Improving Practical Reasoning on top of SPARQL

Stefan Bischof[1,2]

[1] Vienna University of Economics and Business, Austria
stefan.bischof@wu.ac.at
[2] Siemens AG Österreich, Austria

Abstract. Reasoning techniques are not well received by the developer community. One reason is the cost of providing SPARQL endpoints with enabled reasoning. Another reason is the missing support for reasoning on numbers, which is needed for tasks such as data analytics. An important aspect of the second problem is the high number of missing values that inherently occur when integrating data on a global scale. In this work we propose two approaches to improving the situation for both problems by first developing the necessary techniques as well as prototypical applications based on query rewriting and a practical evaluation.

Introduction

Although reasoning technologies are standardized and sufficiently implemented, uptake in practical applications is underwhelming. We focus on access to Semantic Web data via SPARQL endpoints and see two problems: reasoning on public endpoints is expensive and the lack of support for reasoning with numbers.

Reasoning on public endpoints is expensive. Semantic Web data is often accessible via open SPARQL endpoints that usually do not support reasoning. There are several reasons for this missing feature:

- explicit specification of reasoning in SPARQL became available only recently with the SPARQL 1.1 recommendation [30],
- suboptimal performance due to SPARQL engines not being highly optimized (the language specification is only a few years old)
- slow query response times due to many concurrent requests performed on *public* SPARQL endpoints, and
- low acceptance and knowledge of the benefits of reasoning with the different OWL 2 profiles and SPARQL.

One option for a user to still get query answers under some reasoning scheme is to use some form of *ontology based data access* (OBDA) [24] query rewriting prior to query execution. Reasoning by rewriting is the target technique for the OWL 2 QL profile. But the resulting rewritings (which usually turn SPARQL basic graph patterns into unions of basic graph patterns) are exponential in size, with respect to the original query and the TBox [10], and can thus result in slow query response times or even time out. These rewritings are also *statically* dependent on the ontology, i.e., when the ontology changes, all the queries have to be rewritten *again*.

Lack of support for reasoning with numbers. Data analysts working with statistical or other numeric data in general are interested in computing new values based on measured values. We focus on numerical values because our use case, the Open City Data Pipeline, operates mainly on statistical data of cities (other types of literal values such as strings or dates are left for future work). These computations are usually specified as functions or equations. *Functions* are practically used in every spreadsheet application. In Business Intelligence (BI) and Data Ware Houses (DWH) several operations use aggregate functions. *Equations* describe relations between numerical attributes. These two representations are already widely used when working with numerical values. But value computation is not supported by OWL 2. On the one hand a suitable semantics should support equations. On the other hand aggregate functions can play a similar role for the Semantic Web as they are already playing for DWH scenarios: aiding decision support with data analytics.

A naive approach to implement some form of functions on RDF databases (triple stores) could be to export all numerical data and the functions to an external numerical reasoner; any kind of statistical or mathematical solving software such as R would be enough. Compute the new values and then materialize them in the database. This approach becomes cumbersome when taking updates and big datasets into account.

Semantic Web data is published by many different organizations, with varying quality and adhering to different ontologies or vocabularies. Incompleteness of data is an inherent characteristic of such a heterogeneous integrated global data base. But whenever we want to compare entities we need comparable data. By using statistical methods for imputing missing values we can compute estimates for unknown values and thus allow comparisons.

State of the Art

Attribute properties (or concrete domains) in Description Logics (DL) [1] are usually defined based on a separate attribute domain (or concrete domain). Any reasoning in this attribute domain is left to a domain specific reasoning method. In practice OWL 2 offers only lightweight reasoning for datatype properties.

Racer [13] provides no SPARQL interface but uses its own functional query language *new Racer Query Language* (nRQL). The system allows for modeling some forms of equation axioms, cf. examples modeling unit conversions by [14], but uses these only for satisfiability testing and not for query answering.

SWRL [15,16] implementations like Pellet [29] implement DL-safe rules [21], that is, rules where each variable appears in at least one non-DL-atom.

Jena [17] provides rule-based inference on top of its triple store TDB in a proprietary rule language with built-in functions, with SPARQL querying on top. Jena can execute rules in backward and forward mode. Jena offers a hybrid rule based reasoning where, e.g., pure RDFS inferencing is executed in a backward-chaining manner, but still can be combined with forward rules.

OBDA has been an important topic in applied and foundational research. Even before SPARQL provided support for this feature, several projects used ontologies to integrate different data sources or to provide views over legacy databases (e.g., [10]). Several directions of optimizations and systems have already been proposed [9,11,18,22,23,26–28].

Expected and Preliminary Results

Equation semantics and RDF syntax. We extend given reasoning semantics to support numerical inferences based on the OWL 2 QL profile. This profile is optimized for backward-chaining algorithms. By extending this profile with an equation semantics we have a good basis for rewriting algorithms.

