# Meta-reasoning over OWL 2 QL using Datalog

Haya Majid Qureshi[1,*,†], Wolfgang Faber[1,*,†]

[1] University of Klagenfurt, Austria

Abstract

There has been increasing interest in enriching ontologies with meta-modeling and meta-querying for the past few years. Unfortunately, the Direct Semantics for OWL2 and SPARQL does not support meta-constructs in a satisfactory way: While meta-axioms can be syntactically expressed using punning, they are not treated as expected semantically. Meta-queries (for example, asking for classes that also occur as individuals) are not defined in SPARQL under the Direct Semantics Entailment Regime. To overcome this, a new semantic flavour for SPARQL, called Metamodeling Semantics Entailment Regime (MSER), has been introduced. In previous work, Cima et al. have proposed a reduction from OWL 2 QL query answering to query answering over Datalog. In this paper, we report on experiments for MSER query answering conducted with various Datalog engines.

**Keywords**

Meta-Reasoning, Ontology, SPARQL, Datalog

## 1. Introduction

For the past few years there has been an urge for enriching ontologies with meta-modelling and meta-querying. Meta-modelling allows for expressing meta-concepts (classes are instances of other classes) and meta-properties (relation between meta-concepts), and therefore makes conceptual modelling more flexible, as argued for instance in [1]. Meta-querying ports this idea to queries as well, allowing the use of the same variable in positions of different types. The de-facto standard language for ontologies, OWL 2, syntactically allows for meta-modelling by means of Punning, using the same name for ontology elements of different type (most notably, class and individual). However, the prevalent Direct Semantics (DS) does not interpret punning in the way intended by meta-modelling, as it will interpret the different occurrences of the same name as different entities. Similar considerations apply to meta-querying as well. SPARQL is the de-facto standard ontology query language. The logical underpinning for SPARQL queries over OWL 2 QL ontologies is defined by the *Direct Semantics Entailment Regime* (DSER) [2]. As the name implies, DSER relies on the Direct Semantics for ontologies and therefore imposes typing constraints on both the ontologies and queries that make meta-querying impossible.

To remedy these limitations, the Meta-modelling Semantics Entailment Regime (MSER) was proposed in [3], which does allow meta-modelling and meta-querying using SPARQL over OWL

*Datalog 2.0 2022: 4th International Workshop on the Resurgence of Datalog in Academia and Industry, September 05, 2022, Genova - Nervi, Italy*

* Corresponding author.

† These authors contributed equally.

haya.qureshi@aau.at (H. M. Qureshi); wolfgang.faber@aau.at (W. Faber)

0000-0002-6622-4950 (H. M. Qureshi); 0000-0002-0330-5868 (W. Faber)

2 QL. It provides a reduction from query-answering over OWL 2 QL to Datalog queries and reported experimental results using two Datalog engines, Logicblox and RDFox. This work summarises results obtained in [4], in which more Datalog engines have been evaluated. Some additional experimental results obtained since the publication of [4] are also included.

This work aims to reflect the idea of query answering under MSER that addresses the feasibility challenge for OWL 2 QL ontology language with (or without) the distinct flavour of metamodeling in Datalog back-end tools. Also, we investigated the problem of typing constraints by DSER via posing meta-queries to OWL 2 QL theories and evaluating the performance of these queries in Datalog engines.

## 2. Preliminaries

In this section, we briefly recall query answering under the Meta-modelling Semantics Entailment Regime (MSER) from [3].

In MSER, SPARQL query answering over OWL 2 QL ontologies is reduced to Datalog query answering. It defines (i) a translation function $\tau$ mapping OWL 2 QL axioms to Datalog facts is summarised Table 1 and (ii) a fixed rule base $\mathcal{R}^{ql}$ that captures inferences in OWL 2 QL reasoning (the full set of rules is available at https://git-ainf.aau.at/Haya.Qureshi/mhf-algo-testing). This representation is closer to a meta-programming representation than other Datalog embeddings that translate each axiom to a rule.

