or NC^1-hard.

Query (8) from Example 1 is expressible as a positive concept query if we assume that authorOf is a rigid role:

$$\boldsymbol{q}(x,t) \quad = \quad \exists \mathsf{authorOf}.(\mathsf{UnderSubmission} \sqcap \bigcirc_P^{24} \mathsf{UnderSubmission})(x,t)$$

(however, it is not expressible otherwise). In general, the query language of positive temporal concepts and roles is incomparable with 2-CQ(<): the former, for example, allows union $\sqcup$ and $\Box_P/\Box_F$, but the latter contains not necessarily tree-shaped queries. It is still open whether the results of Table 4 hold for 2-CQ(<) queries.

4.3 Temporal EL

The description logic $\mathcal{EL}$ is another tractable language, but since CQ answering in (atemporal) $\mathcal{EL}$ is PTime-complete, a more expressive than 2-FOL(<) target language for rewritings in its temporal extension would be required. One candidate could be $\textsc{Datalog}_{1S}$, a decidable extension of Datalog with one unary *successor* function. Evaluating $\textsc{Datalog}_{1S}$ programs is known to be in ExpTime in combined complexity and PSpace-complete for data complexity [40].

Gutiérrez-Basulto et al. [49] considered a temporal extension $\mathcal{TEL}$ of $\mathcal{EL}$, in which concepts are defined by the following grammar:

$$C \quad ::= \quad A_k \quad | \quad \exists P_k.C \quad | \quad C_1 \sqcap C_2 \quad | \quad \Diamond_P C \quad | \quad \Diamond_F C \quad | \quad \bigcirc_P C \quad | \quad \bigcirc_F C.$$

(Note that $\mathcal{EL}$ has no role inverses, P_k^-.) Ontologies in $\mathcal{TEL}$ are finite sets of concept inclusions of the form $C_1 \sqsubseteq C_2$ (and contain no role inclusions). In terms of expressivity, observe that the 'concept' analogues (2′) and (4′) of sentences (2) and (4) in Example 1 belong to $\mathcal{TEL}$ (note that (1) strictly speaking does not belong to $\mathcal{TEL}$, but such an extension would be straightforward). Rigid concepts are also expressible in $\mathcal{TEL}$, and the language has rigid roles.

As the query language, Gutiérrez-Basulto et al. [49] chose atomic queries of the form $A(x,t)$. We mention, however, that queries (6), (7) and (8) can all be defined as $\mathcal{TEL}$-concepts in the ontology: for example,

$$\exists \mathsf{authorOf}.\big(\mathsf{UnderSubmission} \sqcap \bigcirc_P^{24} \mathsf{UnderSubmission}\big) \sqsubseteq Q, \tag{8′}$$

and then Q could be used as an atomic query.

Answering atomic OMQs in the full $\mathcal{TEL}$ turns out to be undecidable (here and below, all the results over $\mathbb{Z}$-models), but this is essentially due to the $\Diamond_P/\Diamond_F$ operators on the right-hand side of concept inclusions. In the fragment with only $\Diamond_P/\Diamond_F$ and only on the left-hand side of concept inclusions (like the language in Section 4.1), which is similar to the *inflationary* DATALOG$_{1S}$ [39], query answering is PTIME-complete for both data and combined complexity.

The fragment $\mathcal{TEL}^{\bigcirc}$ of $\mathcal{TEL}$ that uses only $\bigcirc_P/\bigcirc_F$ operators can express (as a model conservative extension) all axioms of the inflationary $\mathcal{TEL}$. For example, the concept analogue of (4′) (expressing convexity) can be encoded using two additional concept names and the following concept inclusions:

$$\begin{aligned} &\mathsf{Accepted} \sqsubseteq \bigcirc_F \mathsf{AcceptedInF}, && \mathsf{AcceptedInF} \sqsubseteq \bigcirc_F \mathsf{AcceptedInF},\\ &\mathsf{Accepted} \sqsubseteq \bigcirc_P \mathsf{AcceptedInP}, && \mathsf{AcceptedInP} \sqsubseteq \bigcirc_P \mathsf{AcceptedInP},\\ &\mathsf{AcceptedInP} \sqcap \mathsf{AcceptedInF} \sqsubseteq \mathsf{AcceptedInF}; \end{aligned}$$

see also $(4_1'')$–$(4_3'')$. It is not known whether query answering in the full $\mathcal{TEL}^{\bigcirc}$ is decidable. However, it is PTIME-complete for data and PSPACE-complete for combined complexity in its sublanguage without rigid roles, and PSPACE-complete in data and in EXPTIME for combined complexity in the sublanguage where rigid roles can only occur on the left-hand side of concept inclusions. These results are proved by translating the query answering problem into DATALOG$_{1S}$. Moreover, acyclic $\mathcal{TEL}$-OMQs can be rewtritten into 2-FOL$(<,+)$, and the evaluation problem for such OMQs is in PTIME in combined complexity. Making the ontology acyclic in one of the dimensions only (either time or DL), gives the following results: for temporally acyclic ontologies, which include all atemporal $\mathcal{EL}$ ontologies, it is PTIME-complete in both combined and data complexity; for DL-acyclic ontologies, OMQ answering is non-elementary for combined complexity but NC^1-complete for data complexity.

5 Interval-Based Temporal Ontology-Mediated Query Answering

In the ontology and query languages considered in Sections 2–4, time was assumed to be point-based and discrete. It is well-known, however, that both features may cause difficulties for modelling certain application domains.

We begin with the view of time as intervals, that is, sequences of points. The standard way of storing temporal information in databases is by attaching a validity time *interval* to tuples. For example, a relational table EmployeeSalaries with columns EmployeeID, MonthlySalary, FromTime and ToTime contains tuples such as $(\mathsf{e007}, \mathsf{£3000}, 01/01/2008, 05/01/2014)$. The simplest and most intuitive way of representing this information in the point-based setting is to stipulate that such a tuple is a shorthand for the sequence of tuples

$$(\mathsf{e007}, £3000, 01/01/2008),\ (\mathsf{e007}, £3000, 02/01/2008), \dots, (\mathsf{e007}, £3000, 05/01/2014)$$

provided that it is known *a priori* that a day is the minimal unit of time required in the application. Such a conversion of intervals to points, performed explicitly or implicitly by a query-answering engine, is known to cause an exponential blow-up to the worst-case execution time (since timestamps are encoded in binary, see, e.g., [4]). At the same time, this conversion is not always sound. Consider, for instance, the tuple $(\mathsf{tb007}, 1500, 11{:}25, 11{:}29)$ from a table for a turbine performance monitoring system with columns TurbineID, AverageRotationSpeed, FromTime and ToTime. Clearly, the tuple $(\mathsf{tb007}, 1500, 11{:}27)$ would not make much sense

since 1500 is the *average* rotation speed over the given interval. These examples suggest replacing the point-based setting with an interval-based view of time, where the truth-values of predicates are assigned to time intervals rather than points.

We discuss two interval-based temporal logics and related formalisms for ontology-mediated query answering.

5.1 Halpern-Shoham Interval Temporal Logic

In the interval temporal logic $\mathcal{HS}$ introduced by Halpern and Shoham [53], formulas are interpreted over the set of intervals of any given linear order. More precisely, let $\mathfrak{T} = (T, \leq)$ be a linear order, that is, $\leq$ is a reflexive, transitive, antisymmetric and connected binary relation on T. (As usual, $x < y$ is a shortcut for '$x \leq y$ and $x \neq y$'.) For example, the rationals $(\mathbb{Q}, \leq)$ and reals $(\mathbb{R}, \leq)$ are dense linear orders, while the integers $(\mathbb{Z}, \leq)$ and the natural numbers $(\mathbb{N}, \leq)$ are discrete ones. By an *interval in* $\mathfrak{T}$ we mean any ordered pair $\langle i, j\rangle$ such that $i \leq j$, and denote by $\mathsf{int}(\mathfrak{T})$ the set of all intervals in $\mathfrak{T}$. Note that $\mathsf{int}(\mathfrak{T})$ contains all the *punctual intervals* of the form $\langle i, i\rangle$, which is often referred to as the *non-strict* semantics. Under the *strict* semantics adopted by Allen [2], punctual intervals are disallowed.

Temporal ABoxes in the interval-based paradigm consist of assertions such as

$$A(a, \iota) \quad \text{and} \quad S(a, b, \iota)$$

saying that, respectively, $A(a)$ and $S(a, b)$ hold true at the interval $\iota \in \mathsf{int}(\mathfrak{T})$. For example, an ABox containing timetabling data of a summer school can have the assertions:

$$\mathsf{TutorialDay}(\text{'Semantic Web'}, \langle 07/26/2017\ 08{:}00, 07/26/2005\ 16{:}00\rangle),$$
$$\mathsf{LunchBreak}(\text{'Semantic Web'}, \langle 07/26/2017\ 11{:}30, 07/26/2005\ 12{:}30\rangle).$$

A *de facto* standard way of defining a language expressing statements (constraints) over intervals is by incorporating Allen's [2] interval relations defined as shown in Fig. 2.[2] Since all of these relations are irreflexive, we refer to this definition as the *irreflexive semantics*. As an alternative, the *reflexive semantics* is obtained by replacing each $<$ in Fig. 2 with $\leq$. We write $\mathfrak{T}(\leq)$ or $\mathfrak{T}(<)$ to indicate that the semantics is reflexive or, respectively, irreflexive.

Equipped with Allen's relations, we can express, for example, the query asking for the names of the tutorials that are followed by a lunch break and the times of those lunch breaks:

$$\boldsymbol{q}(x, \chi) \quad = \quad \exists \rho \left(\mathsf{LunchBreak}(x, \chi) \wedge \mathsf{TutorialDay}(x, \rho) \wedge \bar{\mathsf{A}}(\chi, \rho)\right),$$

where χ and ρ are variables ranging over time intervals. Over the ABox with the two statements above, this query would not return any answers because the lunch break is in the middle of the Semantic Web tutorial (D) rather than after it ($\bar{\mathsf{A}}$). In fact, such queries are supported by the SQL:2011 standard [60], which adopts the strict semantics for some of Allen's relations and the non-strict for others.

The Halpern-Shoham interval temporal logic $\mathcal{HS}$ [53] is a propositional modal logic with diamond operators of the form $\langle \mathsf{R}\rangle$ for Allen's interval relations R. The propositional variables of $\mathcal{HS}$ are interpreted by sets of intervals of a given linear order $\mathfrak{T}$ where they are assumed to hold true, and a formula $\langle \mathsf{R}\rangle\varphi$ is true at an interval $\iota \in \mathsf{int}(\mathfrak{T})$ iff φ is true at some interval ι' such that $\iota\, \mathsf{R}\, \iota'$. This semantics can be extended to first-order or description logic in a

[2] It is to be noted that there are two slightly different versions of A and $\bar{\mathsf{A}}$ in the literature.

