Künstliche Intelligenz manuscript No.
(will be inserted by the editor)

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

Veronika Thost

Received: date / Accepted: date

1 Introduction

Ontologies play an important role as a semantic layer for data access in various areas such as the Semantic Web, medicine, and enterprise applications. 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) may thus *infer* additional information, only implicit in the given data, to answer queries. Moreover, they usually employ the *open-world assumption*, which means that knowledge that is not stated explicitly in the data and cannot be inferred is neither assumed to be true nor false. This faithfully models the real world and differs from database query answering, where knowledge not present in the data is assumed 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. This has been generally recognized and several standardized ontologies have recently been published [12,14]. However, often, the processed data is changing and thus temporal in that it is associated to specific points in time, and this temporal dimension may be critical for analysis or for describing and recognizing real-world contexts; e.g., sensors produce *data streams*. Classical OBDA regards however only a single moment [7], which means that information about time cannot be used for reasoning and is thus lost.

In the doctoral dissertation [17], we therefore investigate 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*, such as *DL-Lite* [11,2], which are known to allow for efficient reasoning in the atemporal setting and are successfully applied in practice. We also present a so-called rewritability result for ontology-based temporal query answering, which suggests ways for implementation. In this article, we present an overview of our results. They may guide the choice of a query language for temporal OBDA in data-intensive applications that require fast processing, such as context recognition.

2 Ontology-Based Temporal Query Answering

Consider the following example context: *"a user watches a video, but, after a while, works with a text editor that hides the video window."* In such a situation, the operating system could optimize resource consumption by decreasing quality parameters of the video. For recognizing the context, it has to be encoded into a *temporal*

Veronika Thost
E-mail: veronika.thost@tu-dresden.de

Center for Advancing Electronics Dresden (cfaed)
Institute of Theoretical Computer Science
TU Dresden

Components in user focus in the past, and not visible anymore (now)?

$(\Diamond_P \exists y.\mathtt{User}(y) \land \mathtt{FocusesOn}(y,x) \land \mathtt{Component}(x)) \land \mathtt{NotVisible}(x)$

$$\mathtt{VideoPlayer} \sqsubseteq \mathtt{Application} \sqcap \mathtt{EnergyIntensive} \sqcap \neg\mathtt{SystemCritical} \quad (1)$$

$$\mathtt{Window} \sqsubseteq \mathtt{Component} \quad (2)$$

$$\mathtt{LooksAt} \sqsubseteq \mathtt{FocusesOn} \quad (3)$$

20:11	20:13	20:15
User(bob)	User(bob)	User(bob)
VideoPlayer(a1)		
Window(w1), Component(w1)	Component(w1)	Component(w1)
HasPart(a1, w1)	LooksAt(bob, w1)	NotVisible(w1)

Figure 1: Example OBDA scenario with a temporal query addressing a temporal knowledge base that consists of an ontology and three fact bases. The symbols User and Component are assumed to describe static knowledge, which once given (here at 20:11) holds always in time. The fact Component(w1) is implied by Window(w1) and axiom (2) at 20:11. FocusesOn(bob, w1) is implied at 20:13. The query is answered at 20:15 (“now”) and yields the answer $x = \mathtt{w1}$.

query and answered over data about different points in time (i.e., different system states). Ontologies can augment this approach by providing an abstract, user-friendly interface to the data and by stating general domain knowledge, which can be taken into account during query answering.

The setting we focus on is depicted in Figure 1. The temporal query addresses a *temporal knowledge base* (TKB) that consists of an ontology and fact bases. We specifically focus on *temporal conjunctive queries* (TCQs), which combine standard conjunctive queries (CQs) with operators of linear temporal logic (LTL). That is, the CQs address single time points. The above TCQ combines the CQs $\exists y.\mathtt{User}(y) \land \mathtt{FocusesOn}(y,x) \land \mathtt{Component}(x)$ and $\mathtt{NotVisible}(x)$. It queries for all applications x for which, at some time in the past ($\Diamond_P$) there existed ($\exists$) a user y focusing on them, and ($\land$) that are not visible anymore now—now, since the CQ $\mathtt{NotVisible}(x)$ is not prefixed by a temporal operator. x is called *answer variable.* A *description logic* (DL) ontology may, for example, contain axioms stating that every video player is an application (1) or that everyone looking at something focuses on that object (3). The fact bases collect the data we get from sensors at the respective time points. The query is to be answered w.r.t. the last time point n (“now”) for which data is available.