**Table 1**
$\tau$ Function

| | $\alpha$ | $\tau(\alpha)$ | $\alpha$ | $\tau(\alpha)$ |
|---|---|---|---|---|
| $\mathcal{P}^{ql,\mathcal{T}}_{\mathcal{O}}$ | $c1 \sqsubseteq c2$ | isacCC(c1, c2) | $r1 \sqsubseteq \neg\, r2$ | disjrRR(r1,r2) |
| | $c1 \sqsubseteq \exists r2^{-}.c2$ | isacCI(c1, r2, c2) | $c1 \sqsubseteq \neg\, c2$ | disjcCC(c1,c2) |
| | $\exists r1 \sqsubseteq \exists r2.c2$ | isacRR(r1,r2,c2 | $c1 \sqsubseteq \neg\exists r2^{-}$ | disjcCI(c1,r2) |
| | $\exists r1^{-} \sqsubseteq c2$ | isacIC(r1,c2) | $\exists r1 \sqsubseteq \neg\, c2$ | disjcRC(r1,c2) |
| | $\exists r1^{-} \sqsubseteq \exists r2.c2$ | isacIR(r1,r2,c2) | $\exists r_1 \sqsubseteq \neg\exists r2$ | disjcRR(r1,r2) |
| | $\exists r1^{-} \sqsubseteq \exists r2^{-}.c2$ | isacII(r1,r2,c2) | $\exists r1 \sqsubseteq \neg\exists r2^{-}$ | disjcRI(r1,r2) |
| | $r1 \sqsubseteq r2$ | isarRR(r1,r2) | $\exists r1^{-} \sqsubseteq \neg\, c2$ | disjcIC(r1,c2) |
| | $r1 \sqsubseteq r2^{-}$ | isarRI(r1,r2) | $\exists r1^{-} \sqsubseteq \neg\exists r2$ | disjcIR(r1,r2) |
| | $c1 \sqsubseteq \exists r2.c2$ | isacCR(c1,r2,c2) | $\exists r1^{-} \sqsubseteq \neg\exists r2^{-}$ | disjcII(r1,r2) |
| | $\exists r1 \sqsubseteq c2$ | isacRC(r1,c2) | $r1 \sqsubseteq \neg\, r2^{-}$ | disjrRI(r1,r2) |
| | $\exists r1 \sqsubseteq \exists r2^{-}.c2$ | isacRI(r1,r2,c2) | irref(r) | irrefl(r) |
| | refl(r) | refl(r) | | |
| $\mathcal{P}^{ql,\mathcal{A}}_{\mathcal{O}}$ | c(x) | instc(c,x) | $x \neq y$ | diff(x,y) |
| | r(x, y) | instr(r,x,y) | | |

## 3. Experiments

In this section we briefly describe the experiments that we have conducted, including the tools we used, the ontologies and queries we considered, and report on the outcomes. For a detailed discussion, see [4]. All material is available at https://git-ainf.aau.at/Haya.Qureshi/

mhf-algo-testing. We have implemented MSER in Java. For the Datalog back-end, we have evaluated nine tools, which stem from different paradigms. These tools are: **RDFox**, **LogicBlox**, **XSB**, **Clingo**, **DLV2**, **DLVHex**, **HexLite**, **Alpha** and **NoHR** .

Our experiments are based on the widely used Lehigh University Benchmark (LUBM)[1] dataset (with 1 and 9 universities) and Making Open Data Effectively USable (MODEUS)[2] ontologies in four sizes.

The **LUBM** datasets describe a university domain with information like departments, courses, students, and faculty. This dataset comes with 14 queries with different characteristics (low selectivity vs high selectivity, implicit relationships vs explicit relationships, small input vs large input, etc.).

The **MODEUS** ontologies describe the *Italian Public Debt* domain with information like financial liability or financial assets to any given contracts [5]. It comes with 8 queries. These queries are pure meta-queries as they span over several levels of the knowledge base. MODEUS ontologies are meta-modelling ontologies with meta-classes and meta-properties.