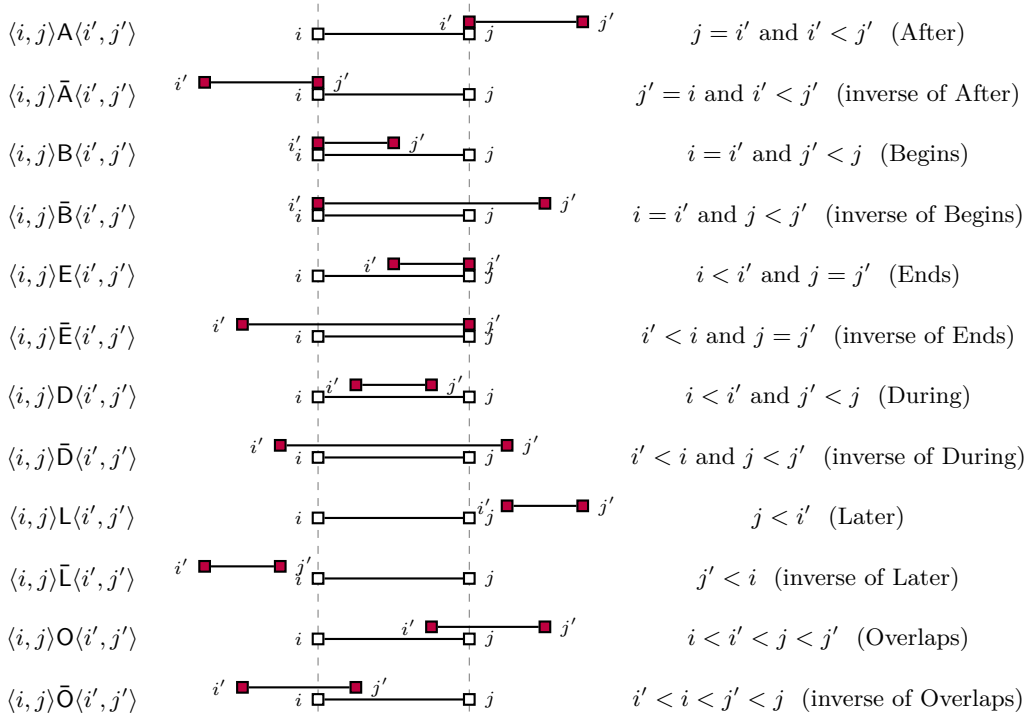

Figure 2 Allen's interval relations under the irreflexive semantics.

natural way. For example, we can give the following definition of 'a morning session' in a DL version of $\mathcal{HS}$:

$$[\mathsf{U}]\big(\langle\bar{\mathsf{B}}\rangle\mathsf{TutorialDay} \sqcap \langle\mathsf{A}\rangle\mathsf{LunchBreak} \sqsubseteq \mathsf{MorningSession}\big), \tag{10}$$

where U is the *universal* relation between intervals, and $[\mathsf{U}]$ means 'at all intervals'. In English, this axiom says that an object d is a $\mathsf{MorningSession}$ in an interval ι—$\mathsf{MorningSession}(d, \iota)$ in symbols—if there is an interval ι' such that $\iota\ \bar{\mathsf{B}}\ \iota'$ and $\mathsf{TutorialDay}(d, \iota')$, and also there is an interval ι'' such that $\iota\ \mathsf{A}\ \iota''$ and $\mathsf{LunchBreak}(d, \iota'')$. The query $\boldsymbol{q}(x, \chi) = \mathsf{MorningSession}(x, \chi)$ mediated by ontology $\{(10)\}$ over the ABox above would return the certain answer

('Semantic Web', ⟨07/26/2017 08:00, 07/26/2017 11:30⟩)

meaning that Semantic Web in the time slot between 8:00 and 11:30 is a morning session.

The elegance and expressive power of $\mathcal{HS}$ have attracted attention of many areas of computer science and AI. However, promising applications have been hampered by the fact, already discovered by Halpern and Shoham [53], that $\mathcal{HS}$ is highly undecidable (for example, validity over $\mathbb{Z}$ and $\mathbb{R}$ is Π^1_1-hard). For recent studies of the computational complexity of reasoning with various fragments of $\mathcal{HS}$, we refer the reader to [34, 33, 63, 1, 32] and references therein.

A tractable fragment of $\mathcal{HS}$ and its DL-Lite and datalog extensions that can be used for temporal ontology-mediated query answering have recently been suggested [11, 58]. We briefly discuss these two formalisms in the remainder of Section 5.

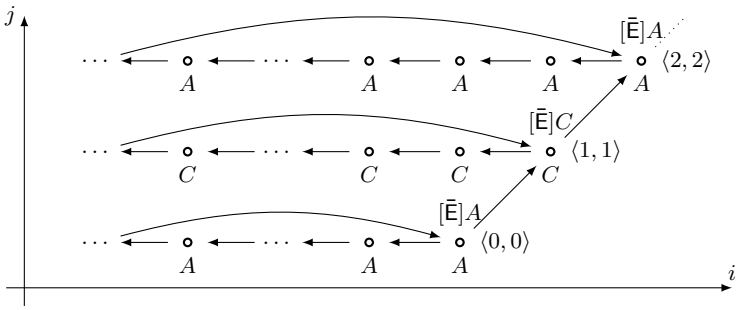

Figure 3 Deriving $A(\iota)$ from $\mathcal{O}$ and $\mathcal{A}$ in Example 17.

5.2 Description logic $\mathcal{HS}$-$Lite^{\mathcal{H}}_{horn}$

The language of $\mathcal{HS}$-$Lite^{\mathcal{H}}_{horn}$ is an extension of DL-$Lite^{\mathcal{H}}_{horn}$ [6]. It contains *individual names* $a_1, a_2, \dots$, *concept names* $A_1, A_2, \dots$, and *role names* $P_1, P_2, \dots$. *Basic roles* R, *basic concepts* B, *temporal roles* S and *temporal concepts* C are given by the following grammar:

$$B ::= \top \mid A_k \mid \exists R.\top, \qquad C ::= B \mid [\mathsf{R}]C \mid \langle \mathsf{R}\rangle C,$$
$$R ::= P_k, \mid P_k^- \qquad S ::= R \mid [\mathsf{R}]S \mid \langle \mathsf{R}\rangle S,$$

where R is one of Allen's interval relations or the universal relation U and $[\mathsf{R}]$ is the dual of $\langle \mathsf{R}\rangle$, that is, $[\mathsf{R}]\varphi$ holds at an interval ι iff φ holds at all intervals ι' such that $\iota \,\mathsf{R}\, \iota'$. An $\mathcal{HS}$-$Lite^{\mathcal{H}}_{horn}$ TBox is a finite set of *concept* and *role inclusions* and *disjointness constraints* of the form

$$C_1 \sqcap \dots \sqcap C_k \sqsubseteq C^+, \qquad C_1 \sqcap \dots \sqcap C_k \sqsubseteq \bot,$$
$$S_1 \sqcap \dots \sqcap S_k \sqsubseteq S^+, \qquad S_1 \sqcap \dots \sqcap S_k \sqsubseteq \bot,$$

where C^+ and S^+ denote temporal concepts and roles *without* occurrences of diamond operators $\langle \mathsf{R}\rangle$; cf. Section 4.2. (The consequences of allowing $\langle \mathsf{R}\rangle$ on the right-hand side of inclusions will be discussed in the sequel). An $\mathcal{HS}$-$Lite^{\mathcal{H}}_{horn}$ ABox is a finite set of atoms of the form $A_k(a, \iota)$ and $P_k(a, b, \iota)$, where ι is an interval of the linear order in question.

It was shown [11] that answering atomic OMQs in $\mathcal{HS}$-$Lite^{\mathcal{H}}_{horn}$ is PTIME-complete for both combined and data complexity provided that either concept inclusions contain no $\exists R.\top$ on the right-hand side, or role inclusions contain no temporal relations apart from U. Originally, the result was shown for $(\mathbb{Z}, \leq)$ only; however, in the light of later findings [32], it can also be extended to any *dense* linear order $(T, \leq)$ and $(T, <)$. A failure to prove decidability for discrete linear orders under the irreflexive semantics, say, $(\mathbb{Z}, <)$, even for the language without roles, led to a separate systematic investigation of the propositional fragment $\mathcal{HS}^{\Box}_{horn}$ of $\mathcal{HS}$. Formally, this fragment can be defined as pairs of the form $(\mathcal{O}, \{A(a, \iota)\})$, where $\mathcal{O}$ is an $\mathcal{HS}$-$Lite^{\mathcal{H}}_{horn}$ TBox without any occurrence of role names. The satisfiability problem for $\mathcal{HS}^{\Box}_{horn}$ was shown [32] to be undecidable for unbounded discrete linear orders such as $(\mathbb{N}, <)$ and $(\mathbb{Z}, <)$ under the *irreflexive* semantics (in contrast to PTIME-completeness for dense orders under any semantics). We illustrate the expressiveness of $\mathcal{HS}^{\Box}_{horn}$ over $(\mathbb{N}, <)$ by the following example.

▶ **Example 17.** Let $\mathcal{A} = \{A(\langle 0, 0\rangle)\}$ and $\mathcal{O}$ be an $\mathcal{HS}^{\Box}_{horn}$ ontology with the following axioms:

$$[\mathsf{E}]A \sqcap \langle \mathsf{E}\rangle \top \sqsubseteq A, \qquad [\mathsf{E}]C \sqcap \langle \mathsf{E}\rangle \top \sqsubseteq C, \qquad \langle \bar{\mathsf{E}}\rangle [\mathsf{B}][\bar{\mathsf{E}}]A \sqsubseteq C, \qquad \langle \bar{\mathsf{E}}\rangle [\mathsf{B}][\bar{\mathsf{E}}]C \sqsubseteq A,$$

Under the irreflexive semantics over $\mathbb{N}$, we have $\mathcal{O}, \mathcal{A} \models A(\iota)$, for any $\iota = \langle n, 2m \rangle$ with $n, m \in \mathbb{N}$; see Fig. 3, where intervals $\langle i, j \rangle$ are represented as the points (i, j) on the Euclidean plane. Under the reflexive semantics, we have $\mathcal{O}, \mathcal{A} \models A(\iota)$ for $\iota = \langle 0, 0 \rangle$ only.

Furthermore, by admitting $\langle \mathsf{R} \rangle$-operators on the right-hand side of concept inclusions of $\mathcal{HS}^{\Box}_{horn}$, we make it undecidable under any semantics and any unbounded linear orders. In fact, this extended logic remains undecidable under the irreflexive semantics even when restricted to *binary* concept inclusions (that is, the *core* fragment).