Example 1. An ontology for statistical data of cities could include the following relation between the properties *population*, city *area* and *population density*:

$$\texttt{:popDensity} = \frac{\texttt{:population}}{\texttt{:area}}$$

Algorithm for SPARQL query answering. We extend the ontology languages by expressions allowing value computation, which includes especially aggregates.

To implement SPARQL query answering on top of RDF databases we will devise a (up to some extent) scalable algorithm for the specified equation semantics. The algorithm will work in a backward-chaining manner by rewriting input SPARQL queries to new SPARQL queries which have the relevant part of the ontology encoded inside them. This approach builds upon known algorithms from OBDA, such as PerfectRef [10].

Example 2. Using the knowledge from Example 1, a SPARQL query asking for the population density of Berlin (:Berlin :popDensity ?PD) could be rewritten to a union query with (i) the original basic graph pattern and (ii) a bind graph pattern computing the population density from the population and area values:

```
SELECT ?PD WHERE {
  { :Berlin :popDensity ?PD }
  UNION
  { :Berlin :population ?P . :Berlin :area ?A . BIND (?P/?A AS ?PD) } }
```

We will implement the algorithm based on state of the art frameworks and libraries and plan to publish the source code as open source.

Algorithm for schema agnostic SPARQL query answering. By exploiting the new SPARQL 1.1 features “property paths” and “value assignment” we can give an algorithm to rewrite queries independently of the ontology producing typically also shorter queries (excluding cases where ontological reasoning does not influence the query results, for example an empty ontology).

Since the algorithm is performing a dynamic reasoning in the sense that the ontology is involved not at compile time (i.e., query rewriting time) but only at query evaluation time, the query evaluation performance is expected to be slower than PerfectRef rewritings. To improve this situation we will propose an approach for optimization by partial path materialization.

We will implement the algorithm based on state of the art frameworks and libraries and plan to publish the source code as open source.

Approach to handle incomplete numerical data. We will present an approach to fill up incomplete numerical data by standard machine learning methods. This step is necessary to have a better basis and comparison for the equation rewriting algorithm described above. We apply this approach on statistical data from the city domain where several datasets are available as open data.

Proof of practicality. A practical evaluation based on standard SPARQL benchmarks will demonstrate how the approach performs for SPARQL query answering. We reuse existing OBDA and OWL benchmarks as far as possible for comparability [2,12,19,20,27,31].

Methodical Approach

Literature review. As usual a first step includes an extensive literature review in the area to find and categorize related research. This includes especially works on OBDA and description logics with concrete attributes.

Devise RDF syntax and semantics based on state of the art. The RDF vocabulary or OWL 2 ontology will be based on well known ontologies in the domain. We aim for an intuitive semantics to express relations between number properties. We show this intuition by mapping known equations, e.g., from Eurostat, to our ontology. Preliminary results are published [7].

Create an algorithm for query answering with equations. We will devise a reasoning algorithm following the specified equation semantics. We implement the algorithm based on existing open source RDF and SPARQL implementations. We will also investigate the computational complexity of the query rewriting including suitable ontology language fragments. Preliminary results are published [7].

Prototypical implementation. The Open City Data Pipeline uses several techniques to collect, clean, process, and republish statistical open city data as Linked Open Data. Figure 1 shows the architecture and main components of the system. The current version of the web UI as well as Linked Open Data is available at `http://citydata.wu.ac.at`. Preliminary results are published [6,8,25].

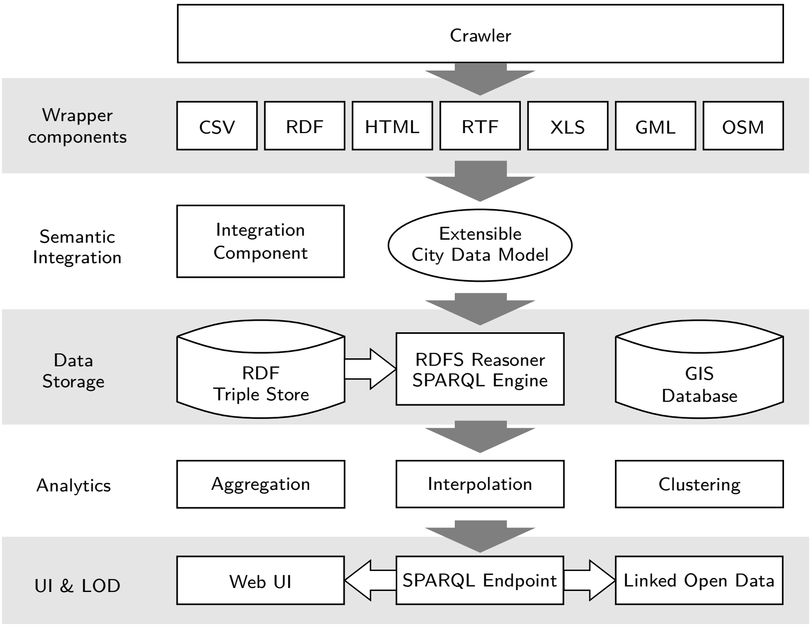

Fig. 1. City Data Pipeline architecture showing components for crawling wrapping, cleaning, integrating, and presenting information