We ran experiments on a Linux batch server, running Ubuntu 20.04.3 LTS (GNU/Linux 5.4.0-88-generic x86_64) on one AMD EPYC 7601 (32-Core CPU), 2.2GHz, Turbo max. 3.2GHz. The machine is equipped with 515GB RAM and a 4TB hard disk. Java applications used OpenJDK 11.0.11 with a maximum heap size of 25GB. For each query, we have limited RAM to 8GB and runtime to 15 minutes. *OFT* and *OFM* refer to exceeding the time and memory limits, respectively.

### 3.1. Results

We next report the results of our experiments. All reported times are in seconds and include loading the Datalog program including facts and rules and answering the query. The best performance for each query is highlighted in bold face.

In Tables 2 we report the performance on standard queries over LUBM, respectively. While for the smaller ontology almost all queries could be answered by all systems within the resource limits, performance varies considerably. This situation is exasperated for the larger ontology, for which LogicBlox, NoHR, and Alpha could not answer any of the queries. On the other hand, Clingo and DLV2 exhibit consistently fast performance.

In Table 3, we have considered the meta-queries *mq1*, *mq4*, *mq5*, and *mq10* from [6] as they contain variables in-property positions and are long conjunctive queries. We have also considered two special-case queries *sq1* and *sq2* from [3] to exercise the MSER features and identify the new challenges introduced by the additional expressivity over the ABox queries. Basically, in special-case queries, we check the impact of DISJOINTWITH and meta-classes in a query. For this, like in [3], we have introduced a new class named *TypeOfProfessor* and make *FullProfessor*, *AssociateProfessor* and *AssistantProfessor* an instance of this new class and also we define *FullProfessor*, *AssociateProfessor* and *AssistantProfessor* to be disjoint from each other. Then, in *sq1* we are asking for all those $y$ and $z$, where $y$ is a professor, $z$ is a type of professor and $y$ is an instance of $z$. In *sq2*, we have asked for different pairs of professors.

[1]http://swat.cse.lehigh.edu/projects/lubm/
[2]http://www.modeus.uniroma1.it/modeus/node/6

**Table 2**
LUBM with standard-queries (execution times in seconds)

| | **q1** | **q2** | **q3** | **q5** | **q6** | **q7** | **q9** | **q10** | **q11** | **q12** | **q13** | **q14** |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **LUBM(1)** | | | | | | | | | | | | |
| **LogicBlox** | 91.63 | 92.35 | 93.63 | 93.26 | 91.37 | 93.38 | 90.98 | 91.92 | 94.82 | 91.84 | 92.10 | 92.65 |
| **RDFox** | 2.350 | 3.370 | 2.380 | 2.370 | 2.360 | 2.380 | 2.390 | 2.360 | 2.580 | 2.360 | 2.380 | 2.380 |
| **XSB** | **0.070** | 4.160 | **0.070** | **0.070** | **0.070** | 0.800 | 32.500 | **0.070** | **0.070** | **0.070** | **0.070** | **0.070** |
| **NoHR** | 52.82 | 66.710 | 52.650 | 59.650 | 59.670 | 60.430 | 88.010 | 55.500 | 53.700 | 57.300 | 50.690 | 50.600 |
| **Clingo** | 0.110 | **0.110** | 0.110 | 0.110 | 0.110 | **0.110** | **0.110** | 0.100 | 0.110 | 0.110 | 0.100 | 0.110 |
| **DLV2** | 0.100 | 0.120 | 0.110 | 0.110 | 0.110 | **0.110** | **0.110** | 0.110 | 0.110 | 0.110 | 0.110 | 0.110 |
| **DLVHex** | 7.250 | 7.240 | 7.250 | 7.780 | 7.810 | 7.230 | 7.780 | 7.290 | 7.240 | 7.200 | 7.250 | 7.840 |
| **HexLite** | 36.290 | 2.950 | 49.720 | OFT | OFT | 447.290 | 94.100 | 36.15 | 92.330 | 16.860 | 16.880 | OFT |
| **Alpha** | 94.280 | 91.010 | 97.280 | 96.020 | 93.310 | 94.670 | 95.760 | 92.940 | 97.740 | 98.220 | 92.910 | 100.040 |
| **LUBM(9)** | | | | | | | | | | | | |
| **LogicBlox** | OFM | OFM | OFM | OFM | OFM | OFM | OFM | OFM | OFM | OFM | OFM | OFM |
| **RDFox** | 51.580 | 50.600 | 49.530 | 50.610 | 51.610 | 51.620 | 50.540 | 49.520 | 50.590 | 49.460 | 50.460 | 50.470 |
| **XSB** | 45.770 | 526.140 | 18.270 | 18.160 | 18.230 | 69.830 | OFT | 18.650 | 18.350 | 18.300 | 18.330 | 17.750 |
| **NoHR** | OFT | OFT | OFT | OFT | OFT | OFT | OFT | OFT | OFT | OFT | OFT | OFT |
| **Clingo** | 1.320 | 1.320 | 1.320 | 1.330 | 1.320 | 1.320 | 1.330 | 1.330 | 1.330 | 1.330 | 1.330 | 1.320 |
| **DLV2** | **0.980** | **1.090** | **0.860** | **1.120** | **1.070** | **1.090** | **1.090** | **1.100** | **1.090** | **1.080** | **1.090** | **1.050** |
| **DLVHex** | 386.320 | 387.790 | 385.880 | 389.610 | 412.470 | 386.750 | 386.230 | 384.920 | 387.130 | 386.650 | 386.860 | 386.860 |
| **HexLite** | 646.540 | 30.020 | OFT | OFT | OFT | OFT | OFT | 636.570 | OFT | OFT | 221.820 | OFT |
| **Alpha** | OFM | OFM | OFM | OFM | OFM | OFM | OFM | OFM | OFM | OFM | OFM | OFM |

