DaRLing: a Datalog OWL 2 RL Rewriter

A. Fiorentino, M. Manna and J. Zangari

University of Calabria, Rende, Italy
lastname@mat.unical.it

Ontology-mediated query answering (OMQA) is an emerging paradigm at the basis of many semantic-centric applications. In this setting, a conjunctive query has to be evaluated against a *knowledge base* consisting of an extensional *database* paired with an *ontology*, which provides a semantic conceptual view of the data. Among the formalisms that are capable to express such a conceptual layer, the *Web Ontology Language* OWL is certainly the most popular one.

OWL is a very powerful formalism. But its unrestricted usage makes reasoning undecidable already in case of very simple tasks such as fact entailment. Hence, expressive yet decidable fragments have been identified. Among them, we focus on the one called OWL 2 RL [1]. From the knowledge representation point of view, OWL 2 RL enables scalable reasoning without scarifying too much the expressiveness. Indeed, it supports all RDFS datatypes and provides a rich variety of semantic constructors, such as: *inverseOf*, *transitiveProperty*, *reflexiveProperty*, *equivalentClass*, *disjointWith*, *unionOf*, *minCardinality*, *allValuesFrom*, *someValuesFrom*, and *sameAs* – among others. But the simple fact of allowing *someValuesFrom* only in the left-hand-side of an axiom guarantees that conjunctive query answering can be performed in polynomial time in data complexity (when the OMQ is considered fixed) and in nondeterministic polynomial time in the general case (the latter being exactly the same computational complexity of evaluating a single conjunctive query over a relational database).

Although a number of important Web semantic resources – such as DBpedia [1] and FOAF [2] – trivially fall in OWL 2 RL, only a few systems have been designed and implemented in this setting. None of them, however, fully satisfy all the following desiderata:

(*i*) being freely available and regularly maintained;
(*ii*) supporting query answering and SPARQL queries;
(*iii*) properly applying the `owl:sameAs` property without adopting the unique name assumption;
(*iv*) dealing with concrete datatypes.

To fill this gap, we conceived *DaRLing* [2], an open-source Datalog rewriter for OWL 2 RL ontological reasoning under SPARQL queries, available on the online webpage https://demacs-unical.github.io/DaRLing/.

Table 1 reports the main tools supporting or implementing natively ontology-mediated query answering over knowledge bases that fall in the RL profile of

[1] See https://wiki.dbpedia.org/
[2] See http://www.foaf-project.org/

Tool	License	Latest release	Query language	*sameAs*	Datatypes
Clipper [3]	Free	Dec 2015	SPARQL-BGP	under UNA	No
DaRLing [2]	Free	Jul 2020	SPARQL-BGP	Yes	Yes
DReW [4]	Free	Mar 2013	SPARQL-BGP	No	No
Orel [5]	Free	Feb 2010	ground queries	No	No
OWL2DLV [6]	Commercial	Jun 2019	SPARQL-BGP	under UNA	Yes
OwlOntDB [7]	-	-	SPARQL-DL$_E$	under UNA	No
RDFox [8]	Commercial	Jun 2020	SPARQL 1.1	Yes	Yes

Table 1: Main tools supporting OMQA over OWL 2 RL ontologies.

OWL 2, or beyond. Concerning the query language, apart from Orel, all the tools support SPARQL patterns: SPARQL 1.1, SPARQL-BGP [9], and SPARQL-DL$_E$ [10]. Finally, the row of OwlOntDB contains some missing value because the system is currently not available. Hence, none of the existing systems fully meet conditions (i)-(iv) above.

