Using Ontology-Based Data Access to Enable Context Recognition in the Presence of Incomplete Information

Kurzfassung

zur Erlangung des akademischen Grades
Doktoringenieur (Dr.-Ing.)

vorgelegt an der
Technischen Universität Dresden
Fakultät Informatik

eingereicht von
Dipl.-Inf. Veronika Thost
geboren am 16. Juni 1988 in Konstanz

Betreuender Hochschullehrer:
Prof. Dr.-Ing. Franz Baader

Dresden, im Januar 2017

1 Introduction

Ontologies play an important role as a semantic layer for data access in various areas such as the Semantic Web [BHL01], medicine [Rec+94; SCC97], and enterprise applications [Bus03; Ara+08; Kha+15]. They capture the terminology of an application domain and describe domain knowledge in a machine-processable way. Formal ontology languages additionally provide semantics to these specifications. In contrast to standard database systems, systems for *ontology-based data access* (OBDA) thus may apply *logical reasoning* to answer queries; they use the ontological knowledge to infer new information, which is only implicitly given in the data. Moreover, they usually employ the *open-world assumption*, which means that knowledge not stated explicitly in the data or inferred is neither assumed to be true nor false. This faithfully models the real world and also differs from database query answering, which assumes knowledge not present in the data to be false.

All these features make ontologies valuable tools for systems that integrate heterogeneous data sources and need to automatically interpret the data, to support data analysis or to fully-automatedly recognize complex contexts; also multi-agent systems profit from the semantic interoperability. This has been generally recognized and several standardized ontologies have recently been published, especially for domains where heterogeneous data sources are usual or different agents have to communicate seamlessly, such as for sensor networks [Com+12] and robotics and automation [Ont15]. Often, the processed data is changing and thus temporal in that it is associated to specific points in time, and this temporal dimension is critical for analysis or for describing and recognizing real-world contexts. Sensors, for example, produce *streams of data*. Classical ontology-based data access regards the knowledge however only w.r.t. a single moment, which means that information about time is not used for reasoning and thus lost; in particular, the queries generally cannot express temporal aspects.

This work therefore investigates temporal query languages that allow to access temporal data through classical ontologies. In particular, we study the computational complexity of temporal query answering regarding ontologies written in *lightweight description logics*, which are known to allow for efficient reasoning in the atemporal setting and are successfully applied in practice [Rec+94; SCC97; Kha+15]. Our results may thus guide the choice of a query language for temporal OBDA in data-intensive applications that require fast processing.

2 Ontology-Based Data Access

Today, many applications need to process large amounts of heterogeneous *data* growing over time—the famous “big data”. *Data integration* is critical for managing and analyzing such information and demands a common, well-defined vocabulary. Otherwise, misinterpretation may lead to lacking or even wrong consequences.

Ontologies play a fundamental role in this context. In computer science, an ontology can be described as in [Ont15, Introduction]: *“It formally specifies the key concepts, properties, relationships, and axioms of a given domain. Unlike taxonomies, which provide only a set of vocabulary and a single type of relationship between terms, an ontology provides a richer set of relationships, constraints, and rules. In general, ontologies make the relevant knowledge*

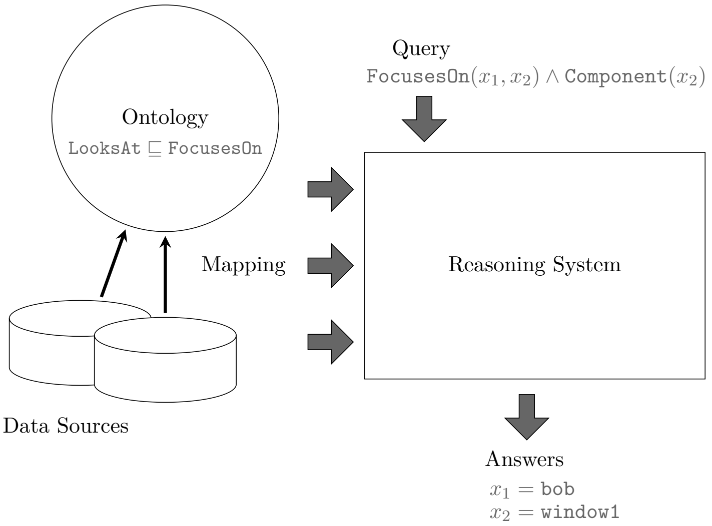

Figure 1: Architecture of ontology-based data access. If the ontological axiom `LooksAt` ⊑ `FocusesOn` ("every tuple in the relation `LooksAt` is also in the relation `FocusesOn`") is taken into account for answering the example query over the data sources (see Figure 2), then a reasoning system outputs the answers depicted.

about a domain explicit in a computer-interpretable format, allowing software to reason over that knowledge to infer new information." In summary, ontologies are formal domain models that provide semantic interoperability and additionally allow for knowledge inference.

The general architecture of ontology-based data access is illustrated in Figure 1. As it is common in data integration, the original data sources are mapped to a global schema—here represented by the ontology—that integrates the sources and allows to access the data using a shared vocabulary while the peculiarities of the sources stay transparent [Len02]; for example, observations of different types of sensors monitoring eye movement (`eye`) or human focus (`foc`) may be mapped to corresponding ontological relations, such as `LooksAt` and `FocusesOn`. Example sources and the mapping (partly) are depicted in more detail in Figure 2. The two first mappings map both relations *Process* and *Window* to the ontological concept `Component` and hence show how distinct sources can be integrated easily. The ontology may, for example, contain a rule as depicted in Figure 1 (in description logic

Process

ID	TYPE	TIME
`p01`	`sys`	10331
`p02`	`vid`	11245

Window

ID	PROC
`window1`	`p02`
`window4`	`p02`

Observation

SENSOR	TYPE	USER	ITEM	TIME
s1	`eye`	`bob`	`window1`	11245
s3	`eye`	`ann`	`book5`	15798

$$\begin{aligned}
\mathit{Process}(x, y, t) &\rightarrow \texttt{Component}(x)\\
\mathit{Window}(x, y) &\rightarrow \texttt{Component}(x)\\
\mathit{Process}(x, y, t) \wedge \mathit{Window}(z, x) &\rightarrow \texttt{PartOf}(z, x)\\
\mathit{Observation}(x, \texttt{foc}, y, z, t) &\rightarrow \texttt{FocusesOn}(y, z)\\
\mathit{Observation}(x, \texttt{eye}, y, z, t) &\rightarrow \texttt{LooksAt}(y, z)\\
\mathit{Process}(x, \texttt{vid}, t) &\rightarrow \texttt{VideoPlayer}(x)
\end{aligned}$$

Figure 2: Data sources and mapping for the example in Figure 1; the variables in the mapping are universally quantified.

notation), stating that someone looking at something focuses on it; this is useful if the system, for some reason, did not receive data from a sensor of type `foc`, directly capturing the focus. Various other applications of ontologies for context recognition are described in [Dar+13; Häh+14]. If the data then is queried through a reasoning system as depicted (i.e., instead of a traditional database system), the ontological knowledge is taken into account by the logical reasoning applied during the answering process. This is commonly referred to as *ontology-based data access* or *ontology-based query answering*[1].