In many applications, there is static knowledge that is true all over time. For instance, a user will always be a user and never become an application. In order to model such knowledge, we also consider the setting where some symbols, such as User and Component, may be declared to be *rigid.* This increases expressivity but usually makes reasoning more complex. In Figure 1, the fact FocusesOn(bob, w1) and the rigid information Component(w1) are implied by the TKB and, in turn, yield $x = \mathtt{w1}$ as an answer to the first CQ at 20:13 (seen from n, “at some time in the past”). Thus, also the TCQ yields $x = \mathtt{w1}$ as an answer, at 20:15.

Most related works on temporal query answering w.r.t. ontologies in lightweight DLs actually consider temporal extensions of these DLs (see, e.g., [4,5,3,13]). That is, temporal logic operators may also occur within the ontology. However, reasoning in this setting has turned out to be surprisingly complex, even undecidable [4]. Results interesting for practice are therefore usually obtained by strongly restricting the ontology and/or query language. This is not necessary in our setting. Further, our approach has the advantage that it focuses on ontologies in standard DLs; hence, it can be applied together with existing ontologies.

3 Complexity Results

We study the computational complexity of the decision problem related to temporal query answering, query entailment [10,9]; that is, the question whether the query can be deduced from the knowledge in the TKB (considering queries containing no answer variables).[1] We investigate both combined and data complexity. For the former, the size of all the input is taken into consideration; the latter is only influenced by the number of fact bases and facts, the *data*. Data complexity has become important recently, since it well captures many practical applications, where the data is growing over time but the ontology and the queries only rarely change. Above all, our results show that the features we have studied can often be considered "for free":

- Regarding combined complexity, we show that there are many popular DLs for which the TCQ entailment problem w.r.t. a TKB is in PSpace, even if rigid symbols are considered. This matches the complexity of satisfiability in LTL [15], which is much less expressive for there are no ontologies at all.
- Concerning data complexity, the result for the DL $DL\text{-}Lite^{\mathcal{H}}_{horn}$ is not much higher than the one of CQ entailment [2], and it even holds with rigid symbols.

4 A Generic Rewritability Result

Ontology-based temporal query answering can also be considered as a rewritability problem (see also [11]). In general, rewritability of ontology-based query answering for given query languages $\mathcal{QL}$ and $\mathcal{QL}'$, and an ontology language $\mathcal{L}$, means that every $\mathcal{QL}$-query q and ontology $\mathcal{O}$ written in $\mathcal{L}$, can be encoded into a $\mathcal{QL}'$-query q' such that, for all facts $\mathcal{F}$, the answers to q w.r.t. the knowledge base $\langle \mathcal{O}, \mathcal{F} \rangle$ are the same as the answers to q' over $\mathcal{F}$—the data alone. Rewritability is often investigated w.r.t. the target query language $\mathcal{QL}'$ of first-order logic (FOL) since these formulas can be expressed in standard SQL. For example, CQ answering w.r.t. ontologies in $DL\text{-}Lite^{\mathcal{H}}_{horn}$ is FOL-rewritable [2], which means that every ontology-based CQ answering problem can be solved by answering a corresponding SQL query over the facts stored in a standard database, which usually is optimized and provides high performance.