It can be seen in Table 3 that the overall performance of meta-query evaluation is similar to the one in Table 2. Clingo and DLV2 exhibits the regular performance. XSB and RDFox shows the good performance on LUBM(1) but their performance get effected by the size of ontology. On the other hand, LogicBlox, NoHR, Alpha, HexLite and DLVHex shows slower performance but deteriorates with the size of the ontology.

In Table 4 we report the performance on the larger MODEUS queries. It can be seen immediately that many of the systems struggle considerably with these. Some considerations on the causes of this are: The MODEUS dataset consists of meta-layers, which appear to cause many tools to do more inferencing. We also conjecture that the presence of many *disjoint* axioms causes particularly many inferences.

On the positive side, DLV2 and XSB exhibit acceptable performance for these queries, with DLV2 being the best overall performer. DLV2 exhibits very stable performance with roughly the same execution time for all queries, which is quite remarkable. We assume that the magic set technique implemented in DLV2 has a huge impact here. The time is affected slightly by the size of the dataset, which is expected, though. XSB uses a top-down evaluation and therefore has similar advantages as the magic set technique.

Interestingly, we believe that at least LogicBlox (and perhaps also RDFox) also implements a magic set technique, yet does not seem to be able to take advantage from it. We conjecture that those systems build quite complicated and large datastructures for the Datalog program, for instance various indices. These systems might perform better when huge amounts of memory