Both aspects of ontological modeling, the formality and the possibility for logical inferencing, are long-standing areas of research in computer science. The importance of formal modeling was recognized early and there are well-established techniques for all kinds of use cases, such as entity-relationship modeling [Che76] for representing database schemas and UML [OMG15] for hard and software artifacts. The search for logics that provide sufficient expressive power and, at the same time, allow for efficient inferencing—alike human reasoning—is performed in the field of artificial intelligence. The usually required efficiency makes the design of such logics challenging and is the reason for the restricted expressiveness of many formalisms. For that reason, the logics are often tailored to certain use cases, and ever new use cases make it an ongoing research.

Particularly user-friendly approaches of ontological knowledge representation emerged with *description logics* in the mid-1980s. Since then, description logics have been studied extensively, and they also represent the logical background of the most prominent ontology language today, the *Web Ontology Language* OWL, a W3C standard [DS04].

3 Lightweight Description Logics as Ontology Languages

Description logics (DLs) are a family of logical formalisms that were originally designed for terminological modeling and *decidable* reasoning, while featuring both sufficient expressivity and readability [Baa+07]. Over time, several further use cases have come to the fore and new DLs tailored to those have been developed. Today, the family comprises *lightweight* DLs, such as $\mathcal{EL}$ [BKM99] and many *DL-Lite* logics [Cal+07; Art+09], which allow for *tractable* reasoning (i.e., reasoning in polynomial time); the prototypical DL $\mathcal{ALC}$ [SS91], which is minimal propositionally complete[2]; and very expressive DLs such as $\mathcal{SROIQ}$ [HKS06], which represents the basis of OWL 2[3]. Note that *DL-Lite* was originally proposed as a particular DL [Cal+04]; the term today however usually refers to a family of DLs comprising logics that have been developed subsequently and provide similar basic features.

DLs generally lie in the two-variable fragment of first-order logic,[4] but have a special, yet intuitive, syntax. A DL allows to model *individual elements*, which represent concrete objects, such as `bob` and `window1`; *concepts*, representing classes of individuals, such as `User` and `Component`; and *roles*, representing (binary) relations between individuals, such as `FocusesOn`. These semantic entities are syntactically described in axioms using *individual*

---

[1] In the following, we usually drop the prefix ontology-based and simply refer to “query answering”. If an ontology is not considered, we use the notion “database query answering”.

[2] A propositional logic is *complete* if every Boolean function can be expressed in a term using propositions that represent the arguments of the function.

[3] The “2” reflects the update of OWL [OWL12].

[4] There are a few exceptions, but these are rarely used today, such as DLs that allow for specifying transitive closures.

names, *concept names*, and *role names*—which, respectively, correspond to constants, unary, and binary predicates in first-order logic.[5] Moreover, complex concept expressions can be constructed using the Boolean operators complement ($\neg$), intersection ($\sqcap$), and union ($\sqcup$); and *role restrictions*. For instance, the expression $\exists\texttt{FocusesOn}.\texttt{Component}$ describes the class of all elements that focus on some component. Operators for constructing role expressions are not so common. Nevertheless, the inverse role operator ($\cdot^{-}$) represents a characteristic feature of *DL-Lite*. For example, applications for which there exists some element that focuses on them can be captured as follows: $\texttt{Application}\sqcap\exists\texttt{FocusesOn}^{-}.\top$,where the concept $\top$ describes the class of all elements. The different DLs are characterized by the syntactic means they provide: the operators for specifying concepts and roles, and the kinds of axioms they allow for.

DL theories are called *knowledge bases* (KBs) and separate the axioms into an *ontology* and an *ABox*. While the ontology contains general domain knowledge, the ABox contains *data* about concrete objects and thus represents a description of (an extract of) the real world. Observe that the ABox can be seen as an instantiation of the global schema described in the previous section (see Figure 1). That is, the DL abstracts from the implementation aspect where the data is actually stored and does not consider the sources to be different from the global view. DL axioms are expressions of two kinds:

- *Assertions*, such as $\texttt{User}(\texttt{bob})$ ("Bob is a user") and $\texttt{LooksAt}(\texttt{bob},\texttt{window1})$ ("Bob looks at an element named `window1`"), occur in ABoxes and describe facts about concrete objects.

- *Inclusions* occur in ontologies and express is-a relations between concepts or roles; for instance: $\texttt{VideoPlayer}\sqsubseteq\texttt{Application}\sqcap\texttt{EnergyIntensive}\sqcap\neg\texttt{SystemCritical}$ ("every video player is an energy-intensive, not system-critical application").

Reasoning over DL knowledge bases originally often concentrated on ontologies and certain *standard reasoning problems*, such as the question whether a concept inclusion (CI) holds in any interpretation. For example, $\mathcal{EL}$ has been applied in terminological reasoning tasks such as the latter for a long time. Only recently, the growing importance of data in practice has lead to an increased interest in *query answering*. The latter usually denotes the task of answering queries over KBs with the goal of retrieving ABox data, and *conjunctive queries* (CQs) currently represent one of the most important query languages in this context. A CQ is a conjunction of first-order atoms where the variables may be existentially quantified and the remaining variables represent the answers to the query. For example, the following CQ $\text{Context}_{\text{Focus}}$ can be used to recognize a complex context, by retrieving all those components x_1 and users x_2 so that x_1 is a subcomponent of some component y_1 which has a part y_2 the user focuses on:

$$\exists y_1,y_2.\texttt{Component}(x_1)\land\texttt{Component}(y_1)\land\texttt{Component}(y_2)\land$$
$$\texttt{PartOf}(x_1,y_1)\land\texttt{PartOf}(y_2,y_1)\land\texttt{FocusesOn}(x_2,y_2).$$

The *DL-Lite* logics have been tailored to conjunctive query answering. This is reflected in the fact that, for many of them, conjunctive queries w.r.t. a KB can be rewritten into first-order queries encoding both the original CQ and the ontological knowledge. This turned

---

[5] The terms "concept" and "role" are generally used for both the syntactic entities, as abbreviations for "concept expression" and "role expression", and the semantic entities.

out to be very efficient since the first-order queries can be represented in SQL and then be evaluated over a standard database containing the ABox data [Cal+17]. For example, if the inclusion $\mathtt{LooksAt} \sqsubseteq \mathtt{FocusesOn}$ is taken into account, then the CQ $\mathrm{Context}_{\mathrm{Focus}}$ is rewritten into a disjunction of CQs which, next to the CQ $\mathrm{Context}_{\mathrm{Focus}}$, contains the following CQ:

$$\begin{aligned}\exists y_1 y_2 . &\mathtt{Component}(x_1) \wedge \mathtt{Component}(y_1) \wedge \mathtt{Component}(y_2) \wedge \\ &\mathtt{PartOf}(x_1, y_1) \wedge \mathtt{PartOf}(y_2, y_1) \wedge \mathtt{LooksAt}(x_2, y_2).\end{aligned}$$

If the rewritten query is used for answering the CQ $\mathrm{Context}_{\mathrm{Focus}}$, then the observations from sensors of type `eye` are also taken into account.