We prove a generic rewritability result for ontology-based temporal query answering [8]; rewritability is defined analogously in the temporal setting, considering a sequence of fact bases and a time-stamped database. The temporal ($\mathcal{QL}$-)queries are formed by combining queries in some atemporal query language $\mathcal{QL}$, such as CQs, with operators of LTL. On the one hand, these queries hence generalize TCQs, but we disallow the negation operator, which may occur in TCQs. Also the ontology language $\mathcal{L}$ is generic. We specifically show that, if $\mathcal{QL}$, $\mathcal{QL}'$, and $\mathcal{L}$ satisfy certain properties, then temporal $\mathcal{QL}$-query answering w.r.t. ontologies in $\mathcal{L}$ can be rewritten into temporal $\mathcal{QL}'$-queries. Most importantly, the properties require that $\mathcal{QL}$-query answering w.r.t. $\mathcal{L}$ ontologies is $\mathcal{QL}'$-rewritable. Since there are many instantiations of $\mathcal{QL}$, $\mathcal{QL}'$, and $\mathcal{L}$, which have been shown to satisfy all our properties, our rewritability result turns out to be quite interesting. For instance, it shows that TCQ answering without negation w.r.t. ontologies in $DL\text{-}Lite^{\mathcal{H}}_{horn}$ can be solved by answering temporal SQL queries over a temporal database.

5 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 scenario that reflects many 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 fact bases allow to capture data streams.

Our results show that there are, indeed, lightweight ontology languages that suit applications that require fast processing. For example, the data complexities we have proven imply that there are efficient parallel algorithms for TCQ answering w.r.t. ontologies written in $DL\text{-}Lite^{\mathcal{H}}_{horn}$.[2] Moreover, if the queries do not contain negation, that problem can even be solved by applying standard temporal database systems. The latter also holds w.r.t. ontologies in $DL\text{-}Lite_{\mathcal{R}}$, the DL underlying OWL 2 QL. For several other *DL-Lite* logics, we however get the same, rather high complexities as for very expressive DLs.

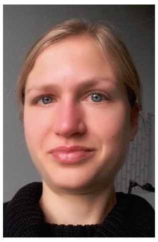
There are many directions for future work. We have recently shown that, even for rather expressive DLs such as $\mathcal{ALCHI}$, TCQ answering is not harder than standard CQ answering in terms of data complexity [16]. We have also started considering metric temporal logic operators such as $\Diamond_{P[0,3]}$ ("at some of the previous three time points"), which allow to capture contexts more precisely [6]. Regarding the implementation of our

[1] For a detailed definition of the problem, we refer to [17].

[2] Containment in the class NC is considered as an indicator for the existence of efficient parallel implementations [1].

algorithms (i.e., the ones yielding the complexity results), it is open which kinds of algorithms are useful in applications, especially, if rigid symbols are considered. On the other hand, application knowledge discerning rigid symbols in advance could improve performance. Our rewritability result leads to a special kind of algorithms, based on rewriting. Maybe implementations can be obtained easily based on existing systems rewriting the atemporal queries. In [8], we describe algorithms for temporal query answering that rely on such existing approaches, but the implementation is still future work.

Acknowledgements I want to thank Stefan Borgwardt for his considerable contributions. This work was supported by the German Research Foundation (DFG) in CRC 912.



Veronika Thost received a diploma in computer science from TU Dresden in 2012. Since then, she is a reseacher at the Institute of Theoretical Computer Science at TU Dresden. She completed her PhD in 2017. Her current work focuses on reasoning over knowledge graphs.

