Ontology-Enriched Data Management with Partially Complete Data

Sanja Pavlović

Institute of Logic and Computation, TU Wien, Austria

Abstract. As fragments of first-order logic, description logics (DLs) make the assumption that what is not known to be true is unknown (open-world assumption). This is adequate when dealing with inherently incomplete data. However, if the data is assumed to be complete, it is more appropriate to employ the closed-world assumption (CWA) stating that what is not known must be false. Real world applications often require that incomplete parts of the data interact with the parts known to be complete, which calls for formalisms that simultaneously support both of these assumptions. In DLs this can be done by specifying which predicates should be interpreted under the closed-world view. The goal of this paper is to outline the research questions related to DLs with partial CWA that we plan to investigate. Our primary focus will be on characterizing computational complexity and relative expressiveness of ontology-mediated queries in the presence of closed predicates, but we also plan to investigate the interactions between closed-predicates and number restrictions. Finally, we develop formalisms based on DLs with CWA to reason about actions and change.

CEUR-WS.org/Vol-2438/paper2.pdf

1 Introduction and Motivation

The data management paradigm termed *ontology-based data access* (*OBDA*) [22] has received a lot of attention from the scientific community. The goal of OBDA is to allow non-expert users to query multiple heterogeneous and possibly incomplete data sources in an easy way. In this approach, the structure of the underlying data is hidden from the users. They are instead presented with an *ontology* that defines a high-level conceptual view of the application domain and provides background knowledge about it in the form of a logical theory. The users can then use the vocabulary of the ontology to pose so called *ontology-mediated queries (OMQs)* that are answered over the data integrated from different sources and supplemented with the facts inferred using the available background knowledge.

Description logics (DLs) are undoubtedly one of the most popular family of formalisms used for expressing ontologies. They enable us to model things using constants, referred to as *individuals*, and unary and binary predicate symbols, called *concept names* and *role names*, respectively. DL knowledge bases consist

Copyright © 2019 for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

of a *a TBox* containing *terminological axioms* that specify relationships between concepts and roles, and an *ABox* containing facts that assert participation of certain individuals in concepts/roles. The differences between individual logics come from allowing or disallowing certain shapes of terminological axioms or certain constructors used for building complex concepts/roles. Regardless, most DLs can be seen as decidable fragments of first-order logic and as such they employ the *open-world assumption (OWA)*. Intuitively, this means that if an assertion is not known to be true or false then its truth value is simply unknown. As a result, when answering queries we have to consider both the world in which this particular piece of information is true and the one in which it is false. This kind of assumption is appropriate when the data that is assumed to be incomplete. However, in traditional database systems, we often assume to have all relevant information. In this setting it is more appropriate to employ the *closed-world assumption (CWA)* which states that what is not known to be true must be false.

Real world applications often require a mix of OWA and CWA. Suppose we want to query a list of cities with a music festival this summer that we found on the web in conjunction with a database containing information about direct flights from/to our city. Given a city that is not on our list, we should not assume that there will be no music festival there, after all it could be that our list is incomplete. Hence, this part of our data requires us to employ the OWA. However, if we are looking for a direct flight to some city and there is no information in our database about such a flight, we should assume that no direct flight exists, i.e., we should apply the CWA. This clearly illustrates the need for formalisms that offer a more flexible support for open-world and closed-world reasoning.

Various approaches have been proposed for combining OWA and CWA in DLs [8,9,2,25], most prominently those based on *closed predicates* [26,16]. We plan to contribute to this area of research by studying the effects that adopting CWA has on reasoning in DLs in terms of computational complexity and added expressiveness as well as by developing formalisms based on DLs with CWA for reasoning about actions and change. Our objectives can be summarized as follows:

1. Characterizing data complexity of query answering in expressive DLs with closed predicates: It has been shown that closed predicates make reasoning harder in some cases [11,16,19] but exact computational implications are still unknown for many important DLs. This is especially for data complexity of languages that combine expressive DLs with closed predicates.
2. Characterizing relative expressiveness of OMQs in the presence of closed predicates compared to classical query languages: We are particularly interested in providing translations from OMQs with closed predicates to non-monotonic extensions of Datalog. Our goal is to advance the state-of-the-art by also considering navigational queries in addition to standard instance and (unions) of conjunctive queries ((U)CQs), i.e., FO formulas with only existentially quantified variables, conjunction and disjunction.
3. Understanding interactions between closed predicates and number restrictions: Regarding some predicates as closed can result in certain open predicates

having only extensions of bounded size. We are interested in coming up with algorithms that identify such predicates.
4. Developing formalisms that use DLs with closed predicates for reasoning about actions and change.