**Table 3**
LUBM with meta-queries (execution times in seconds)

| | **mq1** | **mq4** | **mq5** | **mq10** | **sq1** | **sq2** |
|---|---|---|---|---|---|---|
| | | | **LUBM(1)** | | | |
| **LogicBlox** | 92.99 | 93.07 | 92.88 | 92.13 | 94.82 | 93.00 |
| **RDFox** | 2.840 | 2.370 | 2.370 | 2.360 | 2.370 | 2.390 |
| **XSB** | 0.190 | **0.000** | **0.070** | **0.070** | 6.000 | **0.080** |
| **NoHR** | 65.300 | 49.530 | 54.690 | 54.000 | 56.290 | 54.590 |
| **Clingo** | **0.110** | 0.110 | 0.110 | 0.110 | **0.110** | 0.110 |
| **DLV2** | **0.110** | 0.110 | 0.110 | 0.110 | 0.120 | 0.110 |
| **DLVHex** | 9.890 | 7.210 | 7.770 | 7.230 | 7.750 | 7.760 |
| **HexLite** | OFT | 2.980 | OFT | 100.580 | 241.700 | 50.720 |
| **Alpha** | 93.460 | 95.260 | 96.510 | 100.760 | 96.290 | 94.200 |
| | | | **LUBM(9)** | | | |
| **LogicBlox** | OFM | OFM | OFM | OFM | OFM | OFM |
| **RDFox** | 49.410 | 50.640 | 52.700 | 50.570 | 50.530 | 49.540 |
| **XSB** | 18.770 | 18.300 | 19.750 | 18.800 | 40.680 | 18.130 |
| **NoHR** | OFT | OFT | OFT | OFT | OFT | OFT |
| **Clingo** | 1.330 | 1.320 | 1.330 | 1.330 | 1.320 | 1.320 |
| **DLV2** | **1.150** | **1.090** | **1.130** | **1.090** | **1.100** | **0.980** |
| **DLVHex** | 534.930 | 387.270 | 407.170 | 386.420 | 704.490 | 704.750 |
| **HexLite** | OFT | 632.930 | OFT | OFT | OFT | OFT |
| **Alpha** | OFM | OFM | OFM | OFM | OFM | OFM |

are available and several queries are posed over the same program without reloading it.

## 4. Conclusion

In this work, we have tested several Datalog engines on OWL 2 QL MSER query answering without any restriction, as defined in [3]. While most tools are able to answer queries over smaller ontologies, scalability seems to be an issue for many of them. However, there are some exceptions, notably XSB and DLV2, which also show good performance over large and complex ontologies. Indeed, our experiments show that DLV2 appears to be a promising back-end for meta-querying over OWL 2 QL.

We show that query answering under Datalog reduction of MSER with metamodeling and meta-querying feature is feasible for some tools (or, in our case, just DLV2). At the same time, some suffer from the existence of meta-axioms over several layers. The meta-queries over LUBM do not include meta-axioms. However, most tools could perform well despite the metamodeling capabilities associated with the query language that extracts the information spanning several levels of an ontology. On the other hand, some tools could perform with MSER without the metamodeling feature in ontologies and with standard queries, while others get affected by the size of the ontology.