5.3 Multidimensional datalog$\mathcal{HS}^{\Box}_n$

The former of the two tractable fragments of $\mathcal{HS}$-*Lite*$^{\mathcal{H}}_{horn}$ mentioned above can be generalised in two directions [58]. First, we extend *DL-Lite*$^{\mathcal{H}}_{horn}$ TBoxes without $\exists$ on the right-hand side of concept inclusions to arbitrary datalog programs. Second, following [23], we extend the interval logic $\mathcal{HS}$ to a multidimensional hyperrectangle (or block) logic $\mathcal{HS}_n$. Let $\mathfrak{T} = (T, \lhd)$ be either $(\mathbb{Z}, \leq)$ or $(\mathbb{R}, <)$. (In fact, one can take any discrete order under the reflexive semantics and any dense order under the reflexive or irreflexive semantics.) Fix some $n \geq 1$ and a linear order $\mathfrak{T}_\ell = (T_\ell, \lhd_\ell)$ as above, for $1 \leq \ell \leq n$. A *hyperrectangle* in the n-dimensional space $\mathfrak{T} = \prod_{\ell=1}^{n} \mathfrak{T}_\ell$ is any n-tuple $\boldsymbol{\iota} = (\iota_1, \dots, \iota_n)$ such that $\iota_\ell \in \mathsf{int}(\mathfrak{T}_\ell)$, for $1 \leq \ell \leq n$. The set of hyperrectangles in $\mathfrak{T}$ is denoted by $\mathsf{hyp}(\mathfrak{T})$. Given $\boldsymbol{\iota}, \boldsymbol{\kappa} \in \mathsf{hyp}(\mathfrak{T})$ and an interval relation R, we write $\boldsymbol{\iota}\, \mathsf{R}_\ell\, \boldsymbol{\kappa}$ if $\iota_\ell\, \mathsf{R}\, \kappa_\ell$ and $\iota_i = \kappa_i$, for $i \neq \ell$.

A *data instance* (ABox), $\mathcal{A}$, is now a finite set of *facts* of the form $P(\boldsymbol{c}, \boldsymbol{\iota})$, where P is an m-ary predicate symbol, $\boldsymbol{c}$ an m-tuple of individual constants, for some $m \geq 0$, and $\boldsymbol{\iota} \in \mathsf{hyp}(\mathfrak{T})$. This fact says that $P(\boldsymbol{c})$ is true in the hyperrectangle $\boldsymbol{\iota}$. We denote by $\mathsf{num}_\ell(\mathcal{A})$ the set of $i, j \in T_\ell$ with $\iota_\ell = \langle i, j \rangle$, for some $\boldsymbol{\iota}$ mentioned in $\mathcal{A}$, and by $\mathsf{int}(\mathcal{A})$ the set of $\langle i, j \rangle \in \mathsf{int}(\mathfrak{T}_\ell)$ with $i, j \in \mathsf{num}_\ell(\mathcal{A})$, for $1 \leq \ell \leq n$.

An *individual term*, τ, is an individual variable, x, or a constant, a. A datalog$\mathcal{HS}^{\Box}_n$ *program*, Π, is a finite set of *rules* of the form

$$A^+ \leftarrow A_1 \wedge \dots \wedge A_k, \qquad\qquad \bot \leftarrow A_1 \wedge \dots \wedge A_k, \tag{11}$$

where $k \geq 1$, each A_i is either an inequality $(\tau \neq \tau')$ with individual terms τ and τ' or defined by the grammar

$$A \quad ::= \quad P(\tau_1, \dots, \tau_m) \quad | \quad [\mathsf{R}]_\ell A \quad | \quad \langle \mathsf{R} \rangle_\ell A, \tag{12}$$

for an m-ary predicate P and individual terms τ_j, and A^+ does not contain any diamond operators $\langle \mathsf{R} \rangle_\ell$. As usual, the atoms $A_1, \dots, A_k$ constitute the *body* of the rule, while A^+ or $\bot$ its *head*. We also impose other standard datalog restrictions on datalog$\mathcal{HS}^{\Box}_n$ programs. (Clearly, we cannot allow $\langle \mathsf{R} \rangle_\ell$ in the heads as this would make our logic undecidable, as discussed above.)

An *interpretation*, $\mathfrak{M}$, for datalog$\mathcal{HS}^{\Box}_n$ programs is based on a *domain* $\Delta \neq \emptyset$ (for the individual variables and constants) and the space $\mathfrak{T}$. For any m-ary predicate P, m-tuple $\boldsymbol{c}$ from Δ and $\boldsymbol{\iota} \in \mathsf{hyp}(\mathfrak{T})$, $\mathfrak{M}$ specifies whether P is *true on* $\boldsymbol{c}$ *in* $\boldsymbol{\iota}$, in which case we write $\mathfrak{M}, \boldsymbol{\iota} \models P(\boldsymbol{c})$. Let $\mathfrak{d}$ be an *assignment* of elements of Δ to the individual variables (we adopt the standard name assumption: $\mathfrak{d}(a) = a$, for every individual constant a). We then set inductively:

$$\begin{array}{lll}
\mathfrak{M}, \boldsymbol{\iota} \models^{\mathfrak{d}} P(\boldsymbol{\tau}) & \text{iff} & \mathfrak{M}, \boldsymbol{\iota} \models P(\mathfrak{d}(\boldsymbol{\tau})), \qquad\qquad \mathfrak{M}, \boldsymbol{\iota} \not\models^{\mathfrak{d}} \bot, \\
\mathfrak{M}, \boldsymbol{\iota} \models^{\mathfrak{d}} \tau \neq \tau' & \text{iff} & \mathfrak{d}(\tau) \neq \mathfrak{d}(\tau'), \\
\mathfrak{M}, \boldsymbol{\iota} \models^{\mathfrak{d}} [\mathsf{R}]_\ell A & \text{iff} & \mathfrak{M}, \boldsymbol{\kappa} \models^{\mathfrak{d}} A \text{ for all } \boldsymbol{\kappa} \text{ with } \boldsymbol{\iota}\, \mathsf{R}_\ell\, \boldsymbol{\kappa}, \\
\mathfrak{M}, \boldsymbol{\iota} \models^{\mathfrak{d}} \langle \mathsf{R} \rangle_\ell A & \text{iff} & \mathfrak{M}, \boldsymbol{\kappa} \models^{\mathfrak{d}} A \text{ for some } \boldsymbol{\kappa} \text{ with } \boldsymbol{\iota}\, \mathsf{R}_\ell\, \boldsymbol{\kappa}.
\end{array}$$

Figure 4 Configurations for Int and Cov in Example 18.

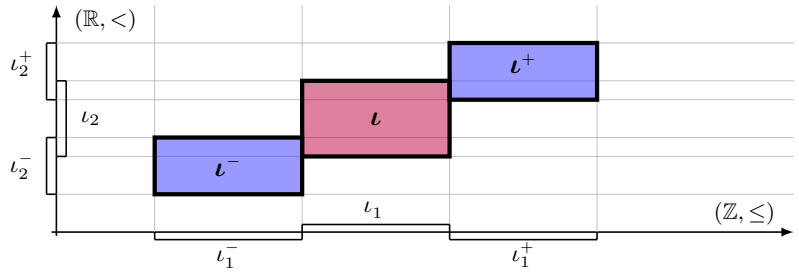

Figure 5 Rule for TemperatureRise in Example 19.

We say that $\mathfrak{M}$ *satisfies* Π *under* $\mathfrak{d}$ if

$$\mathfrak{M}, \boldsymbol{\iota} \models^{\mathfrak{d}} A \quad \text{whenever} \quad \mathfrak{M}, \boldsymbol{\iota} \models^{\mathfrak{d}} A_i \text{ for } 1 \leq i \leq k,$$

for all $\boldsymbol{\iota} \in \mathsf{hyp}(\mathfrak{T})$ and all rules $A \leftarrow A_1 \wedge \dots \wedge A_k$ in Π. $\mathfrak{M}$ is a *model* of Π and $\mathcal{A}$ if it satisfies Π under every assignment, and $\mathfrak{M}, \boldsymbol{\iota} \models P(\boldsymbol{c})$, for every fact $P(\boldsymbol{c}, \boldsymbol{\iota})$ in $\mathcal{A}$. Π and $\mathcal{A}$ are *consistent* if they have a model.

▶ **Example 18.** Denote by $\langle \mathsf{Int} \rangle$ the binary modal operator such that $A\langle \mathsf{Int} \rangle A'$ holds at a hyperrectangle $\boldsymbol{\kappa}$ iff A holds at some $\boldsymbol{\iota}$, A' at some $\boldsymbol{\iota}'$, and $\boldsymbol{\kappa} = \boldsymbol{\iota} \cap \boldsymbol{\iota}'$. One can show that rules such as $B \leftarrow A\langle \mathsf{Int} \rangle A'$ are expressible as datalog$\mathcal{HS}_n^\Box$ programs. For example, for $n = 2$, there are $13^2 = 169$ different relative positions of two rectangles; see, e.g., [65, Fig. 4] for an illustration. Those configurations where the rectangles have non-empty intersection are encoded by datalog$\mathcal{HS}_n^\Box$ rules such as $B \leftarrow \langle \bar{\mathsf{E}} \rangle_1 \langle \bar{\mathsf{B}} \rangle_2 A \wedge \langle \bar{\mathsf{B}} \rangle_1 \langle \bar{\mathsf{E}} \rangle_2 A'$ for the configuration in Fig. 4a. Similarly, one can express the rule $B \leftarrow A\langle \mathsf{Cov} \rangle A'$ such that $A\langle \mathsf{Cov} \rangle A'$ holds at $\boldsymbol{\kappa}$ iff $\boldsymbol{\kappa}$ is the smallest hyperrectangle containing some $\boldsymbol{\iota}$ with A and $\boldsymbol{\iota}'$ with A'; see Fig. 4b.

An *interval term*, ϑ, is either an interval or an *interval variable*. A *conjunctive query* (CQ) is a formula of the form $\boldsymbol{q}(\boldsymbol{x}, \boldsymbol{\chi}) = \exists \boldsymbol{x}' \exists \boldsymbol{\chi}' \, \Phi(\boldsymbol{x}, \boldsymbol{x}', \boldsymbol{\chi}, \boldsymbol{\chi}')$, where Φ is a conjunction of atoms $P(\boldsymbol{\tau}, \boldsymbol{\vartheta})$ for tuples $\boldsymbol{\tau}$ and $\boldsymbol{\vartheta}$ of individual and interval terms, respectively, and $\mathsf{R}(\vartheta, \vartheta')$, for an interval relation R, such that all individual and interval variables in Φ are from $\boldsymbol{x} \cup \boldsymbol{x}'$ and $\boldsymbol{\chi} \cup \boldsymbol{\chi}'$, respectively. A datalog$\mathcal{HS}_n^\Box$ program Π and a CQ $\boldsymbol{q}(\boldsymbol{x}, \boldsymbol{\chi})$ constitute an *ontology-mediated query* (OMQ) $\boldsymbol{Q}(\boldsymbol{x}, \boldsymbol{\chi}) = (\Pi, \boldsymbol{q}(\boldsymbol{x}, \boldsymbol{\chi}))$.

