Front. Comput. Sci.
https://doi.org/10.1007/s11704-018-7105-1

LETTER

The more irresistible Hi($\mathcal{SRIQ}$) for meta-modeling and meta-query answering

Zhenzhen GU (✉)[1,2], Songmao ZHANG[1]

1 Academy of Mathematics and Systems Science, Chinese Academy of Sciences, Beijing 100080, China
2 University of Chinese Academy of Sciences, Beijing 100049, China

1 Introduction and main contributions

There are two expressive sub-languages in OWL 2, where OWL 2 Full is of the highest expressivity compared with OWL 2 DL, so that its reasoning has turned undecidable. The most distinctive feature of OWL 2 Full is meta-modeling, i.e., names can have multiple uses, which, unfortunately, causes reasoning undecidability in OWL 2 Full. Meta-modeling can be frequently spotted in real-word domain knowledge bases (KBs), for example, the FMA KB for canonical human anatomy, OpenCyc and SUMO for commonsense. In these KBs, most of the names for classes or roles are also used as individuals, leading them to fall into the category of OWL 2 Full. In contrast with OWL 2 DL, reasoning in OWL 2 Full has largely been unexplored, and there are no Reasoners tailored for OWL 2 Full. The gap between meta-modeling requirement in reality and the lack of studies on reasoning and querying in OWL 2 Full raises a challenge.

To tackle this issue, we discuss meta-modeling extension in the description logic (DL) $\mathcal{SRIQ}$. The reason that the most expressive DL $\mathcal{SROIQ}$ for OWL 2 is not involved because the decidability of its conjunctive query answering remains unknown, whereas in $\mathcal{SRIQ}$, Diego Calvanese shows that conjunctive query answering is decidable [1]. The main contributions of this work can be summarized as follows:

1) By extending meta-modeling in $\mathcal{SRIQ}$, we define a sub-language of OWL 2 Full, called Hi($\mathcal{SRIQ}$). And, meta-queries are introduced for meta-knowledge querying.
2) We provide a way of reducing satisfiability checking and meta-query answering in Hi($\mathcal{SRIQ}$) to the corresponding reasoning tasks in $\mathcal{SRIQ}$. Based on this, we conclude that meta-modeling extension to $\mathcal{SRIQ}$ does not increase the reasoning complexity.
3) For scalability, we discuss a sub-language of Hi($\mathcal{SRIQ}$), called Hi($\mathcal{SRIF}$), and provide a divide-and-conquer approach for reasoning and meta-query answering.
4) For efficiency, we present three heuristics to optimize the procedure of meta-query answering in Hi($\mathcal{SRIQ}$).

The technical details, proofs and evaluations can be found in the support information.

2 Hi($\mathcal{SRIQ}$) and meta-queries

Hi($\mathcal{SRIQ}$) is defined from $\mathcal{SRIQ}$ by merging the sets of names for classes, roles and individuals into one set of names, thus all the names can have multiple uses without any restriction. The following knowledge described in OpenCyc KB can be captured by Hi($\mathcal{SRIQ}$):

$$\mathit{Football\_team} \sqsubseteq_c \mathit{SportsTeam},$$
$$\mathit{SportsTeamTypeBySport}(\mathit{Football\_team}).$$

The semantics of Hi($\mathcal{SRIQ}$) is specified by the ν-semantics

Received March 28, 2017; accepted May 22, 2017

E-mail: guzhenzhen0720@163.com

as defined in [2] which takes a similar way as OWL 2 RDF Based Semantics to interpret multiple uses of names. The v-satisfiability of Hi($\mathcal{SRIQ}$) is defined as usual.