In Section 2 we describe the state-of-the-art of research closely related to the topic of this thesis. Section 3 outlines the concrete problems that we plan to investigate and Section 4 describes the results that we have obtained so far.

2 State-of-the-Art

Description Logics with Closed Predicates. Allowing some predicates to be *closed* is a prominent way of supporting partial CWA in DLs. One of the earliest such approaches proposed replacing ABoxes with *DBoxes* [26]. Syntactically, both ABoxes and DBoxes are sets of assertions, however there is a semantic difference between the two: a knowledge base with a DBox $\mathcal{D}$ requires its models to interpret every concept and role name occurring in $\mathcal{D}$ exactly as given in $\mathcal{D}$. These predicates are thus considered closed – their extensions are fixed and must be the same in every model of the knowledge base. In contrast, the rest of the predicates are open and their extensions might vary among interpretations. A generalization of this approach was presented in [16] which does not interpret all predicates in the DBox in this way but allows the user to explicitly specify a subset of the signature that is considered closed.

Naturally, closed predicates have an effect on reasoning. In ontology-mediated query answering (OMQA) we are usually concerned with *certain answers*, i.e., tuples of individuals that are answers to the query in *every* model of the knowledge base. However, if we have closed predicates, we are no longer interested in arbitrary models but only in those that "obey" the closed predicates in the way we described above. This can alter our set of certain answers and introduce non-monotonicity.

Computational complexity of standard OMQA is very well understood. There is a wide range of complexity results in the literature that cover many different DLs and query languages using various techniques. For lightweight DLs like the members of the DL-Lite and $\mathcal{EL}$ families, answering of conjunctive queries is tractable in data complexity but not in combined complexity (see e.g. a recent survey [4]). For expressive DLs that extend $\mathcal{ALC}$ answering of conjunctive queries is usually coNP-complete in terms of data-complexity. The same task is 2ExpTime-complete in combined complexity for some extensions of $\mathcal{ALC}$ [14,10,19].

Regarding the complexity of query answering in the presence of closed predicates, the picture is not as detailed but some initial work has been done. For example, [11] shows that closed predicates make query answering in $\mathsf{DL\text{-}Lite}_{\mathcal{F}}$ intractable in data complexity. This was expanded upon in [16] and some results on combined complexity can be found in [19]. However, these works mainly focus on lightweight DLs. For expressive DLs with closed predicates, it was recently shown that evaluation of UCQs in $\mathcal{ALCHI}$ with closed predicates is in coNP in data complexity [18]. To the best of our knowledge, no other results on data complexity of query answering in expressive DLs with closed predicates are known.

Description Logics and Navigational Queries. Since DLs only allow the use of unary and binary predicate symbols, models of DL knowledge bases can be seen as a graphs whose nodes represent domain elements labeled with concept names and edges represent relationships between these elements labeled with role names. In this setting, it makes sense to consider query languages that can navigate these graphs and answer reachability questions. Such languages would allow us to find, e.g., all cities with a music festival that are reachable from our city by plane, either directly, or with one or more layovers. In general, this kind of questions cannot be answered with traditional query languages used in DLs like (unions of) conjunctive queries. A proposed solution is to consider *(two-way) regular path queries ((2)RPQs)* [7], which are the base of navigational query languages used for semi-structured data like SPARQL 1.1 and XPath. Intuitively, RPQs allow us to select nodes that are connected by a directed path whose label belongs to the specified regular language, while 2RPQs give us the possibility traverse the edges in both directions.

(2)RPQs and their extensions like *conjunctive (two-way) regular path queries (C(2)RPQs)* and *nested regular path queries (NRPQs)* have been studied also in the context of OMQA [6,3], however only in the case without closed predicates.

Rewriting OMQs into FO/Datalog. Reducing ontology-mediated query answering to traditional problems like evaluating FO queries over relational data is a popular area of research. This is done by finding rewritings that given any OMQ $Q = (\mathcal{T}, q)$ with a TBox $\mathcal{T}$ in some DL and a query q in some query language, define a new FO query q' that when evaluated over $\mathcal{A}$ produces exactly whose answers that are certain answers Q, for any ABox $\mathcal{A}$. If this is possible, we say that this DL is *first-order rewritable* for the selected query language.