▶ **Example 19.** Suppose $\mathfrak{T} = \mathfrak{T}_1 \times \mathfrak{T}_2$, where $\mathfrak{T}_1 = (\mathbb{Z}, \leq)$ represents time and $\mathfrak{T}_2 = (\mathbb{R}, <)$ temperature. Imagine that a turbine monitoring system is receiving from sensors a stream of data of the form $\mathsf{Blade}(\mathsf{ID140}, (\iota_1, \iota_2))$, where ID140 is a blade ID and $\iota_2 \in \mathsf{int}(\mathbb{R}, <)$ is the observed temperature range during the time interval $\iota_1 \in \mathsf{int}(\mathbb{Z}, \leq)$. Then the rule

$$\mathsf{TemperatureRise}(x) \leftarrow \langle \bar{\mathsf{A}} \rangle_1 \langle \bar{\mathsf{O}} \rangle_2 \mathsf{Blade}(x) \wedge \langle \mathsf{A} \rangle_1 \langle \mathsf{O} \rangle_2 \mathsf{Blade}(x)$$

says that the temperature of blade x is rising over a rectangle (ι_1, ι_2) if $\mathsf{Blade}(x, (\iota_1^-, \iota_2^-))$ and $\mathsf{Blade}(x, (\iota_1^+, \iota_2^+))$ hold at some (ι_1^-, ι_2^-) and (ι_1^+, ι_2^+) located as shown in Fig. 5. The temperature drop is defined analogously:

$$\mathsf{TemperatureDrop}(x) \leftarrow \langle \bar{\mathsf{A}} \rangle_1 \langle \mathsf{O} \rangle_2 \mathsf{Blade}(x) \wedge \langle \mathsf{A} \rangle_1 \langle \bar{\mathsf{O}} \rangle_2 \mathsf{Blade}(x).$$

To find the blades x and the time intervals χ such that the temperature of x was rising before χ, reaching 1500° in χ, and dropping after that, we can use the following CQ:

$$\exists\rho\exists\chi^-\exists\rho^-\exists\chi^+\exists\rho^+\,\big[\mathsf{Blade}(x,(\chi,\rho))\wedge\mathsf{TemperatureRise}(x,(\chi^-,\rho^-))\wedge\mathsf{A}(\chi^-,\chi)\wedge \mathsf{TemperatureDrop}(x,(\chi^+,\rho^+))\wedge\mathsf{A}(\chi,\chi^+)\wedge\mathsf{O}(\rho,\langle 1500,1600\rangle)\big].$$

Let $\boldsymbol{Q}(\boldsymbol{x},\boldsymbol{\chi})=(\Pi,\boldsymbol{q}(\boldsymbol{x},\boldsymbol{\chi}))$ be an OMQ and $\mathcal{A}$ a data instance. A *certain answer to* $\boldsymbol{Q}(\boldsymbol{x},\boldsymbol{\chi})$ *over* $\mathcal{A}$ is any pair $(\boldsymbol{a},\boldsymbol{\delta})$ of a tuple $\boldsymbol{a}$ of individual constants in $\mathcal{A}$ and a tuple $\boldsymbol{\delta}$ from $\mathsf{int}(\mathcal{A})$ of the same length as $\boldsymbol{x}$ and $\boldsymbol{\chi}$, respectively, satisfying the following condition: for every model $\mathfrak{M}$ of Π and $\mathcal{A}$, there is a map h of the individual terms in $\boldsymbol{q}$ to Δ and the interval terms to $\bigcup_\ell \mathsf{int}(\mathfrak{T}_\ell)$ preserving constants and dimensions such that $h(\boldsymbol{x})=\boldsymbol{a}$, $h(\boldsymbol{\chi})=\boldsymbol{\delta}$, and

$$\mathfrak{M},h(\boldsymbol{\vartheta})\models P(h(\boldsymbol{\tau})),\ \text{for every atom } P(\boldsymbol{\tau},\boldsymbol{\vartheta}) \text{ in } \boldsymbol{q},\quad \text{and}$$
$$\mathsf{R}(h(\vartheta),h(\vartheta'))\ \text{holds in the corresponding } \mathfrak{T}_\ell,\ \text{for every atom } \mathsf{R}(\vartheta,\vartheta') \text{ in } \boldsymbol{q}.$$

The problem of checking whether $(\boldsymbol{a},\boldsymbol{\delta})$ is a certain answer to $\boldsymbol{Q}(\boldsymbol{x},\boldsymbol{\chi})$ over $\mathcal{A}$ is shown to be PTime-complete for data complexity and ExpTime-complete for combined complexity; for propositional datalog$\mathcal{HS}_n^\Box$ programs, the problem is PTime-complete for combined complexity [58]. Any datalog$\mathcal{HS}_n^\Box$ OMQ $\boldsymbol{Q}(\boldsymbol{x},\boldsymbol{\chi})=(\Pi,\boldsymbol{q}(\boldsymbol{x},\boldsymbol{\chi}))$ can also be rewritten to a standard polynomial-size datalog program $\Pi^\dagger$ with a goal $G(\boldsymbol{x},\boldsymbol{\chi})$ such that, for any data instance $\mathcal{A}$, a tuple $(\boldsymbol{a},\boldsymbol{\delta})$ is a certain answer to $\boldsymbol{Q}(\boldsymbol{x},\boldsymbol{\chi})$ over $\mathcal{A}$ iff $\Pi^\dagger,\mathcal{A}\models G(\boldsymbol{a},\boldsymbol{\delta})$. We refer the reader to [58] for some initial experiments on the expressive power and efficiency of ontology-based query answering with datalog$\mathcal{HS}_n^\Box$ using two real-world scenarios.

6 Dense Time and Metric Temporal Logics

The problems with discreteness of time are related to the fact that a minimal unit of time in some cases may be unknown or inconvenient to use. Suppose, for example, that the time unit is set to be 'a minute' for the turbine performance monitoring system with timestamped data of the form $(\mathsf{tb007},1500,11{:}27)$. When a newer model of turbine is installed with measurements taken at the rate of one per second, we shall have to redefine the minimal unit accordingly. This means, in particular, that the timestamps of the old data will also have to be multiplied by 60 together with all the operators used in the ontology and queries (e.g., $\bigwedge_{i=0}^{60}\bigcirc_P^i\mathsf{LowSpeed}\sqsubseteq\mathsf{Alert}$ saying that an alert is to be issued if a turbine maintained low speed for 1 hour). On the other hand, if we assume that time is dense and use rational numbers to refer to time instants, then we can represent timestamps such as 11:27:30 of the new turbine as $i+\frac{1}{2}$ (assuming that 11:27 and 11:28 correspond to integer numbers i and $i+1$, respectively), keeping the old timestamps and the ontology intact. However, for dense time, we cannot use the inherently discrete *LTL* operators in the ontology and queries any longer, and shall have to switch to a different temporal formalism with, say, *metric* interval operators, in which case the axiom above will have to be rewritten as $\boxminus_{[0,60]}\mathsf{LowSpeed}\sqsubseteq\mathsf{Alert}$, where $\boxminus_{[0,60]}\mathsf{LowSpeed}$ is true at a moment i iff $\mathsf{LowSpeed}$ holds at every j such that $i-j\in[0,60]$.

6.1 *datalogMTL*$^\Box$

In the standard metric temporal logic *MTL* [3], the temporal domain is the real numbers $\mathbb{R}$, while the intervals ϱ in the constrained temporal operators such as $\boxminus_\varrho$ (always in the past within the interval ϱ from now) have natural numbers or ∞ as their endpoints. For various

applications, it would be more appropriate to assume that the endpoints of ϱ are non-negative rational numbers or ∞, while the temporal domain is the rational numbers $\mathbb{Q}$ (however, in theory, not much will change if we take $\mathbb{R}$ as the temporal domain). Thus, by an *interval*, ι, we mean in this section any nonempty subset of $\mathbb{Q}$ of the form $[i, j]$, $[i, j)$, $(i, j]$ or (i, j), where $i, j \in \mathbb{Q} \cup \{-\infty, \infty\}$ and $i \leq j$. (We identify $(i, \infty]$ with (i, ∞), $[-\infty, i]$ with $(-\infty, i]$, etc.) The set of all intervals in $\mathbb{Q}$ is denoted by $\mathsf{int}(\mathbb{Q})$. A *range*, ϱ, is an interval with non-negative endpoints.

As in Section 5.3, we take datalog as the domain ontology language and combine it with *MTL*. Thus, a data instance (ABox), $\mathcal{A}$, is a finite set of facts of the form $P(\boldsymbol{a})@\iota$, where P is an m-ary predicate symbol, $\boldsymbol{a}$ an m-tuple of individual constants, for some $m \geq 0$, and $\iota \in \mathsf{int}(\mathbb{Q})$. This fact says that $P(\boldsymbol{a})$ is true at *each point of time* in the interval ι. To reflect this subtle semantical difference from Section 5, we write $P(\boldsymbol{a})@\iota$ rather than $P(\boldsymbol{a}, \iota)$. The following facts are an example of a data instance:

$$\begin{aligned} &\mathsf{Turbine}(\mathsf{tb0})@(-\infty, \infty), && \mathsf{ActivePowerAbove1.5}(\mathsf{tb0})@[13{:}00{:}00, 13{:}00{:}10), \\ &&& \mathsf{ActivePowerAbove1.5}(\mathsf{tb0})@[13{:}00{:}08, 13{:}00{:}15), \\ &&& \mathsf{ActivePowerBelow0.15}(\mathsf{tb0})@[13{:}00{:}17, 13{:}01{:}25). \end{aligned} \tag{13}$$

Brandt et al. [29] consider *atomic queries* of the form $\boldsymbol{q}(\boldsymbol{x}, \chi) = P(\boldsymbol{\tau})@\chi$, where P is a predicate name, $\boldsymbol{x}$ is a tuple of all individual variables occurring in the terms $\boldsymbol{\tau}$, and χ an *interval variable*. For example, the answers to the query $\boldsymbol{q}(\chi) = \mathsf{ActivePowerAbove1.5}(\mathsf{tb0})@\chi$ over the data instance above contain (among others) the intervals $[13{:}00{:}00, 13{:}00{:}10)$, $(13{:}00{:}05, 13{:}00{:}10)$, and $[13{:}00{:}00, 13{:}00{:}15)$ (the semantics will be defined below), as this information is contained, explicitly or implicitly, in $\mathcal{A}$. In a practical OBDA system, however, the returned result should be limited to the last interval only, $[13{:}00{:}00, 13{:}00{:}15)$, because it *includes* all other answers.

The temporal ontology language *datalogMTL*$^\Box$ uses the rules of the form (11), where the atoms A are defined by the grammar

$$A \quad ::= \quad \top \quad | \quad P(\tau_1, \dots, \tau_m) \quad | \quad \boxplus_\varrho A \quad | \quad \boxminus_\varrho A \quad | \quad A_1 \, \mathcal{S}_\varrho \, A_2 \quad | \quad A_1 \, \mathcal{U}_\varrho \, A_2$$

and A^+ is as above but without any 'non-deterministic' operators $\mathcal{U}_\varrho$ and $\mathcal{S}_\varrho$; cf. (12). We also use standard abbreviations $\diamondminus_\varrho A = \top \, \mathcal{S}_\varrho \, A$ and $\diamondplus_\varrho A = \top \, \mathcal{U}_\varrho \, A$. A *datalogMTL*$^\Box$ program is a finite set of rules.

▶ **Example 20.** For instance, the rule

$$\begin{aligned} \mathsf{ActivePowerTrip}(x) \leftarrow \mathsf{Turbine}(x) \wedge \boxminus_{[0,1m]} \mathsf{ActivePowerBelow0.15}(x) \wedge \\ \diamondminus_{[60s,63s]} \boxminus_{[0,10s]} \mathsf{ActivePowerAbove1.5}(x) \end{aligned} \tag{14}$$

says that an active power trip happens when the active power of a turbine was above 1.5MW for a period of at least 10 seconds, maximum 3 seconds after which there was a period of at least one minute where the active power was below 0.15MW, as shown in Fig. 6.