References

1. Arora, S., Barak, B.: Computational Complexity–A Modern Approach. Cambridge University Press (2009)
2. Artale, A., Calvanese, D., Kontchakov, R., Zakharyaschev, M.: The *DL-Lite* family and relations. Journal of Artificial Intelligence Research **36**, 1–69 (2009)
3. Artale, A., Kontchakov, R., Kovtunova, A., Ryzhikov, V., Wolter, F., Zakharyaschev, M.: First-order rewritability of ontology-mediated temporal queries. In: Q. Yang, M. Wooldridge (eds.) Proc. of the 24th Int. Joint Conference on Artificial Intelligence (IJCAI'15), pp. 2706–2712. AAAI Press (2015)
4. Artale, A., Kontchakov, R., Lutz, C., Wolter, F., Zakharyaschev, M.: Temporalising tractable description logics. In: V. Goranko, X.S. Wang (eds.) Proc. of the 14th Int. Symposium on Temporal Representation and Reasoning (TIME'07), pp. 11–22. IEEE Press (2007)
5. Artale, A., Kontchakov, R., Ryzhikov, V., Zakharyaschev, M.: A cookbook for temporal conceptual data modelling with description logics. ACM Transactions on Computational Logic **15**(3), 25 (2014)
6. Baader, F., Borgwardt, S., Koopmann, P., Ozaki, A., Thost, V.: Metric temporal description logics with interval-rigid names. In: Proc. of the 11th Int. Symposium on Frontiers of Combining Systems (FroCoS'17) (2017)
7. Baader, F., Calvanese, D., McGuinness, D.L., Nardi, D., Patel-Schneider, P.F. (eds.): The Description Logic Handbook: Theory, Implementation, and Applications, 2 edn. Cambridge University Press (2007)
8. Borgwardt, S., Lippmann, M., Thost, V.: Temporalizing rewritable query languages over knowledge bases. Journal of Web Semantics **33**, 50–70 (2015)
9. Borgwardt, S., Thost, V.: Temporal query answering in *DL-Lite* with negation. In: G. Gottlob, G. Sutcliffe, A. Voronkov (eds.) Proc. of the 1st Global Conference on Artificial Intelligence, GCAI'15, *EPiC Series in Computing*, vol. 26, pp. 51–65. EasyChair (2015)
10. Borgwardt, S., Thost, V.: Temporal query answering in the description logic $\mathcal{EL}$. In: Q. Yang, M. Wooldridge (eds.) Proc. of the 24th Int. Joint Conference on Artificial Intelligence (IJCAI'15), pp. 2819–2825. AAAI Press (2015)
11. Calvanese, D., De Giacomo, G., Lembo, D., Lenzerini, M., Rosati, R.: Tractable reasoning and efficient query answering in description logics: The *DL-Lite* family. Journal of Automated Reasoning **39**(3), 385–429 (2007)
12. Compton, M., Barnaghi, P., Bermudez, L., Garcıa-Castro, R., Corcho, O., Cox, S., Graybeal, J., Hauswirth, M., Henson, C., Herzog, A., Huang, V., Janowicz, K., Kelsey, W.D., Phuoc, D.L., Lefort, L., Leggieri, M., Neuhaus, H., Nikolov, A., Page, K., Passant, A., Sheth, A., Taylor, K.: The SSN ontology of the W3C Semantic Sensor Network Incubator Group. Journal of Web Semantics **17**, 25–32 (2012)
13. Gutiérrez-Basulto, V., Jung, J.C., Kontchakov, R.: Temporalized $\mathcal{EL}$ ontologies for accessing temporal data: Complexity of atomic queries. In: S. Kambhampati (ed.) Proc. of the 25th Int. Joint Conference on Artificial Intelligence (IJCAI'16), pp. 1102–1108. AAAI Press (2016)
14. Ontologies for Robotics and Automation Working Group: Ieee standard ontologies for robotics and automation. IEEE Std 1872-2015 pp. 1–60 (2015)
15. Sistla, A.P., Clarke, E.M.: The complexity of propositional linear temporal logics. Journal of the ACM **32**(3), 733–749 (1985)
16. Thost, V.: News on temporal conjunctive queries. In: Proceedings of the 1st Web Stream Processing Workshop (2017). To appear.
17. Thost, V.: Using ontology-based data access to enable context recognition in the presence of incomplete information. Ph.D. thesis, TU Dresden (2017). Available at http://nbn-resolving.de/urn:nbn:de:bsz:14-qucosa-227633