**Table 4**
MODEUS with meta-queries (execution times in seconds)

| | **mq0** | **mq1** | **mq2** | **mq3** | **mq4** | **mq5** | **mq6** | **mq8** |
|---|---|---|---|---|---|---|---|---|
| **MEF-00** | | | | | | | | |
| **LogicBlox** | OFM | OFM | OFM | OFM | OFM | OFM | OFM | OFM |
| **RDFox** | OFM | OFM | OFM | OFM | OFM | OFM | OFM | OFM |
| **XSB** | 39.750 | 41.910 | 51.300s | 45.130 | 40.020 | 51.390 | 74.040 | 44.440 |
| **NoHR** | OFT | 682.770 | OFM | OFT | 232.580 | OFT | 314.250 | 210.97 |
| **Clingo** | OFM | OFM | OFM | OFM | OFM | OFM | OFM | OFM |
| **DLV2** | **5.450** | **5.870** | **7.570** | **5.290** | **4.750** | **6.700** | **5.040** | **4.780** |
| **DLVHex** | OFM | OFM | OFM | OFM | OFM | OFM | OFM | OFM |
| **HexLite** | OFM | OFM | OFM | OFM | OFM | OFM | OFM | OFM |
| **Alpha** | OFM | OFM | OFM | OFM | OFM | OFM | OFM | OFM |
| **MEF-01** | | | | | | | | |
| **LogicBlox** | OFM | OFM | OFM | OFM | OFM | OFM | OFM | OFM |
| **RDFox** | OFM | OFM | OFM | OFM | OFM | OFM | OFM | OFM |
| **XSB** | 94.100 | 82.860 | 97.160 | 95.890 | 84.470 | 108.490 | 128.750 | 80.440 |
| **NoHR** | 757.19 | OFT | OFM | 745.890 | 331.340 | OFT | 406.360 | 280.880 |
| **Clingo** | OFM | OFM | OFM | OFM | OFM | OFM | OFM | OFM |
| **DLV2** | **6.400** | **8.060** | **9.740** | **6.540** | **5.750** | **7.520** | **5.630** | **5.530** |
| **DLVHex** | OFM | OFM | OFM | OFM | OFM | OFM | OFM | OFM |
| **HexLite** | OFM | OFM | OFM | OFM | OFM | OFM | OFM | OFM |
| **Alpha** | OFM | OFM | OFM | OFM | OFM | OFM | OFM | OFM |
| **MEF-02** | | | | | | | | |
| **LogicBlox** | OFM | OFM | OFM | OFM | OFM | OFM | OFM | OFM |
| **RDFox** | OFM | OFM | OFM | OFM | OFM | OFM | OFM | OFM |
| **XSB** | 50.450 | 44.500 | 58.770 | 46.280 | 41.160 | 57.620 | 80.210 | 45.190 |
| **NoHR** | 617.270 | OFT | OFM | OFT | 340.620 | OFT | 509.190 | 209.420 |
| **Clingo** | OFM | OFM | OFM | OFM | OFM | OFM | OFM | OFM |
| **DLV2** | **5.110** | **5.610** | **7.410** | **5.390** | **4.500** | **6.520** | **4.640** | **4.590** |
| **DLVHex** | OFM | OFM | OFM | OFM | OFM | OFM | OFM | OFM |
| **HexLite** | OFM | OFM | OFM | OFM | OFM | OFM | OFM | OFM |
| **Alpha** | OFM | OFM | OFM | OFM | OFM | OFM | OFM | OFM |
| **MEF-03** | | | | | | | | |
| **LogicBlox** | OFM | OFM | OFM | OFM | OFM | OFM | OFM | OFM |
| **RDFox** | OFM | OFM | OFM | OFM | OFM | OFM | OFM | OFM |
| **XSB** | 94.230 | 80.740 | 99.580 | 83.800 | 92.660 | 107.240 | 131.800 | 82.500 |
| **NoHR** | 713.410 | OFT | OFM | 757.010 | 306.180 | OFM | 396.000 | 289.420 |
| **Clingo** | OFM | OFM | OFM | OFM | OFM | OFM | OFM | OFM |
| **DLV2** | **6.830** | **7.350** | **10.010** | **6.880** | **5.540** | **7.250** | **5.450** | **5.540** |
| **DLVHex** | OFM | OFM | OFM | OFM | OFM | OFM | OFM | OFM |
| **HexLite** | OFM | OFM | OFM | OFM | OFM | OFM | OFM | OFM |
| **Alpha** | OFM | OFM | OFM | OFM | OFM | OFM | OFM | OFM |

## References


[1] B. Motik, On the properties of metamodeling in OWL, Journal of Logic and Computation 17 (2007) 617–637. doi:10.1093/logcom/exm027.

[2] B. Glimm, Using SPARQL with RDFS and OWL Entailment, in: Reasoning Web International Summer School, Springer, 2011, pp. 137–201.

[3] G. Cima, G. De Giacomo, M. Lenzerini, A. Poggi, On the SPARQL metamodeling semantics entailment regime for OWL 2 QL ontologies, in: Proceedings of the 7th International Conference on Web Intelligence, Mining and Semantics, 2017, pp. 1–6.

[4] H. M. Qureshi, W. Faber, An Evaluation of Meta-reasoning over OWL 2 QL, in: International Joint Conference on Rules and Reasoning, Springer, 2021, pp. 218–233.

[5] M. Lenzerini, L. Lepore, A. Poggi, Metaquerying made practical for OWL 2 QL ontologies, Information Systems 88 (2020) 101294.

[6] R. Kontchakov, M. Rezk, M. Rodriguez-Muro, G. Xiao, M. Zakharyaschev, Answering SPARQL queries over databases under OWL 2 QL entailment regime, in: International Semantic Web Conference, Springer, 2014, pp. 552–567.