The semantics of query answering in *datalogMTL*$^\Box$ is essentially point-based. Thus, an *interpretation*, $\mathfrak{M}$, is based on a *domain* $\Delta \neq \emptyset$ for the individual variables and constants. For any m-ary predicate P, m-tuple $\boldsymbol{c}$ from Δ, and any moment of time $i \in \mathbb{Q}$, the interpretation $\mathfrak{M}$ specifies whether P is *true on* $\boldsymbol{c}$ *at* i, in which case we write $\mathfrak{M}, i \models P(\boldsymbol{c})$. As before, $\mathfrak{d}$ is an *assignment* of elements of Δ to the individual variables (we adopt the standard name

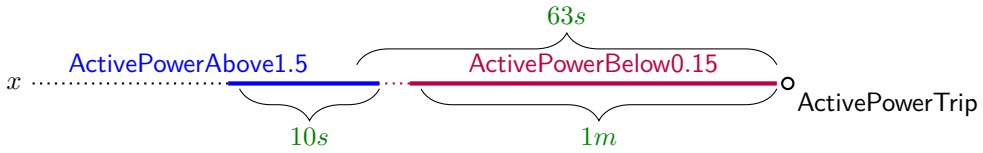

Figure 6 ActivePowerTrip.

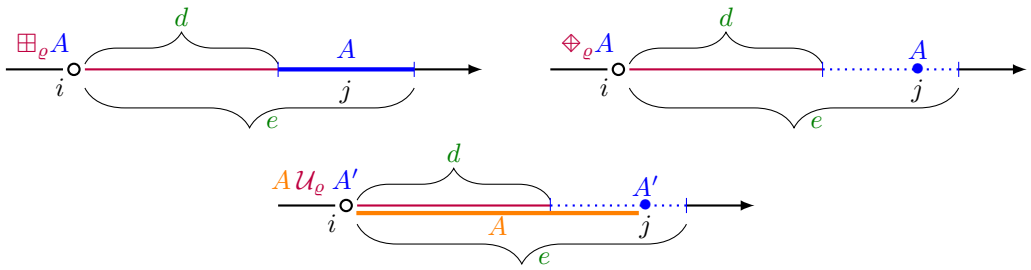

Figure 7 Semantics of metric temporal operators for $\varrho = [d, e]$.

assumption: $\mathfrak{d}(a) = a$, for every individual constant a). We then set inductively:

$$
\begin{aligned}
&\mathfrak{M}, i \models^{\mathfrak{d}} \top, \quad \text{and} \quad \mathfrak{M}, i \not\models^{\mathfrak{d}} \bot,\\
&\mathfrak{M}, i \models^{\mathfrak{d}} P(\boldsymbol{\tau}) \quad \text{iff} \quad \mathfrak{M}, i \models P(\mathfrak{d}(\boldsymbol{\tau})),\\
&\mathfrak{M}, i \models^{\mathfrak{d}} (\tau \neq \tau') \quad \text{iff} \quad \mathfrak{d}(\tau) \neq \mathfrak{d}(\tau'),\\
&\mathfrak{M}, i \models^{\mathfrak{d}} \boxplus_{\varrho} A \quad \text{iff} \quad \mathfrak{M}, j \models^{\mathfrak{d}} A \text{ for all } j \text{ with } j - i \in \varrho,\\
&\mathfrak{M}, i \models^{\mathfrak{d}} \boxminus_{\varrho} A \quad \text{iff} \quad \mathfrak{M}, j \models^{\mathfrak{d}} A \text{ for all } j \text{ with } i - j \in \varrho,\\
&\mathfrak{M}, i \models^{\mathfrak{d}} A_1 \,\mathcal{U}_{\varrho}\, A_2 \quad \text{iff} \quad \mathfrak{M}, i' \models^{\mathfrak{d}} A_2 \text{ for some } i' \text{ with } i' - i \in \varrho \text{ and}\\
&\qquad\qquad \mathfrak{M}, j \models^{\mathfrak{d}} A_1 \text{ for all } j \in (i, i'),\\
&\mathfrak{M}, i \models^{\mathfrak{d}} A_1 \,\mathcal{S}_{\varrho}\, A_2 \quad \text{iff} \quad \mathfrak{M}, i' \models^{\mathfrak{d}} A_2 \text{ for some } i' \text{ with } i - i' \in \varrho \text{ and}\\
&\qquad\qquad \mathfrak{M}, j \models^{\mathfrak{d}} A_1 \text{ for all } j \in (i', i).
\end{aligned}
$$

Figure 7 illustrates the semantics of the future-time operators for $\varrho = [d, e]$. Note that ranges ϱ in the temporal operators can be punctual $[d, d]$, in which case $\boxplus_{[d,d]} A$ is equivalent to $\diamondplus_{[d,d]} A$, and $\boxminus_{[d,d]} A$ to $\diamondminus_{[d,d]} A$. We say that $\mathfrak{M}$ *satisfies* a *datalogMTL*$^{\Box}$ program Π under an assignment $\mathfrak{d}$ if, for *all* $i \in \mathbb{Q}$ and all the rules $A \leftarrow A_1 \wedge \dots \wedge A_k$ in Π, we have

$$\mathfrak{M}, i \models^{\mathfrak{d}} A \quad \text{whenever} \quad \mathfrak{M}, i \models^{\mathfrak{d}} A_n \quad \text{for } 1 \leq n \leq k.$$

We call $\mathfrak{M}$ a *model* of Π and $\mathcal{A}$ and write $\mathfrak{M} \models (\Pi, \mathcal{A})$ if $\mathfrak{M}$ satisfies Π under every assignment, and $\mathfrak{M}, i \models P(\boldsymbol{a})$ for any $P(\boldsymbol{a})@\iota$ in $\mathcal{A}$ and any $i \in \iota$. Π and $\mathcal{A}$ are *consistent* if they have a model.

A *datalogMTL*$^{\Box}$ *ontology-mediated query* is of the form $(\Pi, \boldsymbol{q}(\boldsymbol{x}, \chi))$, where Π is a *datalogMTL*$^{\Box}$ program and $\boldsymbol{q}(\boldsymbol{x}, \chi)$ is an atomic query $P(\boldsymbol{\tau})@\chi$. A *certain answer* to $(\Pi, \boldsymbol{q}(\boldsymbol{x}, \chi))$ over a data instance $\mathcal{A}$ is a pair $(\boldsymbol{a}, \iota)$ such that $\boldsymbol{a}$ is a tuple of constants from $\mathcal{A}$ of the same length as $\boldsymbol{x}$, ι an interval and, for any $i \in \iota$, any model $\mathfrak{M}$ of Π and $\mathcal{A}$, and any assignment $\mathfrak{d}$ mapping $\boldsymbol{x}$ to $\boldsymbol{a}$, we have $\mathfrak{M}, i \models^{\mathfrak{d}} P(\boldsymbol{\tau})$. In this case, we write $\mathfrak{M}, i \models \boldsymbol{q}(\boldsymbol{a})$. To illustrate, the *datalogMTL*$^{\Box}$ query $(\Pi, \mathsf{ActivePowerTrip}(\mathsf{tb0})@\chi)$, where Π consists of rule (14), returns $[13{:}01{:}17, 13{:}01{:}18)$ as a certain answer over the data instance above.

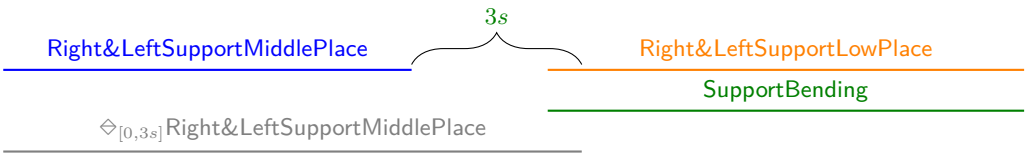

Figure 8 SupportBending in Example 21.

▶ **Example 21.** We illustrate the importance of the operators $\mathcal{S}_\varrho$ and $\mathcal{U}_\varrho$ using an example inspired by the ballet moves ontology [76]. Suppose we want to say that SupportBending is a move spanning from the beginning to the end of Right&LeftSupportLowPlace provided that it is preceded by Right&LeftSupportMiddlePlace, which ends within $3s$ from the beginning of the Right&LeftSupportLowPlace, as shown in Fig. 8. We can define the SupportBending move using the following rule:

$$\mathsf{SupportBending} \leftarrow \mathsf{Right\&LeftSupportLowPlace}\ \mathcal{S}_{[0,\infty)}\ \diamondsuit_{[0,3s]}\mathsf{Right\&LeftSupportMiddlePlace}.$$

Note that defining SupportBending in *datalogMTL*$^\Box$ would be problematic if only the $\Box$ and $\diamondsuit$ operators were available.

Atomic OMQ evaluation with *datalogMTL*$^\Box$ has been studied by Brandt et al. [29]. In particular, it was shown to be decidable and ExpSpace-complete for combined complexity. This result holds even with punctual temporal operators (with range $[d, d]$), in which case the propositional *MTL* is known to be undecidable [4]; on the other hand, the propositional *MTL* is ExpSpace-complete if the punctual operators are not allowed [3]; see also [66, 67]. In fact, the undecidability result in the presence of punctual operators holds even for the propositional (predicates of arity 0 only) fragment of *datalogMTL*$^\Box$ extended by $\diamondsuit_\varrho$ and $\boxminus_\varrho$ operators in the head of rules [29] (cf. $\mathcal{HS}^\Box_{horn}$ in Section 5.2). Furthermore, it was shown that, for *nonrecursive datalogMTL*$^\Box$ programs, query answering is PSpace-complete for combined complexity and in AC^0 for data complexity.

6.2 Use Cases

The metric temporal ontology language *datalogMTL*$^\Box$ has been used to construct ontologies and support query answering in three practical use-cases [29, 76], which will be briefly discussed below.

Turbine Monitoring at Siemens At Siemens, service centres store aggregated turbine sensor data instances such as (13). A *datalogMTL*$^\Box$ ontology has been designed [29] to define events (representing normal or abnormal behaviour) that are of interest to engineers monitoring the performance of turbines. One such event is active power trip defined by (14). As another example, we show a (partial) definition of normal restart:

$$\begin{aligned}
&\mathsf{NormalRestart}(x) \leftarrow \mathsf{NormalStart}(x) \wedge \diamondsuit_{(0,1h]}\mathsf{NormalStop}(x),\\
&\mathsf{NormalStop}(x) \leftarrow \mathsf{CoastDown1500to200}(x) \wedge\ \diamondsuit_{(0,9m]}\ \big[\mathsf{CoastDown6600to1500}(x) \wedge\\
&\qquad\qquad \diamondsuit_{(0,2m]}\ \big(\mathsf{MainFlameOff}(x) \wedge\ \diamondsuit_{(0,2m]}\ \mathsf{ActivePowerOff}(x)\big)\big],\\
&\mathsf{MainFlameOff}(x) \leftarrow \boxminus_{[0s,10s]}\mathsf{MainFlameBelow0.1}(x).
\end{aligned}$$

(The complete definition of normal restart contains 12 rules.) The purpose of this ontology is to enable a convenient access to temporal information for an engineer who can pose succinct queries such as $\boldsymbol{q}(x, \chi) = \mathsf{NormalRestart}(x)@\chi$ (find the turbines that had a normal restart) without having to write explicitly (or even to know) the complex definition of this event.