Meta-queries (MQs) are defined from conjunctive queries (CQs) by allowing variables to occur in the class and role positions. Thus by MQs, the schema knowledge and data can be queried in a uniform way. For example, the query asking for what types of sports team *BarcelonaDragons* is can be formally represented as:

$$?x(BarcelonaDragons) \wedge SportsTeam(?x) \rightarrow q(?x).$$

For a query Q and KB $\mathcal{K}$, we use $\mathsf{ans}(Q, \mathcal{K})$ to denote the set of all the certain answers of Q over $\mathcal{K}$. Moreover, we just consider the MQs that each variable occurring in the class or role position also occurs in the head of the query.

3 Reasoning with Hi($\mathcal{SRIQ}$)

To reason over and query Hi($\mathcal{SRIQ}$) KBs, we first present a way of reducing v-satisfiability checking and CQ answering in Hi($\mathcal{SRIQ}$) to the corresponding reasoning tasks in $\mathcal{SRIQ}$. Then we show that MQ answering in Hi($\mathcal{SRIQ}$) can be captured by CQ answering.

• **v-satisfiability checking and CQ answering** The reasoning reduction from Hi($\mathcal{SRIQ}$) to $\mathcal{SRIQ}$ can be simply realized by the OWL 2 DL punning technique which treats OWL 2 Full ontologies as OWL 2 DL ontologies by interpreting the multiple uses of names as different semantic entities. However, the completeness cannot be guaranteed solely by this technique. This is caused by the fact that under v-semantics, the behaviors of names used as individuals will effect the same names used as classes or roles, i.e., if a Hi($\mathcal{SRIQ}$) KB implies $a \approx b$ (individual equivalence) then it also implies axioms $a =_c b$ (class equivalence) and $a =_r b$ (role equivalence). However, under punning, such effects of individuals on classes and roles do not exist anymore.

In order to obtain completeness, we exhaustively generate all the possible combinations of individual equivalence relations may implied by a Hi($\mathcal{SRIQ}$) KB. And in order not to increase the size of the original KB, we replace the equivalent individuals with a same name in the original KB rather than adding extra class equivalence axioms and role equivalence axioms. Concretely, for a Hi($\mathcal{SRIQ}$) KB $\mathcal{K}$ with n different individuals, there are totally B_n possibilities about individual equivalence relations, where B_n is the Bell number (see Wikipedia). Thus we can construct B_n $\mathcal{SRIQ}$ KBs for $\mathcal{K}$, and $\mathcal{K}$ is v-satisfiable iff there exists one constructed $\mathcal{SRIQ}$ KB that is satisfiable. Suppose $O_1 - O_m$ are all the constructed $\mathcal{SRIQ}$ KBs that are satisfiable. For a CQ Q, we can generate m CQs $Q_1 - Q_m$, and $\mathsf{ans}(Q, \mathcal{K}) = \cap_{i=1}^{m} \mathsf{ans}^{+}(Q_i, O_i)$ holds, where $\mathsf{ans}^{+}(Q_i, O_i)$ is the set of answers obtained by extending the answers in $\mathsf{ans}(Q_i, O_i)$ using the corresponding individual equivalence relations. By this reduction, we can get that v-satisfiability checking and CQ answering in Hi($\mathcal{SRIQ}$) have complexity of 2ExpTime and 3ExpTime, respectively.

• **Meta-query answering** The reasoning reduction from MQ answering to CQ answering is realized by materializing the class/role variables with names. Concretely, for a MQ Q and Hi($\mathcal{SRIQ}$) KB $\mathcal{K}$, a CP-binding of Q over $\mathcal{K}$ is a function ξ that maps each class (role) variable of Q to a name occurring in $\mathcal{K}$. And $\mathsf{ans}(Q, \mathcal{K}) = \cup_{\xi \in \mathcal{B}} \mathsf{ans}(Q\xi, \mathcal{K})$ holds where $\mathcal{B}$ is the set of all the CP-bindings of Q over $\mathcal{K}$ and $Q\xi$ is the CQ obtained by applying ξ over Q. Based on this, we get that the complexity of MQ answering in Hi($\mathcal{SRIQ}$) is 3ExpTime.