The *DaRLing* rewriter takes in input an RDF dataset (ABox) $\mathcal{A}$, an OWL 2 RL ontology (TBox) $\mathcal{T}$ and a SPARQL query $q(\mathbf{x})$, and constructs an equivalent program P with an output predicate *ans* of arity $|\mathbf{x}|$. Formally, for each $|\mathbf{x}|$-tuple of domain constants, $\mathcal{A} \cup \mathcal{T} \models q(\mathbf{t})$ if, and only if, the atom $ans(\mathbf{t})$ can be derived via P, where P is a Datalog program using inequality and stratified negation. A rewrite module is implemented for the translation of a Web ontology into a Datalog program. The rewriting process is subjected to an optimized version of a well-known normalization procedure [11] which aims to simplify the complex nature of axioms before translation takes place.

The system builds on top of the OWL API. It supports different input formats and knowledge bases organized in multiple files. Moreover, it can produce a suitable rewriting also if some inputs are missing. For example, in case the ABox is missing, then the generated program is simply equivalent to the pair TBox plus query.

By default, *DaRLing* rewrites under the *Unique Name Assumption* (UNA), i.e., presumes that different names represent different objects of the world. However, it is possible to explicitly choose to enable rewriting with the "*sameAs* management mode". The semantics of *sameAs* presupposes the enabling of matches between syntactically different but equivalent individuals.

In this setting, an expensive task – both in terms of time and memory consumption – is represented by the materialization of *sameAs*-cliques due to the enormous extension size that the latter typically assume. To accomplish this task, *DaRLing* generates a fragment of Datalog encoding in the rewriting of the input ontology. In particular, in order to avoid recursion over the *sameAs* predicate, the computation of the transitive closure is not "explicit" but aims to connect all the elements of any clique to the (lexicographical) minimum of that clique.

	Materialize		Query-driven	
	Clipper	*DaRLing*	Clipper	*DaRLing*
LUBM	199.29	204.02	24.94	28.26
Adolena	102.21	97.82	137.84	135.09
Stock Exchange	302.77	296.39	316.24	299.90
Vicodì	51.99	52.15	22.07	21.71
DBpedia	-	306.65	-	21,683.00

Table 2: Experiments on LUBM, Adolena, Stock Exchange, Vicodì and DBpedia. Times are in seconds.

Finally, in order to preserve the semantics of the *sameAs* property, *DaRLing* makes use of auxiliary rules to enable matching of equivalent individuals for each join between variables. As an example, consider an ontology featuring the rule

$$\mathit{DogOwner}(X) \leftarrow \text{hasPet}(X, Y) \ \wedge \ \mathit{Dog}(Y).$$

together with the following set of facts:

$$\mathit{hasPet}(\text{“Peter”}, \text{“Brian”}).$$
$$\mathit{Dog}(\text{“BrianGriffin”}).$$
$$\mathit{sameAs}(\text{“Brian”}, \text{“BrianGriffin”}).$$

Note how, despite that the fact sameAs(“Brian”,“BrianGriffin”) has the purpose of making the constants “Brian” and “BrianGriffin” interchangeable, the fact $\mathit{DogOwner}$(“Peter”) is not derived as it should. The system therefore, once the cliques have been calculated over a fresh predicate *sameClique*, adds the following rule to the Datalog ontology:

$$\mathit{DogOwner}(X) \leftarrow \mathit{hasPet}(X, Y_1),\ \mathit{Dog}(Y_2),\ \mathit{sameClique}(Y, Y_1),\ \mathit{sameClique}(Y, Y_2).$$

This latter rule, recognizing individuals “Brian” and “BrianGriffin” as equivalent since they belong to the same clique, derives $\mathit{DogOwner}$(“Peter”).