Weather Monitoring The MesoWest[3] project makes publicly available historical records of the weather stations across the US showing such parameters of meteorological conditions as temperature, wind speed and direction, amount of precipitation, etc. From this data, one can extract facts such as

$$\mathsf{NorthWind}(\mathsf{KBVY})@(15{:}14, 15{:}24], \qquad \mathsf{HurricaneForceWind}(\mathsf{KMNI})@(15{:}21, 15{:}31],$$
$$\mathsf{Precipitation}(\mathsf{KBVY})@(15{:}14, 15{:}24], \qquad \mathsf{TempAbove0}(\mathsf{KBVY})@(15{:}14, 15{:}24],$$
$$\mathsf{TempAbove0}(\mathsf{KMNI})@(15{:}21, 15{:}31],$$
$$\mathsf{LocatedInCounty}(\mathsf{KBVY}, \mathsf{Essex})@(-\infty, \infty), \qquad \mathsf{LocatedInState}(\mathsf{KBVY}, \mathsf{MA})@(-\infty, \infty),$$

where KBVY, KMNI are IDs of the stations (according to the standard definition, the hurricane force wind is above 118 km/h). A snippet of a weather ontology giving meteorological definitions (such as 'a hurricane is a hurricane force wind lasting one hour or longer') is shown below:

$$\boxminus_{[0,1h]}\, \mathsf{Hurricane}(x) \leftarrow \boxminus_{[0,1h]} \mathsf{HurricaneForceWind}(x),$$
$$\mathsf{ShoweryCounty}(x) \leftarrow \mathsf{LocatedInCounty}(u_1, x) \land \mathsf{LocatedInCounty}(u_2, x) \land \mathsf{Precipitation}(u_1) \land \mathsf{NoPrecipitation}(u_2) \land \diamondminus_{(0,30m]}\, \mathsf{Precipitation}(u_2),$$
$$\mathsf{HurricaneAffectedState}(x) \leftarrow \mathsf{LocatedInState}(u, x) \land \mathsf{Hurricane}(u),$$
$$\boxminus_{[0,24h]}\, \mathsf{ExcessiveHeat}(x) \leftarrow \boxminus_{[0,24h]} \mathsf{TempAbove24}(x) \land \diamondminus_{[0,24h]}\, \mathsf{TempAbove41}(x),$$
$$\mathsf{HeatAffectedCounty}(x) \leftarrow \mathsf{LocatedInCounty}(u, x) \land \mathsf{ExcessiveHeat}(u),$$
$$\mathsf{CyclonePatternState}(x) \leftarrow \mathsf{LocatedInState}(u_1, x) \land \mathsf{LocatedInState}(u_2, x) \land \mathsf{LocatedInState}(u_3, x) \land \mathsf{LocatedInState}(u_4, x) \land \mathsf{EastWind}(u_1) \land \mathsf{NorthWind}(u_2) \land \mathsf{WestWind}(u_3) \land \mathsf{SouthWind}(u_4).$$

The purpose of using the temporal ontology in the weather use-case is to enable a weather expert to find information about complex meteorological events by using succinct queries.

BalOnSe: Ontology of Dance Movements This use-case is concerned with user annotations of ballet videos such as

$$\mathsf{LeftLegGestureMiddleBack}(\mathsf{video1})@[12s, 13s]$$

saying that the movement LeftLegGestureMiddleBack is shown in video1 from 00:12:00 to 00:13:00. The ballet ontology [76] reflects the terminology developed by ballet researchers and contains rules such as

$$\boxplus_{[0,3s]} \mathsf{PlieReleve}(x) \leftrightarrow \boxplus_{[0,1s]}\, \mathsf{RightSupportMidPlace}(x) \land \boxplus_{[0,1s]} \mathsf{LeftSupportMidPlace}(x) \land \boxplus_{[1,2s]}\, \mathsf{RightSupportLowPlace}(x) \land \boxplus_{[1,2s]} \mathsf{LeftSupportLowPlace}(x) \land \boxplus_{[2,3s]}\, \mathsf{RightSupportHighPlace}(x) \land \boxplus_{[2,3s]} \mathsf{LeftSupportHighPlace}(x)$$

defining the composite movement *plie releve* as a sequence of simpler movements occurring simultaneously or in a sequence. The video annotations together with the ontology are then used to enhance the search capabilities of a video search system for ballet learners and scholars. Thus, searching the term *plie releve* will return the videos (and time spans in them) showing this movement, even if the annotation for this sequence is not explicitly present in the database, but is deducible from the ontology and other annotations.

[3] `http://mesowest.utah.edu/`

7 Ontology-Based Data Access and Implementations

In real-world applications, the data instances (ABoxes) are not created from scratch. In fact, they are obtained from existing relational or RDF databases by means of *mappings* (queries in the language of a data source) in order to produce a high-level conceptual view of the data. Such ABoxes can be materialised and stored as, e.g., RDF triples, or remain virtual (as a potential result of applying the mapping to the data), in which case an ontology-mediated query may be evaluated by rewriting it into a set of queries in the language(s) of the data sources. In the this section, we briefly address the problem of converting raw data to an ABox in the context of temporal data. After that we present some prototypical implementations of temporal ontology-based data access and evaluations of their performance.

7.1 From Raw to Conceptual Temporal Data

Suppose turbine sensor measurements are stored in a relational table TB_Sensor:

turbineId	dateTime	activePower	rotorSpeed	mainFlame	...
tb0	2015-04-04 12:20:48	2	1550	0	
tb0	2015-04-04 12:20:49	1.8	1400	null	
tb0	2015-04-04 12:20:52	1.7	1350	1	
		...			

There are three major options for conceptualising this data if, for instance, we are interested in the situations when the rotor speed was below 1500:

- $\mathsf{RotorSpeedBelow1500}(\mathsf{tb0}, i)$, $\mathsf{RotorSpeedBelow1500}(\mathsf{tb0}, i+1)$, where i is the timestamp in the first row of the table, and so $i+1$ is the number of the second timestamp. This is the most simplistic approach that ignores the distance between the timestamps, but it is suitable if the timestamps are present in the database at regular intervals (which is not the case above), or only the sequence of events rather than the duration of the gap between them is important.
- $\mathsf{RotorSpeedBelow1500}(\mathsf{tb0}, i)$, $\mathsf{RotorSpeedBelow1500}(\mathsf{tb0}, i+3)$, where i is the timestamp in the first row of the table. Here, we obviously make an assumption that the time unit in our application domain is 'a second', and so this approach takes into account the duration of the gap between the events.
- $\mathsf{RotorSpeedBelow1500}(\mathsf{tb0}, \langle i, i+3\rangle)$, where $\langle i, i+3\rangle$ is a time interval. Here, we use a real-world assumption that a rotor speed sensor sends its measurements only when the current value of the speed is sufficiently different from the previous measurement, and this value is assumed to hold for all the times until the next one is produced. Note also that some sensors may produce aggregated (e.g., average) value taken over some period.

The choice of how to conceptualise the data depends on the application domain. Below, we follow the third approach and show a mapping (in the syntax similar to the standard R2RML mapping language, where in the body we use standard SQL with window operators) that extracts the data instance related to the situations when active power was above 1.5MW:

$\mathsf{ActivePowerAbove1.5}(\mathsf{tbid})@[\mathtt{ledge}, \mathtt{redge}) \leftarrow$

```
SELECT tbid, ledge, redge FROM (
    SELECT turbineId AS tbid,  LAG(dateTime, 1) OVER (w) AS ledge,
           LAG(activePower, 1) OVER (w) AS lag_activePower,  dateTime AS redge
    FROM TB_Sensor
    WINDOW w AS (PARTITION BY turbineId ORDER BY dateTime)) tmp
WHERE lag_activePower > 1.5
```

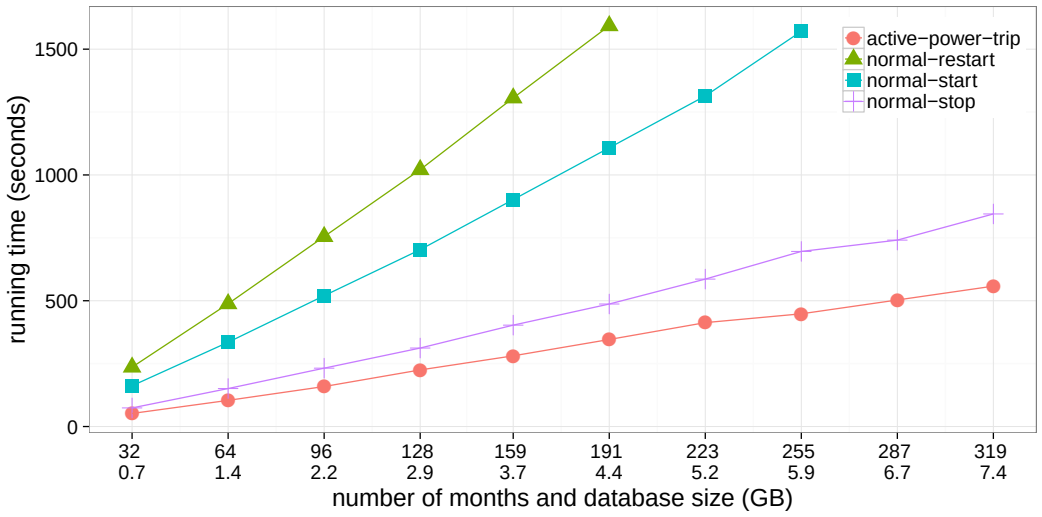

Figure 9 Performance of queries in the Siemens use-case.

The mapping above applied to TB_Sensor will produce the following instance:

$$\mathsf{ActivePowerAbove1.5(tb0)@[12{:}20{:}48, 12{:}20{:}49)},$$
$$\mathsf{ActivePowerAbove1.5(tb0)@[12{:}20{:}49, 12{:}20{:}52)}.$$

Note that we use the definition of interval from Section 6 and make an assumption (reflecting our intuition on how sensors produce their measurements) that the intervals involved are all of the form $[i, j)$. Clearly, we can add similar mappings for the concepts RotorSpeedAbove1500 and MainFlameBelow0.1.

7.2 Implementation

We report on the implementation of temporal ontology-based data access and its evaluation [29]. The ontology language supported by this implementation is $\mathit{datalog}_{nr}\mathit{MTL}^{\Box}$ consisting of nonrecursive $\mathit{datalogMTL}^{\Box}$ programs, and the system rewrites $\mathit{datalog}_{nr}\mathit{MTL}^{\Box}$ OMQs to standard SQL queries with views. The performance the rewritings for the Siemens use-case described in Section 6.2 was evaluated on an HP Proliant server with 24 Intel Xeon CPUs (@3.47GHz), 106GB of RAM and five 1TB 15K RPM HD, which used PostgreSQL as a database engine. The maximum physical memory consumption in the experiments was 12.9GB.