Finding such rewritings comes with many benefits: they allow the reuse the existing technology that is highly optimized and backed up by decades of research, they help us compare the expressive powers of the query languages, and in some cases allow us to transfer known results like complexity bounds (see e.g., [5])

Unfortunately, not all DLs admit FO rewritings. In such cases, the alternative is to go for translations into variants of Datalog One of the first such approaches which reduces reasoning in the expressive DL $\mathcal{SHIQ}$ to reasoning in disjunctive Datalog in exponential time was presented in [12]. Regarding polynomial translation of expressive DLs into Datalog, the earliest such rewriting was presented in [21]. Recently, a new rewriting technique for ontology-mediated instance queries in $\mathcal{ALCHOI}$ was introduced [1] that also takes into consideration closed predicates. This approach is different from previously proposed approaches in that it is based on game-theoretic characterization of query answering.

Description Logics and Non-Monotonic Rules. Hybrid languages that couple description logic ontologies with non-monotonic rules represent another way of accommodating partial CWA in DLs. Differences among individual formalisms arise from the kind of interactions that are allowed between the ontology and the rule component. For example, one of the first hybrid formalisms is *r-hybrid* [24]. In this approach, a knowledge base is a pair $\mathcal{H} = (\mathcal{T}, \mathcal{P})$, where $\mathcal{T}$ is an FO

theory/DL ontology and $\mathcal{P}$ is a disjunctive logic program with negation. The non-monotonic semantics of r-hybrid is intuitively given as follows: all predicates occurring in the ontology are considered to be open. That is, in a model $\mathcal{I}$ of $\mathcal{H}$ these predicates can have arbitrary extensions as long as $\mathcal{I}$ still satisfies $\mathcal{T}$ (in a classical sense). The remainder of the predicates occurring in the rules are considered closed and their extensions cannot be arbitrary – they must be justified by the program. Recently, an extension of this framework was introduced [2] which allows closed predicates to also occur in the ontology component by enabling us to explicitly say which part of the signature is considered closed. A different approach to coupling ontologies and rules are *dl-programs* [9] which give standard answer set programs the ability to pose queries to a DL knowledge bases with the possibility of (temporarily) modifying the ABox before querying.

The main problem of combining DL ontologies with rules is that it often leads to the loss of decidability [13]. To combat this issue, the usual approach is to introduce a safeness condition that ensures that variables in the program range only over a finite number of constants. Different notions of safeness are available in the literature (see e.g., [24,23,2]).

3 Research Questions and Goals

We next summarize the research questions that we plan to tackle in this thesis:

1. **Studying Relative Expressiveness of OMQs with Closed Predicates**
   In the previous section we motivated the importance of rewritings of OMQs into well-established query languages. Unfortunately, only very few rewritability results are available for OMQs with closed predicates: the results from [17] showing that quantifier free UCQs in $\mathsf{DL\text{-}Lite}_{\mathcal{R}}$ with closed predicates are FO rewritable, the one in [26] showing how to rewrite instance queries in $\mathcal{ALC}$ with closed predicates into FO queries (when possible) and the recent result in [1] showing that instance queries in $\mathcal{ALCHOI}$ with closed predicates can be rewritten into a variation of Datalog. The last result differs significantly from standard rewriting approaches in that it targets a *non-monotonic* extension of Datalog (Datalog with negation under the stable model semantics), it is not based on resolution and, above all, it is polynomial. This approach is thus an excellent starting point of our investigation during which we hope to provide a clear picture of how the expressive power of OMQs in the presence of closed predicates compares to classical query languages. We plan to extend the approach in [1] to cover larger classes of OMQs as well as to come up with new rewriting techniques for reducing query answering in richer logics, like those with counting quantifiers, to known query languages.
   Our focus will be on instance queries in expressive description logics – most importantly $\mathcal{ALCHOIQ}$, as well as on RPQs and their extensions in both lightweight and expressive DLs with closed predicates. As our target language we also choose Datalog with negation under the stable model semantics, since it is more expressive than FO, it offers a natural support for recursion which

lies in the heart of RPQs, and it is non-monotonic. This last property is important as OMQs with closed predicates have a non-monotonic nature.
We are mainly interested in finding polynomial rewritings. To obtain a more detailed picture, we will try to find smallest fragments of Datalog that can capture the considered query language. We will also try to provide bidirectional translations showing that identified fragments of Datalog can be translated into the originally considered query language, which will establish the two languages possess the same expressive power. Finally, for the cases in which no (polynomial) rewritings seems to exist, which happens if the complexity of the query language is higher than the target language, we will find restricted fragments for which it is still possible to provide translations.