Table 2 summarizes the result of an experimental evaluation aimed at demonstrating the applicability of *DaRLing*. We considered well-known ontologies along with some queries and for each of them, we first generated Clipper and *DaRLing* rewritings. Then, we measured I-DLV [12, 13] times when executed over different datasets and provided with such rewritings. In particular, I-DLV was executed under to different scenarios: in the scenario *materialize* the system is forced to materialize the whole ontology and then prompted to answer to each query individually; in the scenario *query-driven* the system still runs each query one by one, but performs a more efficient evaluation tailored on the query at hand by enabling the magic sets technique [14]. We calculated the average running times in seconds per query and then reported the sums of the results thus obtained for each benchmark. Clipper has not been tested on DBpedia, since this ontology

requires a proper handling of the *sameAs* property and Clipper works only under UNA. Results show that performance achieved by I-DLV when using *DaRLing* outputs is comparable w.r.t. *Clipper*. Eventually, experiments on DBpedia (a real-world OWL 2 RL knowledge base) show how *DaRLing*'s rewriting strategy enables scalable query answering even in case the UNA is not a viable option.

References

1. Motik, B., Cuenca Grau, B., Horrocks, I., Wu, Z., Fokoue, A., Lutz, C.: OWL 2 Web Ontology Language Profiles (Second Edition). W3C Recommendation. World Wide Web Consortium (December 2012)
2. Fiorentino, A., Zangari, J., Manna, M.: Darling: A datalog rewriter for OWL 2 RL ontological reasoning under SPARQL queries. Theory and Practice of Logic Programming **20**(6) (2020) 958–973
3. Eiter, T., Ortiz, M., Simkus, M., Tran, T., Xiao, G.: Query rewriting for Horn-SHIQ plus rules. In: Proceedings of AAAI'12, AAAI Press (2012)
4. Xiao, G., Eiter, T., Heymans, S.: The drew system for nonmonotonic dl-programs. In: Proceedings of CSWS'12, Springer (2012) 383–390
5. Krötzsch, M., Mehdi, A., Rudolph, S.: Orel: Database-driven reasoning for OWL 2 profiles. In: Proceedings of DL'10. Volume 573 of CEUR Workshop Proceedings., CEUR-WS.org (2010)
6. Allocca, C., Calimeri, F., Civili, C., Costabile, R., Cuteri, B., Fiorentino, A., Fuscà, D., Germano, S., Laboccetta, G., Manna, M., Perri, S., Reale, K., Ricca, F., Veltri, P., Zangari, J.: Large-scale reasoning on expressive horn ontologies. In: Proceedings of Datalog 2.0. Volume 2368 of CEUR Workshop Proceedings., CEUR-WS.org (2019) 10–21
7. Faruqui, R.U., MacCaull, W.: O wl O nt DB: A scalable reasoning system for OWL 2 RL ontologies with large aboxes. In: Proceedings of FHIES'12. Volume 7789 of LNCS., Springer (2012) 105–123
8. Nenov, Y., Piro, R., Motik, B., Horrocks, I., Wu, Z., Banerjee, J.: Rdfox: A highly-scalable RDF store. In: Proceedings of ISWC'15. Volume 9367 of LNCS., Springer (2015) 3–20
9. Harris, S., Seaborne, A.: SPARQL 1.1 Query Language. W3C Recommendation. World Wide Web Consortium (March 2018)
10. Sirin, E., Parsia, B.: SPARQL-DL: SPARQL query for OWL-DL. In: Proceedings of OWLED'07. Volume 258 of CEUR Workshop Proceedings., CEUR-WS.org (2007)
11. Kazakov, Y.: Consequence-driven reasoning for horn SHIQ ontologies. In: Proceedings of IJCAI'09. (2009) 2040–2045
12. Calimeri, F., Fuscà, D., Perri, S., Zangari, J.: I-DLV: the new intelligent grounder of DLV. Intelligenza Artificiale **11**(1) (2017) 5–20
13. Calimeri, F., Fuscà, D., Perri, S., Zangari, J.: I-DLV: The new intelligent grounder of DLV. In: Proceedings of AI*IA'16. Volume 10037 of LNCS., Springer (2016) 192–207
14. Alviano, M., Leone, N., Veltri, P., Zangari, J.: Enhancing magic sets with an application to ontological reasoning. Theory and Practice of Logic Programming **19**(5-6) (2019) 654–670