Siemens supplied a sample of data for one running turbine, denoted tb0, over 4 days in the form of the table TB_Sensor. This sample was replicated to imitate the data for one turbine over 10 different periods ranging from 32 to 320 months. Four queries $\mathsf{ActivePowerTrip(tb0)}@\chi$, $\mathsf{NormalStart(tb0)}@\chi$, $\mathsf{NormalStop(tb0)}@\chi$, and $\mathsf{NormalRestart(tb0)}@\chi$ were evaluated with a timeout of 30 minutes. The execution times are given in Fig. 9, which shows their linear growth in the number of months and, consequently, in the size of data. Note that the normal restart (start) query timeouts on the data for more than 15 (respectively, 20) years, which is more than enough for the monitoring and diagnostics tasks at Siemens, where the two most common application scenarios for sensor data analytics are daily monitoring (that is, analytics of high-frequency data of the previous 24 hours) and fleet-level analytics of

key-performance indicators over one year. In both cases, the computation time of the results is far less a crucial cost factor than the lead-time for data preparation.

The evaluation was performed for the weather OMQs with MesoWest data (see Section 6.2) as well. On the other hand, the system SPARK capable of parallel query processing, in place of PostgreSQL, was evaluated showing large performance improvements in some cases; for details consult [30].

References

1 Luca Aceto, Dario Della Monica, Valentin Goranko, Anna Ingólfsdóttir, Angelo Montanari, and Guido Sciavicco. A complete classification of the expressiveness of interval logics of Allen's relations: the general and the dense cases. *Acta Inf.*, 53(3):207–246, 2016.

2 James F. Allen. Maintaining knowledge about temporal intervals. *Commun. ACM*, 26(11):832–843, 1983.

3 Rajeev Alur, Tomás Feder, and Thomas A. Henzinger. The benefits of relaxing punctuality. *J. ACM*, 43(1):116–146, 1996.

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

5 Natalia Antonioli, Francesco Castanò, Spartaco Coletta, Stefano Grossi, Domenico Lembo, Maurizio Lenzerini, Antonella Poggi, Emanuela Virardi, and Patrizia Castracane. Ontology-based data management for the Italian public debt. In *Proc. of the 8th Int. Conf. on Formal Ontology in Information Systems, FOIS 2014*, pages 372–385. IOS Press, 2014.

6 Alessandro Artale, Diego Calvanese, Roman Kontchakov, and Michael Zakharyaschev. The *DL-Lite* family and relations. *J. Artif. Intell. Res. (JAIR)*, 36:1–69, 2009.

7 Alessandro Artale and Enrico Franconi. Temporal description logics. In *Handbook of Temporal Reasoning in Artificial Intelligence*, volume 1 of *Foundations of Artificial Intelligence*, pages 375–388. Elsevier, 2005.

8 Alessandro Artale, Roman Kontchakov, Alisa Kovtunova, Vladislav Ryzhikov, Frank Wolter, and Michael Zakharyaschev. First-order rewritability of temporal ontology-mediated queries. In *Proc. of the 24th Int. Joint Conf. on Artificial Intelligence, IJCAI'15*, pages 2706–2712. IJCAI/AAAI, 2015.

9 Alessandro Artale, Roman Kontchakov, Vladislav Ryzhikov, and Michael Zakharyaschev. The complexity of clausal fragments of LTL. In *Proc. of the 19th Int. Conf. on Logic for Programming, Artificial Intelligence, and Reasoning, LPAR'13*, volume 8312 of *LNCS*, pages 35–52. Springer, 2013.

10 Alessandro Artale, Roman Kontchakov, Vladislav Ryzhikov, and Michael Zakharyaschev. A cookbook for temporal conceptual data modelling with description logics. *ACM Trans. Comput. Log.*, 15(3):25:1–25:50, 2014.

11 Alessandro Artale, Roman Kontchakov, Vladislav Ryzhikov, and Michael Zakharyaschev. Tractable interval temporal propositional and description logics. In *Proc. of the 29th Conf. on Artificial Intelligence, AAAI'15*, pages 1417–1423. AAAI Press, 2015.

12 Alessandro Artale, Roman Kontchakov, Frank Wolter, and Michael Zakharyaschev. Temporal description logic for ontology-based data access. In *Proc. of the 23rd Int. Joint Conf. on Artificial Intelligence, IJCAI'13*, pages 711–717. IJCAI/AAAI, 2013.

13 Franz Baader. Ontology-based monitoring of dynamic systems. In *Proc. of the 14th Int. Conf. on Principles of Knowledge Representation and Reasoning, KR'14*, pages 678–681. AAAI Press, 2014.

14 Franz Baader, Stefan Borgwardt, Patrick Koopmann, Ana Ozaki, and Veronika Thost. Metric temporal description logics with interval-rigid names (extended abstract). In *Proc. of the 30th Int. Workshop on Description Logics, DL'17*. CEUR-WS, 2017.

15 Franz Baader, Stefan Borgwardt, and Marcel Lippmann. Temporalizing ontology-based data access. In *Proc. of the 24th Int. Conf. on Automated Deduction, CADE-24*, volume 7898 of *LNCS*, pages 330–344. Springer, 2013.

16 Franz Baader, Stefan Borgwardt, and Marcel Lippmann. Temporal conjunctive queries in expressive description logics with transitive roles. In *Proc. of the 28th Australasian Joint Conf. on Advances in Artificial Intelligence, AI'15*, volume 9457 of *LNCS*, pages 21–33. Springer, 2015.

17 Franz Baader, Stefan Borgwardt, and Marcel Lippmann. Temporal query entailment in the description logic SHQ. *J. Web Semantics*, 33:71–93, 2015.

18 Franz Baader, Sebastian Brandt, and Carsten Lutz. Pushing the $\mathcal{EL}$ envelope. In *Proc. of the 19th Int. Joint Conf. on Artificial Intelligence, IJCAI-05*, pages 364–369. IJCAI/AAAI, 2005.

19 Franz Baader, Diego Calvanese, Deborah L. McGuinness, Daniele Nardi, and Peter F. Patel-Schneider, editors. *The Description Logic Handbook: Theory, Implementation, and Applications*. Cambridge University Press, 2003.

20 Franz Baader, Silvio Ghilardi, and Carsten Lutz. LTL over description logic axioms. In *Proc. of the 11th Int. Conf. on Principles of Knowledge Representation and Reasoning, KR 2008*, pages 684–694. AAAI Press, 2008.

21 Franz Baader, Ralf Küsters, and Frank Wolter. Extensions to description logics. In *The Description Logic Handbook*, pages 219–261. Cambridge University Press, 2003.

22 Samantha Bail, Sandra Alkiviadous, Bijan Parsia, David Workman, Mark Van Harmelen, Rafael S. Goncalves, and Cristina Garilao. Fishmark: A linked data application benchmark. In *Proc. of SSWS+HPCSW 2012*, pages 1–15. CEUR-WS, 2012.

23 Philippe Balbiani, Jean-François Condotta, and Luis Fariñas del Cerro. Tractability results in the block algebra. *J. Log. Comput.*, 12(5):885–909, 2002.

24 Meghyn Bienvenu, Balder ten Cate, Carsten Lutz, and Frank Wolter. Ontology-based data access: A study through disjunctive datalog, CSP, and MMSNP. *ACM Trans. on Database Systems*, 39(4):33:1–33:44, 2014.

25 Stefan Borgwardt, Marcel Lippmann, and Veronika Thost. Temporal query answering in the description logic *DL-Lite*. In *Proc. of the 9th Int. Symposium on Frontiers of Combining Systems, FroCoS'13*, volume 8152 of *LNCS*, pages 165–180. Springer, 2013.

26 Stefan Borgwardt, Marcel Lippmann, and Veronika Thost. Temporalizing rewritable query languages over knowledge bases. *J. Web Semantics*, 33:50–70, 2015.

27 Stefan Borgwardt and Veronika Thost. Temporal query answering in *DL-Lite* with negation. In *Proc. of the Global Conf. on Artificial Intelligence, GCAI15*, volume 36 of *EPiC Series in Computing*, pages 51–65, 2015.

28 Stefan Borgwardt and Veronika Thost. Temporal query answering in the description logic $\mathcal{EL}$. In *Proc. of the 24h Int. Joint Conf. on Artificial Intelligence, IJCAI'15*, pages 2819–2825. AAAI Press, 2015.

29 Sebastian Brandt, Elem Güzel Kalayci, Roman Kontchakov, Vladislav Ryzhikov, Guohui Xiao, and Michael Zakharyaschev. Ontology-based data access with a horn fragment of metric temporal logic. In *Proc. of the 31st AAAI Conf. on Artificial Intelligence, AAAI'17*, pages 1070–1076. AAAI Press, 2017.

30 Sebastian Brandt, Elem Güzel Kalayci, Vladislav Ryzhikov, Guohui Xiao, and Michael Zakharyaschev. Querying log data with metric temporal logic. *CoRR*, abs/1703.08982, 2017.

31 Torben Braüner and Silvio Ghilardi. First-order modal logic. In *Handbook of Modal Logic*, volume 3 of *Studies in Logic and Practical Reasoning*, pages 549–620. Elsevier, 2007.

32 Davide Bresolin, Agi Kurucz, Emilio Muñoz-Velasco, Vladislav Ryzhikov, Guido Sciavicco, and Michael Zakharyaschev. Horn fragments of the Halpern-Shoham interval temporal logic. *ACM Trans. Comput. Log.*, 18(3), 2017.

33 Davide Bresolin, Dario Della Monica, Valentin Goranko, Angelo Montanari, and Guido Sciavicco. The dark side of interval temporal logic: marking the undecidability border. *Ann. Math. Artif. Intell. (AMAI)*, 71(1-3):41–83, 2014.

34 Davide Bresolin, Dario Della Monica, Angelo Montanari, and Guido Sciavicco. The light side of interval temporal logic: the Bernays-Schönfinkel fragment of CDT. *Ann. Math. Artif. Intell. (AMAI)*, 71(1-3):11–39, 2014.

35 Richard J. Büchi. Weak second-order arithmetic and finite automata. *Zeitschrift für Mathematische Logik und Grundlagen der Mathematik*, 6(1–6):66–92, 1960.

36 Diego Calvanese, Giuseppe De Giacomo, Domenico Lembo, Maurizio Lenzerini, Antonella Poggi, Mariano Rodriguez-Muro, Riccardo Rosati, Marco Ruzzi, and Domenico Fabio Savo. The MASTRO system for ontology-based data access. *Semantic Web*, 2(1):43–53, 2011.

37 Diego Calvanese, Giuseppe De Giacomo, Domenico Lembo, Maurizio Lenzerini, and Riccardo Rosati. Tractable reasoning and efficient query answering in description logics: The *DL-Lite* family. *J. Autom. Reasoning*, 39(3):385–429, 2007.

38 Diego Calvanese, Pietro Liuzzo, Alessandro Mosca, José Remesal, Martin Rezk, and Guillem Rull. Ontology-based data integration in EPNet: Production and distribution of food during the Roman Empire. *Eng. Appl. of AI*, 51:212–229, 2016.