2. **Characterizing Data Complexity of OMQA with Closed Predicates**
Since closed predicates can be simulated with the help of nominals [26], the combined complexity of answering OMQs in the presence of closed predicates is known for some expressive description logics.
However, there are virtually no results on the data complexity of the same task. We plan to shed some light on this issue. Our primary goal here will be to characterize data complexity of $\mathcal{ALCHOIF}$ and $\mathcal{ALCHOIQ}$, which is strongly intertwined with our previous goal of providing rewritings for OMQs with closed predicates. Namely, we hope to be able to provide rewritings of instance queries mediated by ontologies expressed in these logics into Datalog with stable negation which will allow us the transfer of data complexity bounds of our target language. These translations will be based on integer programming techniques that are often used in the literature to develop algorithms for reasoning in these DLs [15].

3. **Investigating Interactions of Closed Predicates with Counting**
Let us begin with an example of the possible interaction between closed predicates and number restrictions: Assume we have a DL TBox $\mathcal{T}$ stating that each employee of a company can take part in at most 5 projects, and that all projects have at least one employee: $\mathsf{Empl} \sqsubseteq \leq 5\ \mathsf{assgdTo}.\mathsf{Proj}$ and $\mathsf{Proj} \sqsubseteq \exists \mathsf{assgdTo}^{-}.\mathsf{Empl}$. Further, assume Empl is a closed predicate and we are given an ABox $\mathcal{A}$ that contains a list of all n employees. Then we can easily infer that there are at most $5n$ projects. We may not know exactly which projects these are, but in any model of the knowledge base $(\mathcal{T}, \{\mathsf{Empl}\}, \mathcal{A})$ the extension of Proj will contain at most 5 elements, i.e., Proj is in a way *bounded* by the TBox and the closed predicates.
This simple observation made us wonder what are the conditions that need to be fulfilled in order for a concept or a role to be bounded in the presence of closed predicates. In particular, we want to develop algorithms to solve the following reasoning task for DL ontologies expressed in logics with number restrictions and closed predicates: given a TBox $\mathcal{T}$, a set of closed predicates Σ and a concept or a role P, is P bounded by $\mathcal{T}$ and Σ? We expect to be able to solve this problem by once again relying on integer programming techniques.

4. **Combining DLs and Rules to Reason About Actions and Change**
A slightly different route we want to take is to come up with formalisms that

combine description logics and Datalog rules with non-monotonic negation to provide reasoning support for systems that should react correctly in all possible situations. In this setting, a DL ontology describes the situations that the system might face, and rules are used to react to these situations. In such a formalism, we can reduce reasoning tasks like ensuring that the system always has a correct reaction to common reasoning problems for hybrid knowledge bases such as consistency checking. We believe that such formalisms will be not only of theoretical interest, but also useful in practice.

4 Results and Conclusion

In this paper we presented the research questions we plan to tackle in this thesis. Our high-level goal is to better understand the effects that the partial CWA has on reasoning in DLs and how DLs with CWA can be used to solve some common practical problems. Regarding our current results, we introduced within the scope of our last research goal from the previous section a new hybrid language called *resilient logic programs (RLPs)* [20]. An RLP couples a Datalog program $\mathcal{P}$ possibly containing negation as failure with a first-order theory (DL ontology) $\mathcal{T}$ and additionally defines a partitioning of the predicates in $\mathcal{P}$ and $\mathcal{T}$ into three sets: the set of output predicates, the set of open predicates, and the set of response predicates. The semantics is then defined as follows: the two components need to agree on a structure I over the output signature, so that no matter how I is extended into a model of $\mathcal{T}$ (by interpreting the open-world predicates), the program $\mathcal{P}$ can find a suitable response J such that all facts in I and J are justified by $\mathcal{P}$. Intuitively, this means that the output and the response predicates are in a way interpreted as closed predicates. RLPs are suitable in situations where we need to come up with robust settings for some system that allow us to successfully process all products/situations that can come in many possible configurations. To make RLPs decidable, we introduce a novel safeness condition that relies on bounded concepts described in item 2. of the previous section. Regarding relative expressiveness of RLPs, we showed that we can capture disjunctive Datalog with negation as failure and we have identified one fragment of RLPs that can be embedded back into this variation of Datalog.

Acknowledgements This thesis is supervised by Magdalena Ortiz and Mantas Šimkus and it is funded by the FWF projects P30360, P30873 and W1255.

References

1. S. Ahmetaj, M. Ortiz, and M. Šimkus. Polynomial datalog rewritings for expressive description logics with closed predicates. In *Proc. of IJCAI 2016*, pages 878–885.
2. L. Bajraktari, M. Ortiz, and M. Šimkus. Combining rules and ontologies into Clopen knowledge bases. In *Proc. of AAAI 2018*, pages 1728–1735.
3. M. Bienvenu, D. Calvanese, M. Ortiz, and M. Šimkus. Nested regular path queries in description logics. In *Proc. of KR 2014*. AAAI Press.