In fact, in several lightweight logics, ontology-based query answering can be rewritten into existing formalisms—though not always into first-order logic. This makes them especially interesting for applications, since mature tools for answering the rewritings often exist already. The practical importance of the lightweight logics is also reflected by the fact that the OWL standard has been complemented by three so-called OWL 2 profiles [Mot+12], which are subsets of OWL 2. Two of them, OWL 2 EL and OWL 2 QL are based on extensions of $\mathcal{EL}$ and a *DL-Lite* logic, respectively.

4 Ontology-Based Temporal Query Answering

The availability and importance of temporal data and ontologies in today's applications motivate our work on querying temporal data through classical ontologies. We investigate *temporal query languages*, which allow to refer to data associated to different moments in time, and regard *ontology-based temporal query answering* as reasoning problem. Thereby, we focus on ontological axioms in *lightweight* logics, which allow for polynomial-time reasoning in the atemporal setting. We specifically regard the DLs $\mathcal{EL}$ and several *DL-Lite* fragments when studying complexity, but extend our results of the last chapter to various other logics.

Observe that temporal extensions of lightweight DLs where temporal operators may be applied to construct ontological concepts have turned out as being surprisingly complex, even undecidable [Art+07]. Nevertheless, research on such formalisms has been going on and identified "islands of tractability" and first-order rewritable formalisms, by restricting the available temporal operators and their applicability [Art+15a; GJK16]. The setting we consider is "easier" since, although we consider temporal data and queries, we do not allow the ontological axioms to contain temporal operators. In particular, decidability of this kind of ontology-based temporal query answering follows in most cases from results for more expressive formalisms [BGL12; BBL15b]. But it was open if the rather high complexities would decrease with lightweight logics. We provide results on the interaction of lightweight DLs and temporal logics and hence complement both strands of research.

The setting we focus on is depicted in Figure 3. The temporal data is represented through a *sequence of logical fact bases*, such as DL ABoxes, each of which contains facts about concrete objects and is associated to a specific point in time. General domain knowledge is described in an ontology and, in contrast to the facts, assumed to hold at all time points. Together, the data and ontology form a *temporal knowledge base* (TKB). Note that we thus can represent a stream of data and, in line with this scenario, consider the queries to be answered over the whole sequence viewed from the *current time point* n ("now"). The *temporal queries* are formed by combining atemporal queries using Boolean operators and operators of *linear temporal logic* (LTL), such as $\Diamond_P$ ("at some time in the past") and $\Diamond_F$

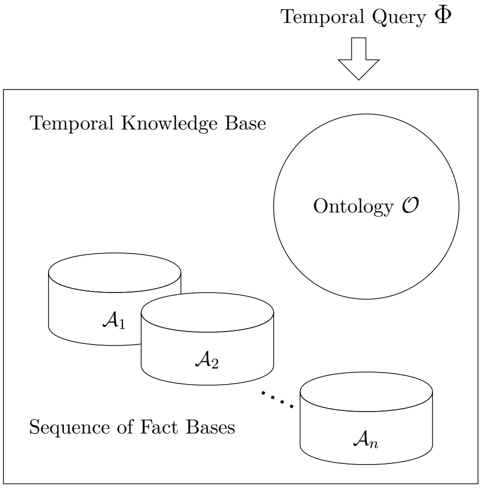

Figure 3: Our setting for temporal OBDA: a temporal query over a temporal knowledge base.

(“at some time in the future”). Large parts of this work focus on *temporal conjunctive queries (TCQs)*, where the atemporal queries are CQs. For instance, a complex context where an energy-intensive application gets out of user focus, which might require a reconfiguration of the system (e.g., by decreasing quality parameters), can be encoded as a TCQ as follows, based on the CQ $\mathrm{Context}_{\mathrm{Focus}}$:

$$\texttt{Application}(x_1) \land \texttt{Running}(x_1) \land \texttt{EnergyIntensive}(x_1) \land \neg\texttt{SystemCritical}(x_1) \land \\ \Diamond_P \left(\mathrm{Context}_{\mathrm{Focus}}(x_1, x_2) \land \Diamond_F \left(\mathrm{Context}_{\mathrm{Focus}}(x_3, x_2) \land \neg\Diamond_F \mathrm{Context}_{\mathrm{Focus}}(x_1, x_2)\right)\right).$$

In natural language, the query describes a situation where, at some time in the past, the user x_2 focused on a component x_1, which is an energy-intensive application, running currently, and not system critical; and, at some time after that, the user has focused on a component x_3, and the user focus then or later never was with x_1 again.

Observe that names such as `Running` and `FocusesOn` are used to describe dynamic knowledge, which may change over time. For describing knowledge that does not do so, certain names are often discerned as being *rigid* [Art+07; BBL15b]; for example, this would be adequate for the role name `PartOf`. We also consider rigid names. It may help to find additional inferences, but usually makes reasoning more complex.

In DLs, the investigation of ontology-based query answering in a temporal setting, targeting data retrieval, and focusing on decidable formalisms has started only in the recent past. Theoretical studies on complexity and rewritability have concentrated on qualitative temporal logics, such as LTL [GK12; Kla13; KM14; Art+15a; BBL15b], and interval-based temporal logics [Art+14; Art+15b]. Also our work has helped to advance the understanding of interactions between temporal queries and DL ontologies [BLT15; BT15b; BT15a]. This foundational work has recently lead to the consideration of metric temporal DLs in [GJO16], where the operators of LTL are annotated with quantitative intervals, such as $\Diamond_F^{[0,5]}$ (“at some time within the next 5 time points”). This is an important feature to describe systems with discrete state changes, and hence data streams.

5 Contributions

The relevance and benefits of ontology-based data access have generally been recognized and are reflected in an increasing number of implementations [CCG10; Ani+12; Kha+16; CMC16; Cal+17]. Due to the amounts of data to be processed and the efficiency requirements of applications, many of the systems focus on ontologies in lightweight logics today. While some of them—to a certain extent—even deal with temporal data already, research on the theoretical side is lacking behind. Temporal query languages that allow to access temporal data through classical ontologies in lightweight DLs have rarely been investigated yet.