4 A practical approach for reasoning with a sub-language of Hi($\mathcal{SRIQ}$)

The reduction procedure illustrated in Section 3 indicates that reasoning and querying a Hi($\mathcal{SRIQ}$) KB may need to consider as many as an exponential size of $\mathcal{SRIQ}$ KBs w.r.t. the size of individuals in this KB. In this section, we define a sub-language of Hi($\mathcal{SRIQ}$), called Hi($\mathcal{SRIF}$), which features that when reasoning and querying a Hi($\mathcal{SRIF}$) KB, solely one $\mathcal{SRIQ}$ KB needs to be processed.

• **Hi($\mathcal{SRIF}$)** Hi($\mathcal{SRIF}$) is designed by imposing restrictions on constructors that can cause uncertainty of individual equivalence relations. For example, axioms in a Hi($\mathcal{SRIF}$) KB containing $\leqslant$ can solely have the form of $\top \sqsubseteq_c \leqslant 1R.\top$, i.e., the constructor $\leqslant$ can only be used to assert functional roles. Under such restrictions, for a Hi($\mathcal{SRIQ}$) KB $\mathcal{K}$, by materializing the implied individual equivalence relations, we can construct a $\mathcal{SRIQ}$ KB O so that $\mathcal{K}$ is satisfiable iff O is satisfiable. And for a CQ Q, a CQ Q' can be constructed so that $\mathsf{ans}(Q, \mathcal{K}) = \mathsf{ans}^{+}(Q', O)$ holds.

• **A divide-and-conquer approach** For large-scale KBs like OpenCyc and NCI, even though only one DL KB is considered, reasoning can still be infeasible. To tackle this issue, we further provide a divide-and-conquer reasoning approach. The basic idea is to construct a set $\mathcal{S}$ of Hi($\mathcal{SRIF}$) KBs with smaller and balanced sizes for a Hi($\mathcal{SRIF}$) KB $\mathcal{K}$ so that (1) $\mathcal{K}$ is v-satisfiable iff each KB in $\mathcal{S}$ is v-satisfiable; and (2) evaluating MQs over $\mathcal{K}$ can be soundly and completely captured by answering partitions of MQs over the KBs in $\mathcal{S}$. By this way, parallelization and distribution techniques can take effect to achieve feasibility and scalability. Take OpenCyc and NCI for example. By this divide-and-conquer approach, satisfiability checking, once infeasible, has successfully been finished within 50 and 10 seconds, respectively.

5 Meta-query answering optimization

Answering a MQ over a Hi($\mathcal{SRIQ}$) KB may need to consider as many as an exponential number of CQs w.r.t. the number of atoms in the query, making MQ answering time costly or even impractical. To tackle this issue, we devise three heuristics to optimize the MQ answering procedure.

• **Heuristic $\mathbf{H_1}$: Meta-query partition** This heuristic evaluates a partition of a MQ rather than the MQ itself so as to reduce the number of CQs need to be considered. For example, consider the following MQ Q:

$$?x(FootballTeam) \wedge ?y(FootballTeam, ?z) \rightarrow q(?x, ?y, ?z).$$

Without query partition, evaluating Q over the OpenCyc KB needs to answer as many as $1,080,855^2$ CQs. However, if we partition Q into the following two sub-queries:

$$?x(FootballTeam) \rightarrow q(?x),$$
$$?y(FootballTeam, ?z) \rightarrow q(?x, ?y),$$

only $2 \times 1,080,855$ CQs need to be answered.