4. M. Bienvenu and M. Ortiz. Ontology-mediated query answering with data-tractable description logics. In *Proc. of Reasoning Web Summer School 2015*, volume 9203 of *LNCS*, pages 218–307. Springer, 2015.
5. D. Calvanese, G. De Giacomo, D. Lembo, M. Lenzerini, and R. Rosati. Tractable reasoning and efficient query answering in description logics: The *DL-Lite* family. *J. Autom. Reasoning*, 39(3):385–429, 2007.
6. D. Calvanese, T. Eiter, and M. Ortiz. Regular path queries in expressive description logics with nominals. In *Proc. of IJCAI 2009*, pages 714–720.
7. I. F. Cruz, A. O. Mendelzon, and P. T. Wood. A graphical query language supporting recursion. In *Proc. of the ACM Special Interest Group on Management of Data 1987 Annual Conference*, pages 323–330, 1987.
8. F. M. Donini, D. Nardi, and R. Rosati. Description logics of minimal knowledge and negation as failure. *ACM Trans. Comput. Log.*, 3(2):177–225, 2002.
9. T. Eiter, G. Ianni, T. Lukasiewicz, R. Schindlauer, and H. Tompits. Combining answer set programming with description logics for the semantic web. *Artif. Intell.*, 172(12-13):1495–1539, 2008.
10. T. Eiter, C. Lutz, M. Ortiz, and M. Šimkus. Query answering in description logics with transitive roles. In *Proc. of IJCAI 2009*, pages 759–764.
11. E. Franconi, Y. A. Ibáñez-García, and I. Seylan. Query answering with DBoxes is hard. *Electr. Notes Theor. Comput. Sci.*, 278:71–84, 2011.
12. U. Hustadt, B. Motik, and U. Sattler. Reasoning in description logics by a reduction to disjunctive datalog. *J. Autom. Reasoning*, 39(3):351–384, 2007.
13. A. Y. Levy and M. Rousset. Combining horn rules and description logics in CARIN. *Artif. Intell.*, 104(1-2):165–209, 1998.
14. C. Lutz. The complexity of conjunctive query answering in expressive description logics. In *Proc. of IJCAR 2008*, volume 5195 of *LNCS*, pages 179–193. Springer.
15. C. Lutz, U. Sattler, and L. Tendera. The complexity of finite model reasoning in description logics. *Inf. Comput.*, 199(1-2):132–171, 2005.
16. C. Lutz, I. Seylan, and F. Wolter. Ontology-based data access with closed predicates is inherently intractable(sometimes). In *Proc. of IJCAI 2013*.
17. C. Lutz, I. Seylan, and F. Wolter. Ontology-mediated queries with closed predicates. In *Proc. of IJCAI 2015*, pages 3120–3126. AAAI Press.
18. C. Lutz, I. Seylan, and F. Wolter. The data complexity of ontology-mediated queries with closed predicates. *CoRR*, abs/1809.00134, 2018.
19. N. Ngo, M. Ortiz, and M. Šimkus. Closed predicates in description logics: Results on combined complexity. In *Proc. of KR 2016*.
20. M. Ortiz, S. Pavlović, and M. Šimkus. Answer set programs challenged by ontologies. In *Proc. of the 32nd International Workshop on Description Logics*, 2019.
21. M. Ortiz, S. Rudolph, and M. Šimkus. Worst-case optimal reasoning for the Horn-DL fragments of OWL 1 and 2. In *Proc. of KR 2010*. AAAI Press.
22. A. Poggi, D. Lembo, D. Calvanese, G. De Giacomo, M. Lenzerini, and R. Rosati. Linking data to ontologies. *J. Data Semantics*, 10:133–173, 2008.
23. R. Rosati. DL+log: Tight integration of description logics and disjunctive datalog. In *Proc. of KR 2006*, pages 68–78. AAAI Press.
24. R. Rosati. On the decidability and complexity of integrating ontologies and rules. *J. Web Sem.*, 3(1):61–73, 2005.
25. K. Sengupta, A. A. Krisnadhi, and P. Hitzler. Local closed world semantics: Grounded circumscription for OWL. In *Proc. of ISWC 2011*. Springer, 2011.
26. I. Seylan, E. Franconi, and J. de Bruijn. Effective query rewriting with ontologies over dboxes. In *Proc. of IJCAI 2009*, pages 923–925.