The aim of this work is therefore to systematically analyze the interaction between LTL and lightweight DL axioms to obtain (worst-case) complexity results for temporal query answering. To this end, we investigate the complexity of the corresponding decision problems, satisfiability and entailment; we also discern different settings depending on which kinds of names are allowed to be rigid. Further, we develop temporal query languages for which ontology-based query answering is rewritable into existing formalisms. The concrete research questions we focus on can be grouped into three areas:

LTL over lightweight description logic axioms What is the complexity of reasoning regarding temporal queries combining lightweight DL axioms via LTL operators? If necessary, can we find constraints for obtaining good results (i.e., matching those for LTL)? [BT15c; BT15b]

Entailment of temporal conjunctive queries What is the complexity of temporal conjunctive query entailment regarding ontologies in $\mathcal{EL}$, Horn fragments of *DL-Lite*, and more expressive *DL-Lite* logics? Are there logics for which we get tractable or rewritability results? [BT15c; BT15b; BT15a]

Rewritability of temporal query answering How can we combine LTL operators with conjunctive queries to obtain rewritability results for temporal conjunctive query answering in lightweight DLs? Is it possible to extend the results to other temporal queries and logics? [BLT13a; BLT13b; BLT15; THÖ15]

In the following, we describe our results in more detail. Figures 4 and 5 present an overview of our complexity results.

6 A General Approach for Solving Satisfiability

We introduce an abstract temporal query language that combines atemporal queries $\mathcal{QL}$ via the operators of LTL and allows to access temporal data through ontologies, as outlined in Section 4. In particular, these *temporal queries* (TQs) (i.e., more precisely, *temporal* $\mathcal{QL}$ *queries*) generalize existing temporal query languages and, at the same time, provide a framework for the design of new formalisms and general investigations. We then focus on DLs $\mathcal{DL}$ as more concrete ontology languages, and on $\mathcal{DL}$-LTL and TCQs as concrete temporal query languages, and analyze an existing approach for solving the satisfiability problem—and thus also the entailment problem [BGL12; BBL15b]. In a nutshell, the idea of this algorithm is to split the TQ satisfiability problem into a satisfiability problem in propositional LTL and one in (atemporal) DL. The latter is summarized by the notion of *r-satisfiability* (r for “rigid”) and represents the knowledge considered at (possibly exponentially many) time points in an atemporal fashion in order to ensure that rigid information from different time points does not contradict each other. We prove that the direct application of this approach yields only containment in NExpTime (NP) w.r.t. combined (data) complexity,[6] which does not fit the low complexity we usually have with lightweight DLs; regarding entailment, we thus get containment in co-NExpTime (co-NP). For that reason, we propose a new approach based on the original algorithm for solving the satisfiability problem in propositional LTL, which requires only polynomial space in combined complexity [SC85]. The algorithm for LTL iteratively constructs a model $(w_i)_{i\geq 0}$ for a given LTL formula φ: it considers a sequence of exponentially many time points i; iteratively regards each of them; guesses a world w_i, representing the propositions satisfied at time point i; and describes polynomial-space tests ensuring the adequacy of w_i (i.e., regarding the worlds guessed in previous steps) and relying on a polynomial amount of information that is kept and updated during the iteration. We extend that algorithm to iteratively construct a model for a given TQ w.r.t. a TKB. That is, the LTL satisfiability part of the problem is solved in general, independently of $\mathcal{QL}$ and $\mathcal{DL}$; specifically, we use the world w_i guessed in the original algorithm to determine the $\mathcal{QL}$ queries that have to be satisfied at time point i. But our procedure still has to be tailored to the specific problem under consideration (e.g., TCQ entailment in $\mathcal{EL}$) regarding r-satisfiability. In particular, this r-satisfiability testing has to be done as follows for obtaining a concrete, polynomial-space algorithm:

- It can rely on a *polynomial amount* of *additional data* which is guessed in the beginning and kept during the iteration.
- It must be done in *tests that require only polynomial space* and, during the iteration, may consider only one world w_i—different from the tests described in [BGL12; BBL15b], which consider exponentially many different worlds.

The characterizations of r-satisfiability by such a polynomial amount of information and conditions that can be tested by using only polynomial space presents a main contribution of our work.

On the other hand, for several TQs and DLs $\mathcal{DL}$ where we cannot obtain such a polynomial-space algorithm, we apply the few means these lightweight DLs offer for showing hardness of satisfiability w.r.t. NExpTime (NP) or even 2-ExpTime. Indeed, several cases considering rigid symbols have turned out to be NExpTime-hard. In a nutshell, this is the case because, there, interactions are possible: we cannot only use the LTL features to discern exponentially many time points and to nondeterministically choose a specific $\mathcal{QL}$ query at each of them, but we can also apply the DL part to correspondingly discern exponentially many (rigid) concepts, instantiated by different individuals, and thus "save" the LTL choices invariant to time, at the respective of those individuals—via rigid names. In addition, the DL part may constrain the choice—via the ontology. By reducing the 2^{n+1}-bounded domino problem [BGG97, Thm. 6.1.2], we prove that it needs a NExpTime Turing machine to decide satisfiability in this setting.

---

[6] In order to obtain this complexity, the satisfiability of conjunctions of $\mathcal{QL}$ queries and negated $\mathcal{QL}$ queries w.r.t. a KB in $\mathcal{DL}$ has to be decidable nondeterministically in polynomial time.

6.1 LTL over Lightweight Description Logic Axioms

Regarding TQs where the $\mathcal{QL}$ queries are axioms of lightweight DLs, we study the combined complexity of the satisfiability problem. The following example for $\mathcal{EL}$-LTL describes that it always ($\Box_F$) must hold that, if all occupants in a room are sleeping at two consecutive ($\bigcirc_F$) observation moments, then all lights and screens are switched off at the second time of observation:

$$\Box_F \Big((\texttt{RoomOccupant} \sqsubseteq \texttt{Sleeping}) \wedge \bigcirc_F (\texttt{RoomOccupant} \sqsubseteq \texttt{Sleeping}) \to$$
$$\bigcirc_F \, (\texttt{Light} \sqsubseteq \exists \texttt{HasState.SwitchedOff}) \wedge \, \bigcirc_F \, (\texttt{Screen} \sqsubseteq \exists \texttt{HasState.SwitchedOff}) \Big).$$

In particular, we consider DLs $\mathcal{DL}$ meeting a few rather weak requirements: (I) conjunction, $\top$, and either $\bot$ or qualified existential restriction are allowed, and (II) satisfiability of conjunctions of $\mathcal{DL}$ literals can be decided in NP. We show that these conditions are satisfied by the popular DL $\mathcal{EL}$ and many *DL-Lite* fragments. Nevertheless, $\mathcal{DL}$-LTL turns out to be NExpTime-hard if rigid symbols are considered. For the case without rigid symbols, we show containment in PSpace, based on our new approach. For $\mathcal{EL}$ and $\textit{DL-Lite}^{\mathcal{H}}_{horn}$, this PSpace result also holds w.r.t. rigid symbols if the considered CIs are *global* (vs. *local*); global CIs must not be prefixed by arbitrary temporal operators but are assumed to hold always in time. Figure 4 shows that the results for global CIs are interesting compared to the high complexities we have for more expressive DLs.