39 Jan Chomicki. Polynomial time query processing in temporal deductive databases. In *Proc. of the 9th ACM Symposium on Principles of Database Systems, PODS'90*, pages 379–391. ACM Press, 1990.

40 Jan Chomicki and Tomasz Imielinski. Temporal deductive databases and infinite objects. In *Proc. of the 7th ACM Symposium on Principles of Database Systems, PODS'88*, pages 61–73. ACM, 1988.

41 Stéphane Demri, Valentin Goranko, and Martin Lange. *Temporal Logics in Computer Science*. Cambridge University Press, 2016.

42 Michael Fisher, Clare Dixon, and Martin Peim. Clausal temporal resolution. *ACM Trans. Comput. Log.*, 2(1):12–56, 2001.

43 Merrick L. Furst, James B. Saxe, and Michael Sipser. Parity, circuits, and the polynomial-time hierarchy. *Mathematical Systems Theory*, 17(1):13–27, 1984.

44 Dov M. Gabbay, Ian Hodkinson, and Mark Reynolds. *Temporal Logic: Mathematical Foundations and Computational Aspects*, volume 1. Oxford University Press, 1994.

45 Dov M. Gabbay, Agi Kurucz, Frank Wolter, and Michael Zakharyaschev. *Many-Dimensional Modal Logics: Theory and Applications*. Elsevier North Holland, 2003.

46 Dov M. Gabbay, Mark A. Reynolds, and Marcelo Finger. *Temporal Logic: Mathematical Foundations and Computational Aspects*, volume 2. Oxford University Press, 2000.

47 Martin Giese, Ahmet Soylu, Guillermo Vega-Gorgojo, Arild Waaler, Peter Haase, Ernesto Jiménez-Ruiz, Davide Lanti, Martín Rezk, Guohui Xiao, Özgür L. Özçep, and Riccardo Rosati. Optique: Zooming in on big data. *IEEE Computer*, 48(3):60–67, 2015.

48 Claudio Gutierrez, Carlos A. Hurtado, and Alejandro A. Vaisman. Introducing time into RDF. *IEEE Trans. Knowl. Data Eng.*, 19(2):207–218, 2007.

49 Víctor Gutiérrez-Basulto, Jean Christoph Jung, and Roman Kontchakov. Temporalized $\mathcal{EL}$ ontologies for accessing temporal data: Complexity of atomic queries. In *Proc. of the 25th Int. Joint Conf. on Artificial Intelligence, IJCAI'16*, pages 1102–1108. IJCAI/AAAI, 2016.

50 Víctor Gutiérrez-Basulto, Jean Christoph Jung, and Ana Ozaki. On metric temporal description logics. In *Proc. of the 22nd European Conf. on Artificial Intelligence, ECAI 2016*, volume 285 of *FAIA*, pages 837–845. IOS Press, 2016.

51 Víctor Gutiérrez-Basulto, Jean Christoph Jung, and Thomas Schneider. Lightweight description logics and branching time: A troublesome marriage. In *Proc. of the 14th Int. Conf. on Principles of Knowledge Representation and Reasoning, KR'14*, pages 278–287. AAAI Press, 2014.

52 Víctor Gutiérrez-Basulto, Jean Christoph Jung, and Thomas Schneider. Lightweight temporal description logics with rigid roles and restricted TBoxes. In *Proc. of the 24th Int. Joint Conf. on Artificial Intelligence, IJCAI 2015*, pages 3015–3021. IJCAI/AAAI, 2015.

53 Joseph Y. Halpern and Yoav Shoham. A propositional modal logic of time intervals. *J. ACM*, 38(4):935–962, 1991.

54 Ian M. Hodkinson, Roman Kontchakov, Agi Kurucz, Frank Wolter, and Michael Zakharyaschev. On the computational complexity of decidable fragments of first-order linear temporal logics. In *Proc. of the 10th Int. Symposium on Temporal Representation and Reasoning and the 4th Int. Conf. on Temporal Logic, TIME-ICTL 2003*, pages 91–98. IEEE Computer Society, 2003.

55 Ullrich Hustadt, Boris Motik, and Ulrike Sattler. Data complexity of reasoning in very expressive description logics. In *Proc. of the 19th Int. Joint Conf. on Artificial Intelligence, IJCAI-05*, pages 466–471. IJCAI/AAAI, 2005.

56 Neil Immerman. *Descriptive complexity*. Springer, 1999.

57 Roman Kontchakov, Carsten Lutz, David Toman, Frank Wolter, and Michael Zakharyaschev. The combined approach to query answering in *DL-Lite*. In *Proc. of the 12th Int. Conf. on Principles of Knowledge Representation and Reasoning, KR 2010*, pages 247–257. AAAI Press, 2010.

58 Roman Kontchakov, Laura Pandolfo, Luca Pulina, Vladislav Ryzhikov, and Michael Zakharyaschev. Temporal and spatial OBDA with many-dimensional Halpern-Shoham logic. In *Proc. of the 25th Int. Joint Conf. on Artificial Intelligence, IJCAI'16*, pages 1160–1166. IJCAI/AAAI, 2016.

59 Alisa Kovtunova. *Ontology-Mediated Query Answering with Lightweight Temporal Description Logics*. PhD thesis, KRDB research centre, Faculty of Computer Science, Free University of Bozen-Bolzano, 2017.

60 Krishna G. Kulkarni and Jan-Eike Michels. Temporal features in SQL:2011. *SIGMOD Record*, 41(3):34–43, 2012.

61 Leonid Libkin. *Elements of Finite Model Theory*. Texts in Theoretical Computer Science. An EATCS Series. Springer, 2004.

62 Carsten Lutz, Frank Wolter, and Michael Zakharyaschev. Temporal description logics: A survey. In *Proc. of the 15th Int. Symposium on Temporal Representation and Reasoning, TIME'08*, pages 3–14. IEEE Computer Society, 2008.

63 Jerzy Marcinkowski and Jakub Michaliszyn. The undecidability of the logic of subintervals. *Fundam. Inform.*, 131(2):217–240, 2014.

64 Boris Motik. Representing and querying validity time in RDF and OWL: A logic-based approach. *J. Web Semantics*, 12:3–21, 2012.

65 Isabel Navarrete, Antonio Morales, Guido Sciavicco, and M. Antonia Cárdenas Viedma. Spatial reasoning with rectangular cardinal relations — the convex tractable subalgebra. *Ann. Math. Artif. Intell. (AMAI)*, 67(1):31–70, 2013.

66 Joël Ouaknine and James Worrell. On the decidability of metric temporal logic. In *Proc. of the 20th Annual IEEE Symposium on Logic in Computer Science, LICS'05*, pages 188–197. IEEE Computer Society, 2005.

67 Joël Ouaknine and James Worrell. Some recent results in metric temporal logic. In *Proc. of the 6th Int. Conf. on Formal Modeling and Analysis of Timed Systems, FORMATS'08*, pages 1–13, 2008.

68 Özgür L. Özçep, Ian Horrocks, Ralf Möller, Thomas Hubauer, Christian Neuenstadt, Mikhail Roshchin, Dmitriy Zheleznyakov, and Evgeny Kharlamov. Deliverable D5.1: A semantics for temporal and stream-based query answering in an OBDA context. Technical report, October 2013.

69 Özgür L. Özçep and Ralf Möller. Ontology based data access on temporal and streaming data. In *Proc. of the 10th Int. Summer School on Reasoning on the Web in the Big Data Era (Reasoning Web'14)*, volume 8714 of *LNCS*, pages 279–312. Springer, 2014.

70 Özgür L. Özçep, Ralf Möller, and Christian Neuenstadt. A stream-temporal query language for ontology based data access. In *Proc. of the 37th Annual German Conf. on AI, KI'14*, pages 183–194. Springer, 2014.

71 Francesco Pagliarecci, Luca Spalazzi, and Gilberto Taccari. Reasoning with temporal aboxes: Combining *DL-Lite*$_{core}$ with CTL. In *Proc. of the 26th Int. Workshop on Description Logics, DL'13*, pages 885–897. CEUR-WS, 2013.

72 Amir Pnueli. The temporal logic of programs. In *Proc. of the 18th Annual Symposium on Foundations of Computer Science, FOCS'77*, pages 46–57. IEEE Computer Society, 1977.

73 Antonella Poggi, Domenico Lembo, Diego Calvanese, Giuseppe De Giacomo, Maurizio Lenzerini, and Riccardo Rosati. Linking data to ontologies. *J. Data Semantics*, 10:133–173, 2008.

74 Andrea Pugliese, Octavian Udrea, and V. S. Subrahmanian. Scaling RDF with time. In *Proc. of the 17th Int. Conf. on World Wide Web, WWW'08*, pages 605–614. ACM, 2008.

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

76 Katerina El Raheb, Theofilos Mailis, Vladislav Ryzhikov, Nicolas Papapetrou, and Yannis E. Ioannidis. Balonse: Temporal aspects of dance movement and its ontological representation. In *Proc. of the 14th Int. Conf., ESWC 2017, Part II*, pages 49–64, 2017.

77 Mariano Rodriguez-Muro, Roman Kontchakov, and Michael Zakharyaschev. Ontology-based data access: Ontop of databases. In *Proc. of the 12th Int. Semantic Web Conf., ISWC'13, Part I*, volume 8218 of *LNCS*, pages 558–573. Springer, 2013.

78 Riccardo Rosati. On conjunctive query answering in $\mathcal{EL}$. In *Proc. of the Int. Workshop on Description Logics, DL'07*, volume 250. CEUR-WS, 2007.

79 Andrea Schaerf. On the complexity of the instance checking problem in concept languages with existential quantification. *J. Intelligent Information Systems*, 2(3):265–278, 1993.

80 Klaus Schild. Combining terminological logics with tense logic. In *Proc. of the 6th Portuguese Conf. on Progress in Artificial Intelligence, EPIA'93*, volume 727 of *Lecture Notes in Computer Science*, pages 105–120. Springer, 1993.

81 Albrecht Schmiedel. Temporal terminological logic. In *Proc. of the 8th National Conf. on Artificial Intelligence, AAAI'90*, pages 640–645. AAAI Press / The MIT Press, 1990.

82 Juan F. Sequeda and Daniel P. Miranker. A pay-as-you-go methodology for ontology-based data access. *IEEE Internet Computing*, 21(2):92–96, 2017.

83 Ahmet Soylu, Martin Giese, Rudolf Schlatte, Ernesto Jiménez-Ruiz, Evgeny Kharlamov, Özgür L. Özçep, Christian Neuenstadt, and Sebastian Brandt. Querying industrial stream-temporal data: An ontology-based visual approach. *J. Ambient Intelligence & Smart Environments*, 9(1):77–95, 2017.

84 David Toman. On incompleteness of multi-dimensional first-order temporal logics. In *Proc. of the 10th Int. Symposium on Temporal Representation and Reasoning and the 4th Int. Conf. on Temporal Logic, TIME-ICTL 2003*, pages 99–106. IEEE Computer Society, 2003.

85 Moshe Y. Vardi. The complexity of relational query languages (extended abstract). In *Proc. of the 14th Annual ACM Symposium on Theory of Computing, STOC'82*, pages 137–146. ACM, 1982.