• **Heuristic $\mathbf{H_2}$: Value transferring** During evaluating a partition of a MQ, the certain answers obtained so far can be transferred to the subsequent sub-query before it is evaluated. By this way, we can avoid blindly materializing class/role variables of this sub-query, and avoid computing the certain answers of this sub-query that do not coincide with the answers obtained so far. For example, consider the MQ Q:

$$SportsTeamTypeBySport(?x) \wedge ?x(BarcelonaDragons) \rightarrow q(?x).$$

Q has one partition consisting of two sub-queries q_1 and q_2:

$$q_1 : SportsTeamTypeBySport(?x) \rightarrow q(?x),$$
$$q_2 : ?x(BarcelonaDragons) \rightarrow q(?x).$$

After answering q_1, suppose we obtain a answer set $\{(a)\}$. If the bind $?x \rightarrow a$ is transferred to q_2, then evaluating q_2 can be realized without trying any CP-bindings, and the certain answers of q_2 that do not bind $?x$ to a will not be computed.

• **Heuristic $\mathbf{H_3}$: CP-binding punning** This heuristic uses the inclusion axioms in a Hi($\mathcal{SRIQ}$) KB to prune the CP-bindings so as to reduce the number of CQs need to be considered when evaluating a MQ under punning. For example, consider the MQ Q :$?x(a) \rightarrow q(?x)$ and the KB $\mathcal{K}$:

$$\mathcal{K} = (\{A_1 \sqsubseteq_c A_2,\ \ldots,\ A_{n-1} \sqsubseteq_c A_n\},\ \{B(a)\}).$$

Evaluating Q over $\mathcal{K}$ by trying the CP-binding $\xi_n = \{?x \rightarrow A_n\}$ returns null certain answers, i.e., $\mathsf{ans}(Q\xi_n, \mathcal{K}) = \emptyset$. Thus, by the inclusion axioms in $\mathcal{K}$, we know that for each binding $\xi_i = \{?x \rightarrow A_i\}$ $(1 \leqslant i \leqslant n-1)$, $\mathsf{ans}(Q\xi_i, \mathcal{K}) = \emptyset$. So the CQs $Q\xi_1 - Q\xi_{n-1}$ need not to be considered anymore.

By these three heuristics, the number of CQs actually considered can be reduced significantly. For example, without any heuristics, evaluating the MQ:

$$?c(Alan\_Turing) \wedge ?p(Alan\_Turing, ?z) \rightarrow q(?c, ?p, ?z).$$

over a set of KBs constructed for the commonsense KB SUMO shall answer 8.1 billion CQs, whereas only 95 CQs need to be evaluated when the heuristics are applied.

6 Conclusion

There have been works on extending decidable DLs with meta-modeling, for instance, [2–6]. Our study in this paper is the first to discuss the theoretical features of meta-modeling extension in $\mathcal{SRIQ}$ where we allow names to have multiple uses without any restrictions. Moreover, we discuss query answering which is essential for realizing knowledge reuse and sharing but rarely mentioned in the related studies. The future work will include developing more optimization techniques and extending Hi($\mathcal{SRIQ}$) to capture nominals.

Acknowledgements This work was supported by the National Key Research and Development Program of China (2016YFB1000902) and the National Natural Science Foundation of China (Grant Nos. 61232015, 61621003).

Supporting information The supporting information is available online at journal.hep.com.cn and link.springer.com.

References

1. Calvanese D, Eiter T, Ortiz M. Answering regular path queries in expressive description logics via alternating tree-automata. Information and Computation, 2014, 237: 12–55
2. Motik B. On the properties of metamodeling in OWL. Journal of Logic and Computation, 2007, 17(4): 617–637
3. De Giacomo G, Lenzerini M, Rosati R. Higher-order description logics for domain metamodeling. In: Proceedings of the AAAI Conference on Artificial Intelligence. 2011, 183–188
4. Motz R, Rohrer E, Severi P. The description logic SHIQ with a flexible meta-modeling hierarchy. Journal of Web Semantics: Science, Services and Agents on the World Wide Web, 2015, 35(4): 214–234
5. Gu Z, Zhang S. Querying large and expressive biomedical ontologies. In: Proceedings of IEEE International Conference on High Performance Computing and Communications. 2015, 491–496
6. Gu Z. Meta-modeling extension of Horn-SROIQ and query answering. In: Proceedings of International Workshop on Description Logics. 2016