6.2 Entailment of Temporal Conjunctive Queries

For TCQ entailment in $\mathcal{EL}$, our results (see Figure 5) are similar to those for $\mathcal{EL}$-LTL, in combined complexity: we have a polynomial-space algorithm and co-NExpTime-hardness. However, in contrast to $\mathcal{EL}$-LTL, the former also holds for the case with rigid concept names. This shows that the local CIs allowed in $\mathcal{EL}$-LTL are rather powerful; note that $\mathcal{EL}$-LTL queries without them can be seen as TCQs w.r.t. a global ontology. For data complexity, tractability only holds for the case without rigid symbols.

Regarding TCQ entailment in Horn fragments of *DL-Lite*, including role inclusions (designated by the superscript $\mathcal{H}$), we show results that are considerably better than those for $\mathcal{EL}$, although the DLs are not very different: containment in PSpace and ALogTime w.r.t. com-

				Global CIs			
$\mathcal{DL}$	(i)	(ii)	(iii)	(i)	(ii)	(iii)	
$\textit{TDL-Lite}^{\bigcirc_F}_{krom}$	PSpace	PSpace	PSpace	PSpace	PSpace	PSpace	[Art+07]
$\textit{TDL-Lite}^{\bigcirc_F}_{horn}$	ExpSpace	ExpSpace	ExpSpace	ExpSpace	ExpSpace	ExpSpace	[Art+07]
$\textit{TDL-Lite}_{bool}$	ExpSpace	ExpSpace	ExpSpace	ExpSpace	ExpSpace	ExpSpace	[Art+07]
$\mathcal{EL}$	PSpace	NExpTime	NExpTime	PSpace	PSpace	PSpace	
$\textit{DL-Lite}^{[\,\mid\mathcal{H}]}_{horn}$	PSpace	NExpTime	NExpTime	PSpace	PSpace	PSpace	
$\textit{DL-Lite}^{[\,\mid\mathcal{H}]}_{bool}$	PSpace	NExpTime	NExpTime	PSpace	≤NExpTime	≤NExpTime	
$\mathcal{ALC}$	ExpTime	NExpTime	2-ExpTime	ExpTime	ExpTime	2-ExpTime	[BGL12]
$\mathcal{SHOQ}$	ExpTime	NExpTime	2-ExpTime	ExpTime	≤NExpTime	2-ExpTime	[Lip14; BGL12]

Figure 4: The complexity of satisfiability in $\mathcal{DL}$-LTL w.r.t (i) no rigid names, (ii) rigid concept, and (iii) rigid role names. Our results are highlighted. All except those marked with ≤ (containment) are tight.

	Data Complexity			Combined Complexity		
	(i)	(ii)	(iii)	(i)	(ii)	(iii)
$\textit{DL-Lite}^{[\ \vert\mathcal{H}]}_{[core\vert horn]}$	ALogTime	ALogTime	ALogTime	PSpace ≥ [SC85]	PSpace	PSpace
$\mathcal{EL}$	P ≥ [Cal+06]	co-NP	co-NP	PSpace ≥ [SC85]	PSpace	co-NExpTime
$\mathcal{ALC}$-$\mathcal{SHQ}$ [BBL15b]	co-NP	co-NP	≤ExpTime	ExpTime	co-NExpTime	2-ExpTime
$\textit{DL-Lite}_{[krom\vert bool]}$	co-NP ≥ [Cal+05]	co-NP	≤ExpTime	ExpTime	co-NExpTime	2-ExpTime
$\textit{DL-Lite}^{\mathcal{H}}_{[krom\vert bool]}$	co-NP	co-NP	≤ExpTime	2-ExpTime	2-ExpTime	2-ExpTime ≤ [BBL15a]

Figure 5: The complexity of TCQ entailment w.r.t. (i) no rigid symbols, (ii) rigid concept, and (iii) rigid role names. Our results are highlighted. All complexities except those marked with ≤ (containment) are tight; ≥ (hardness).

bined and data complexity. Although we could not achieve FO rewritability, the latter result is interesting since containment in ALogTime is considered as an indicator for the existence of efficient parallel implementations [AB09, Thm. 6.27]; also note that, in many applications, data complexity better captures resource consumption than combined complexity.

We also consider TCQ entailment in $\textit{DL-Lite}_{krom}$ and $\textit{DL-Lite}_{bool}$, with(out) role inclusions. These DLs are rather expressive, but do not allow for qualified existential restrictions on the left-hand side of CIs. As described above, we have identified this feature as a cause of complexity for TCQ answering. Nevertheless, TCQ entailment has turned out to be as complex as in more expressive DLs, such as $\mathcal{SHQ}$ [BBL15b]. Further, we prove that role inclusions, which allow to express qualified existential restrictions on the right-hand side of CIs, lead to 2-ExpTime-completeness, which is even higher than the results proven for very expressive DLs [BBL15b].

6.3 Rewritability of Temporal Query Answering

We investigate temporal query answering for different temporal query languages based on many different atemporal query languages studied in the literature, and consider various different lightweight logics as ontology languages. In particular, we achieve a generic rewritability result by disallowing negation in the temporal queries.

7 Conclusions

The goal of the thesis was to systematically analyze ontology-based access to temporal data in terms of computational complexity, and rewritability to existing formalisms. We have focused on a temporal query answering scenario that reflects the needs of the applications of today: the temporal queries are based on LTL, one of the most important temporal logics; the ontologies are written in standard lightweight logics; and the data allows to capture data streams. Altogether, our results show that the features we have studied can often be considered “for free”.

- Regarding combined complexity, we have shown that there are many popular DLs $\mathcal{DL}$ for which the problems of $\mathcal{DL}$-LTL satisfiability or TCQ entailment w.r.t. a $\mathcal{DL}$ TKB are in PSpace, even if rigid symbols are considered. This matches the complexity of satisfiability in LTL, which is much less expressive given the fact that ontologies are not considered at all.
- Comparing the data complexity of TCQ entailment to that of CQ entailment, we only get a similar result for $\mathcal{EL}$ in the case without rigid symbols, and for the expressive *DL-Lite* logics without rigid roles[7]; but the TCQs still provide much more expressivity. Moreover, the results for $\textit{DL-Lite}^{\mathcal{H}}_{horn}$ are not much higher than the AC^0-containment given for standard CQ entailment and even hold w.r.t. rigid symbols.
- There exist many rewritable query languages $\mathcal{QL}$ and lightweight logics $\mathcal{DL}$ that satisfy requirements which we show to imply that positive temporal $\mathcal{QL}$ queries w.r.t. $\mathcal{DL}$ TKBs are similarly rewritable.

Regarding the implementation of our algorithms, there are several open questions to be investigated. Our PSpace results rely on the LTL satisfiability algorithm proposed in [SC85]. Practical algorithms solving LTL satisfiability are however usually exponential-time approaches or apply special techniques, such as parallelization, to cope with the non-determinism since the guessing employed in the original algorithm is hardly feasible in applications [Wul+08; Li+13; Li+15]. It is still open which kinds of algorithms are useful in applications of ontology-based temporal query answering, in particular, if rigid symbols are considered. On the other hand, application knowledge discerning rigid symbols in advance could improve performance. It is further open if our algorithms solving temporal query entailment lead to efficient algorithms for query answering. In practice, the latter problem is generally harder to solve. Our rewritability result leads to a different kind of algorithms, based on rewriting. The question is if it can be implemented easily based on existing systems rewriting atemporal queries. The result itself is rather easily obtained from existing rewritability results for the atemporal case. In [BLT15], we describe different algorithms for temporal query answering that rely on such existing approaches, but the implementation is still future work.

References

[AB09] Sanjeev Arora and Boaz Barak. *Computational Complexity - A Modern Approach.* Cambridge University Press, 2009.

[Ani+12] Darko Anicic, Sebastian Rudolph, Paul Fodor, and Nenad Stojanovic. “Stream Reasoning and Complex Event Processing in ETALIS”. In: *Semantic Web Journal* 3.4 (2012), pp. 397–407.

[Ara+08] José Arancón, Luis Polo, Diego Berrueta, François-Marie Lesaffre, Nicolàs Abajo, and Antonio M. Campos. “Ontology-Based Knowledge Management In The Steel Industry”. In: *The Semantic Web: Real-World Applications from Industry.* Ed. by Jorge Cardoso, Martin Hepp, and Miltiadis D. Lytras. Springer-Verlag, 2008, pp. 243–272.

---

[7] Note that the result for the case with rigid roles is not tight.

[Art+07] Alessandro Artale, Roman Kontchakov, Carsten Lutz, Frank Wolter, and Michael Zakharyaschev. “Temporalising Tractable Description Logics”. In: *Proc. of the 14th Int. Symposium on Temporal Representation and Reasoning (TIME’07)*. Ed. by Valentin Goranko and X. Sean Wang. IEEE Press, 2007, pp. 11–22.

[Art+09] Alessandro Artale, Diego Calvanese, Roman Kontchakov, and Michael Zakharyaschev. “The *DL-Lite* Family and Relations”. In: *Journal of Artificial Intelligence Research* 36 (2009), pp. 1–69.

[Art+14] Alessandro Artale, Davide Bresolin, Angelo Montari, Guido Sciavicco, and Vladislav Ryzhikov. “*DL-Lite* and Interval Temporal Logics: A Marriage Proposal”. In: *Proc. of the 21st European Conference on Artificial Intelligence (ECAI’14)*. Ed. by Torsten Schaub. Vol. 263. Frontiers in Artificial Intelligence and Applications. IOS Press, 2014, pp. 957–958.

[Art+15a] Alessandro Artale, Roman Kontchakov, Alisa Kovtunova, Vladislav Ryzhikov, Frank Wolter, and Michael Zakharyaschev. “First-Order Rewritability of Ontology-Mediated Temporal Queries”. In: *Proc. of the 24th Int. Joint Conference on Artificial Intelligence (IJCAI’15)*. Ed. by Qiang Yang and Michael Wooldridge. AAAI Press, 2015, pp. 2706–2712.

[Art+15b] Alessandro Artale, Roman Kontchakov, Vladislav Ryzhikov, and Michael Zakharyaschev. “Tractable Interval Temporal Propositional and Description Logics”. In: *Proc. of the 29th AAAI Conference on Artificial Intelligence (AAAI’15)*. Ed. by Blai Bonet and Sven Koenig. AAAI Press, 2015, pp. 1417–1423.

[Baa+07] Franz Baader, Diego Calvanese, Deborah L. McGuinness, Daniele Nardi, and Peter F. Patel-Schneider, eds. *The Description Logic Handbook: Theory, Implementation, and Applications.* 2nd ed. Cambridge University Press, 2007.

[BBL15a] Franz Baader, Stefan Borgwardt, and Marcel Lippmann. “Temporal Conjunctive Queries in Expressive Description Logics with Transitive Roles”. In: *Proc. of the 28th Australasian Joint Conference on Artificial Intelligence (AI’15)*. Ed. by Bernhard Pfahringer and Jochen Renz. Vol. 9457. Lecture Notes in Artificial Intelligence. Springer-Verlag, 2015, pp. 21–33.

[BBL15b] Franz Baader, Stefan Borgwardt, and Marcel Lippmann. “Temporal Query Entailment in the Description Logic $\mathcal{SHQ}$”. In: *Journal of Web Semantics* 33 (2015), pp. 71–93.

[BGG97] Egon Börger, Erich Grädel, and Yuri Gurevich. *The Classical Decision Problem.* Perspectives in Mathematical Logic. Springer-Verlag, 1997.

[BGL12] Franz Baader, Silvio Ghilardi, and Carsten Lutz. “LTL over Description Logic Axioms”. In: *ACM Transactions on Computational Logic* 13.3 (2012), 21:1–21:32.

[BHL01] Tim Berners-Lee, James Hendler, and Ora Lassila. “The Semantic Web”. In: *Scientific American* (2001), pp. 96–101.

[BKM99] Franz Baader, Ralf Küsters, and Ralf Molitor. “Computing Least Common Subsumers in Description Logics with Existential Restrictions”. In: *Proc. of the 16th Int. Joint Conference on Artificial Intelligence (IJCAI’99)*. Ed. by Thomas Dean. Vol. 1. Morgan Kaufmann, 1999, pp. 96–101.

[BLT13a] Stefan Borgwardt, Marcel Lippmann, and Veronika Thost. “Temporal Query Answering in *DL-Lite*”. In: *Proc. of the 26th Int. Workshop on Description Logics (DL’13)*. Ed. by Thomas Eiter, Birte Glimm, Yevgeny Kazakov, and Markus Krötzsch. Vol. 1014. CEUR Workshop Proceedings. 2013, pp. 80–92.

[BLT13b] 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)*. Ed. by Pascal Fontaine, Christophe Ringeissen, and Renate A. Schmidt. Vol. 8152. Lecture Notes in Computer Science. Springer-Verlag, 2013, pp. 165–180.

[BLT15] Stefan Borgwardt, Marcel Lippmann, and Veronika Thost. “Temporalizing Rewritable Query Languages over Knowledge Bases”. In: *Journal of Web Semantics* 33 (2015), pp. 50–70.

[BT15a] Stefan Borgwardt and Veronika Thost. “Temporal Query Answering in *DL-Lite* with Negation”. In: *Proc. of the 1st Global Conference on Artificial Intelligence, GCAI’15*. Ed. by Georg Gottlob, Geoff Sutcliffe, and Andrei Voronkov. Vol. 26. EPiC Series in Computing. EasyChair, 2015, pp. 51–65.

[BT15b] Stefan Borgwardt and Veronika Thost. “Temporal Query Answering in the Description Logic $\mathcal{EL}$”. In: *Proc. of the 24th Int. Joint Conference on Artificial Intelligence (IJCAI’15)*. Ed. by Qiang Yang and Michael Wooldridge. AAAI Press, 2015, pp. 2819–2825.

[BT15c] Stefan Borgwardt and Veronika Thost. “Temporal Query Answering in the Description Logic $\mathcal{EL}$ (extended abstract)”. In: *Proc. of the 28th Int. Workshop on Description Logics (DL’15)*. Ed. by Diego Calvanese and Boris Konev. Vol. 1350. CEUR Workshop Proceedings. 2015, pp. 83–87.

[Bus03] Christoph Bussler. “The Role of Semantic Web Technology in Enterprise Application Integration”. In: *IEEE Data Engineering Bulletin* 26.4 (2003), pp. 62–68.

[Cal+04] Diego Calvanese, Giuseppe De Giacomo, Maurizio Lenzerini, Riccardo Rosati, and Guido Vetere. “DL-Lite: Practical Reasoning for Rich DLs”. In: *Proc. of the 2004 Int. Workshop on Description Logics (DL’04)*. Ed. by Volker Haarslev and Ralf Möller. Vol. 104. CEUR Workshop Proceedings. 2004, pp. 92–99.

[Cal+05] Diego Calvanese, Giuseppe De Giacomo, Domenico Lemho, Maurizio Lenzerini, and Riccardo Rosati. “*DL-Lite*: Tractable Description Logics for Ontologies”. In: *Proc. of the 20th Nat. Conference on Artificial Intelligence (AAAI’05)*. Ed. by Manuela M. Veloso and Subbaro Kambhampati. AAAI Press, 2005, pp. 602–607.

[Cal+06] Diego Calvanese, Giuseppe De Giacomo, Domenico Lembo, Maurizio Lenzerini, and Riccardo Rosati. “Data Complexity of Query Answering in Description Logics”. In: *Proc. of the 10th Int. Conference of Knowledge Representation and Reasoning (KR’06)*. Ed. by Patrick Doherty, John Mylopoulos, and Christopher Welty. AAAI Press, 2006, pp. 260–270.

[Cal+07] 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”. In: *Journal of Automated Reasoning* 39.3 (2007), pp. 385–429.

[Cal+17] Diego Calvanese, Benjamin Cogrel, Sarah Komla-Ebri, Roman Kontchakov, Davide Lanti, Martin Rezk, Mariano Rodriguez-Muro, and Guohui Xiao. “Ontop: Answering SPARQL Queries over Relational Databases”. In: *Semantic Web* 8.3 (2017), pp. 471–487.

[CCG10] Jean-Paul Calbimonte, Oscar Corcho, and Alasdair J. G. Gray. “Enabling Ontology-based Access to Streaming Data Sources”. In: *Proc. of the 9th Int. Semantic Web Conference (ISWC’10), Part I*. Ed. by Peter F. Patel-Schneider, Yue Pan, Pascal Hitzler, Peter Mika, Lei Zhang, Jeff Z. Pan, Ian Horrocks, and Birte Glimm. Vol. 6496. Lecture Notes in Computer Science. Springer-Verlag, 2010, pp. 96–111.

[Che76] Peter Pin-Shan Chen. “The Entity-Relationship Model—Toward a Unified View of Data”. In: *ACM Transactions on Database Systems* 1.1 (1976), pp. 9–36. ISSN: 0362-5915.

[CMC16] Jean-Paul Calbimonte, José Mora, and Óscar Corcho. “Query Rewriting in RDF Stream Processing”. In: *Proc. of the 13th Int. Semantic Web Conference (ESWC’16)*. Ed. by Harald Sack, Eva Blomqvist, Mathieu d’Aquin, Chiara Ghidini, Simone Paolo Ponzetto, and Christoph Lange. Vol. 9678. Lecture Notes in Computer Science. Springer-Verlag, 2016, pp. 486–502.

[Com+12] Michael Compton, Payam Barnaghi, Luis Bermudez, Raul Garcıa-Castro, Oscar Corcho, Simon Cox, John Graybeal, Manfred Hauswirth, Cory Henson, Arthur Herzog, Vincent Huang, Krzysztof Janowicz, W. David Kelsey, Danh Le Phuoc, Laurent Lefort, Myriam Leggieri, Holger Neuhaus, Andriy Nikolov, Kevin Page, Alexandre Passant, Amit Sheth, and Kerry Taylor. “The SSN Ontology of the W3C Semantic Sensor Network Incubator Group”. In: *Journal of Web Semantics* 17 (2012), pp. 25–32. ISSN: 1570-8268.

[Dar+13] Waltenegus Dargie, Eldora, Julian Mendez, Christoph Möbius, Kateryna Rybina, Veronika Thost, and Anni-Yasmin Turhan. “Situation Recognition for Service Management Systems Using OWL 2 Reasoners”. In: *Proc. of the 10th Workshop on Context Modeling and Reasoning 2013*. IEEE Computer Society, 2013, pp. 31–36.

[DS04] Mike Dean and Guus Schreiber, eds. *OWL Web Ontology Language Reference*. Available at `http://www.w3.org/TR/owl-ref/`. W3C Recommendation, 10 February 2004.

[GJK16] 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 Conference on Artificial Intelligence (IJCAI’16)*. Ed. by Subbarao Kambhampati. AAAI Press, 2016, pp. 1102–1108.

[GJO16] Víctor Gutiérrez-Basulto, Jean Christoph Jung, and Ana Ozaki. “On Metric Temporal Description Logics”. In: *Proc. of the 22nd European Conference on Artificial Intelligence (ECAI’16).* Ed. by Gal A. Kaminka, Maria Fox, Paolo Bouquet, Eyke Hüllermeier, Virginia Dignum, Frank Dignum, and Frank van Harmelen. Vol. 285. Frontiers in Artificial Intelligence and Applications. IOS Press, 2016, pp. 837–845.

[GK12] Víctor Gutiérrez-Basulto and Szymon Klarman. “Towards a Unifying Approach to Representing and Querying Temporal Data in Description Logics”. In: *Proc. of the 6th Int. Conference on Web Reasoning and Rule Systems (RR’12).* Ed. by Markus Krötzsch and Umberto Straccia. Vol. 7497. Lecture Notes in Computer Science. Springer-Verlag, 2012, pp. 90–105.

[Häh+14] Marcus Hähnel, Julian Mendez, Veronika Thost, and Anni-Yasmin Turhan. “Bridging the Application Knowledge Gap”. In: *Proc. of the 13th Workshop on Adaptive and Reflective Middleware (ARM@Middleware’14).* Ed. by Fábio M. Costa and Anders Andersen. ACM, 2014, 3:1–3:6.

[HKS06] Ian Horrocks, Oliver Kutz, and Ulrike Sattler. “The Even More Irresistible $\mathcal{SROIQ}$”. In: *Proc. of the 10th Int. Conference of Knowledge Representation and Reasoning (KR’06).* Ed. by Patrick Doherty, John Mylopoulos, and Christopher Welty. AAAI Press, 2006, pp. 57–67.

[Kha+15] Evgeny Kharlamov, Dag Hovland, Ernesto Jiménez-Ruiz, Davide Lanti, Hallstein Lie, Christoph Pinkel, Martin Rezk, Martin G. Skjæveland, Evgenij Thorstensen, Guohui Xiao, Dmitriy Zheleznyakov, and Ian Horrocks. “Ontology Based Access to Exploration Data at Statoil”. In: *Proc. of the 14th Int. Semantic Web Conference (ISWC’15), Part II.* Ed. by Marcelo Arenas, Óscar Corcho, Elena Simperl, Markus Strohmaier, Mathieu d’Aquin, Kavitha Srinivas, Paul T. Groth, Michel Dumontier, Jeff Heflin, Krishnaprasad Thirunarayan, and Steffen Staab. Vol. 9366. Lecture Notes in Computer Science. Springer-Verlag, 2015, pp. 93–112.

[Kha+16] Evgeny Kharlamov, Yannis Kotidis, Theofilos Mailis, Christian Neuenstadt, Charalampos Nikolaou, Özgür L. Özçep, Christoforos Svingos, Dmitriy Zheleznyakov, Sebastian Brandt, Ian Horrocks, Yannis E. Ioannidis, Steffen Lamparter, and Ralf Möller. “Towards Analytics Aware Ontology Based Access to Static and Streaming Data”. In: *Proc. of the 15th Int. Semantic Web Conference (ISWC’16), Part II.* Ed. by Paul T. Groth, Elena Simperl, Alasdair J. G. Gray, Marta Sabou, Markus Krötzsch, Freddy Lécué, Fabian Flöck, and Yolanda Gil. Vol. 9981. Lecture Notes in Computer Science. Springer-Verlag, 2016, pp. 344–362.

[Kla13] Szymon Klarman. “Practical Querying of Temporal Data via OWL 2 QL and SQL: 2011”. In: *LPAR 2013, 19th Int. Conference on Logic for Programming, Artificial Intelligence and Reasoning, Short papers proceedings.* Ed. by Kenneth L. McMillan, Aart Middeldorp, Geoff Sutcliffe, and Andrei Voronkov. Vol. 26. EPiC Series in Computing. EasyChair, 2013, pp. 52–61.

[KM14] Szymon Klarman and Thomas Meyer. "Querying Temporal Databases via OWL 2 QL". In: *Proc. of the 8th Int. Conference on Web Reasoning and Rule Systems (RR'14)*. Ed. by Roman Kontchakov and Marie-Laure Mugnier. Vol. 8741. Lecture Notes in Computer Science. Springer-Verlag, 2014, pp. 92–107.

[Len02] Maurizio Lenzerini. "Data Integration: A Theoretical Perspective". In: *Proc. of the 21st Symposium on Principles of Database Systems (PODS'02)*. Ed. by Lucian Popa, Serge Abiteboul, and Phokion G. Kolaitis. ACM, 2002, pp. 233–246.

[Li+13] Jianwen Li, Lijun Zhang, Geguang Pu, Moshe Y. Vardi, and Jifeng He. "LTL Satisfiability Checking Revisited". In: *Proc. of the 20th Int. Symposium on Temporal Representation and Reasoning (TIME'13)*. Ed. by César Sánchez, Kristen Brent Venable, and Esteban Zimányi. IEEE Computer Society, 2013, pp. 91–98.

[Li+15] Jianwen Li, Shufang Zhu, Geguang Pu, and Moshe Y. Vardi. "SAT-Based Explicit LTL Reasoning". In: *Proc. of the 11th Int. Haifa Verification Conference (HVC'15)*. Ed. by Nir Piterman. Vol. 9434. Lecture Notes in Computer Science. Springer-Verlag, 2015, pp. 209–224.

[Lip14] Marcel Lippmann. "Temporalised Description Logics for Monitoring Partially Observable Events". 2014. URL: `http://nbn-resolving.de/urn:nbn:de:bsz:14-qucosa-147977`.

[Mot+12] Boris Motik, Bernardo Cuenca Grau, Ian Horrocks, Zhe Wu, Achille Fokoue, and Carsten Lutz, eds. *OWL 2 Web Ontology Language: Profiles (Second Edition)*. Available at `http://www.w3.org/TR/owl2-profiles/`. W3C Recommendation 11 December 2012, 2012.

[OMG15] Object Management Group (OMG), ed. *OMG Unified Modeling Language (OMG UML) Version 2.5*. Available at `http://www.omg.org/spec/UML/2.5/PDF`. March 2015.

[Ont15] Ontologies for Robotics and Automation Working Group. "IEEE Standard Ontologies for Robotics and Automation". In: *IEEE Std 1872-2015* (2015), pp. 1–60.

[OWL12] W3C OWL Working Group. *OWL 2 Web Ontology Language: Document Overview (Second Edition)*. Available at `http://www.w3.org/TR/owl2-overview/`. W3C Recommendation 11 December 2012, 2012.

[Rec+94] Alan Rector, Also Gangemi, Elena Galeazzi, Andrzej J. Glowinski, and Aangelo Rossi-Mori. "The GALEN CORE Model Schemata for Anatomy: Towards a Re-usable Application-Independent Model of Medical Concepts". In: *Proc. of Medical Informatics Europe (MIE94)*. 1994, pp. 186–189.

[SC85] A. Prasad Sistla and Edmund M. Clarke. "The Complexity of Propositional Linear Temporal Logics". In: *Journal of the ACM* 32.3 (1985), pp. 733–749.

[SCC97] Kent A. Spackman, Keith E. Campbell, and Roger A. Côté. "SNOMED RT: A Reference Terminology for Health Care". In: *Proc. of the AMIA Annual Fall Symposium*. AMIA, 1997, pp. 640–644.

[SS91] Manfred Schmidt-Schauß and Gert Smolka. “Attributive Concept Descriptions with Complements”. In: *Journal of Artificial Intelligence* 48.1 (1991), pp. 1–26.

[THÖ15] Veronika Thost, Jan Holste, and Özgür L. Özçep. “On Implementing Temporal Query Answering in *DL-Lite* (extended abstract)”. In: *Proc. of the 28th Int. Workshop on Description Logics (DL'15)*. Ed. by Diego Calvanese and Boris Konev. Vol. 1350. CEUR Workshop Proceedings. 2015, pp. 552–555.

[Wul+08] Martin De Wulf, Laurent Doyen, Nicolas Maquet, and Jean-François Raskin. “Antichains: Alternative Algorithms for LTL Satisfiability and Model-Checking”. In: *Proc. of the 14th Int. Conference on Tools and Algorithms for the Construction and Analysis of Systems (TACAS'08)*. Ed. by C. R. Ramakrishnan and Jakob Rehof. Vol. 4963. Lecture Notes in Computer Science. Springer-Verlag, 2008, pp. 63–77.