Create an algorithm for OWL 2 QL query answering. We will devise a reasoning algorithm following the OWL 2 QL profile semantics. We implement the algorithm based on existing open source RDF andSPARQL implementations. This algorithm will need only the SPARQL query as input but not the ontology. Since the vanilla version of this rewriting leads to queries with long and nested path queries we will devise at least one optimization making evaluation of the rewritten queries more feasible. We will also investigate the complexity of the query rewriting. Preliminary results are published [3–5] and the online version of the rewriter is available at `http://citydata.wu.ac.at/SPR/`. Listing 1 shows the full rewriting result of a query containing a single triple pattern `{?p a foaf:Person}`.

Evaluation. We will measure the query response times under controlled conditions varying several dimensions: (i) input query, (ii) dataset, and (iii) ontology. Each of these dimensions allows for several changes, be it size or structure. We will investigate SPARQL query response times and use standard benchmarks as far as possible. We will focus on feasibility of the query rewritings and their optimizations. A preliminary unpublished evaluation shows the expected slow query response times for the OWL 2 QL rewriting with potential for improvement.

Listing 1. OWL 2 QL path rewriting of {?p a foaf:Person}

```
SELECT ?p WHERE {
  ?_v0 (((((rdfs:subClassOf|owl:equivalentClass)|^owl:equivalentClass)|((owl:
      ↪ intersectionOf/(rdf:rest)*)/rdf:first))|((owl:onProperty/((((rdfs:
      ↪ subPropertyOf|owl:equivalentProperty)|^owl:equivalentProperty)|(((owl:
      ↪ inverseOf|^owl:inverseOf)/(((rdfs:subPropertyOf|owl:equivalentProperty)
      ↪ |^owl:equivalentProperty))*)/(owl:inverseOf|^owl:inverseOf))))*)/(^owl:
      ↪ onProperty|rdfs:domain)))|((((owl:onProperty/((((rdfs:subPropertyOf|
      ↪ owl:equivalentProperty)|^owl:equivalentProperty)|(((owl:inverseOf|^owl:
      ↪ inverseOf)/(((rdfs:subPropertyOf|owl:equivalentProperty)|^owl:
      ↪ equivalentProperty))*)/(owl:inverseOf|^owl:inverseOf))))*)/(owl:
      ↪ inverseOf|^owl:inverseOf))/(((rdfs:subPropertyOf|owl:equivalentProperty
      ↪ )|^owl:equivalentProperty))*)/rdfs:range))* foaf:Person
  { { ?p rdf:type ?_v0}
        UNION
    { ?_v1 (((rdfs:subPropertyOf|owl:equivalentProperty)|^owl:equivalentProperty
        ↪ )|(((owl:inverseOf|^owl:inverseOf)/(((rdfs:subPropertyOf|owl:
        ↪ equivalentProperty)|^owl:equivalentProperty))*)/(owl:inverseOf|^owl:
        ↪ inverseOf)))*/(^owl:onProperty|rdfs:domain) ?_v0 .
      ?p ?_v1 _:b0
  } }
      UNION
  { ?_v1 ((((rdfs:subPropertyOf|owl:equivalentProperty)|^owl:equivalentProperty)
      ↪ |(((owl:inverseOf|^owl:inverseOf)/(((rdfs:subPropertyOf|owl:
      ↪ equivalentProperty)|^owl:equivalentProperty))*)/(owl:inverseOf|^owl:
      ↪ inverseOf))))*/rdfs:range ?_v0 .
        _:b1 ?_v1 ?p
  } }
```

Acknowledgments. This work was partly funded by the Vienna Science and Technology Fund (WWTF) project ICT12-015 and the European Commissions Seventh Framework Programme project CityPulse under grant agreement no. 609035. The author thanks the anonymous reviewers for their detailed and insightful comments to improve the paper.

References

1. Baader, F., Calvanese, D., McGuinness, D., Nardi, D., Patel-Schneider, P. (eds.): The Description Logic Handbook: Theory, Implementation, and Applications. Cambridge University Press, second edn. (2007)
2. Bail, S., Alkiviadous, S., Parsia, B., Workman, D., Van Harmelen, M., Concalves, R., Garilao, C.: Fishmark: A linked data application benchmark. In: Proc. Int. Joint Workshop on Scalable and High-Performance Semantic Web Systems (SSWS+HPCSW '12). CEUR Workshop Proceedings, vol. 943, pp. 1–15. CEUR-WS.org (2012)

3. Bischof, S., Krötzsch, M., Polleres, A., Rudolph, S.: Schema-agnostic query rewriting in SPARQL 1.1. In: Mika, P., Tudorache, T., Bernstein, A., Welty, C., Knoblock, C., Vrandečić, D., Groth, P., Noy, N., Janowicz, K., Goble, C. (eds.) The Semantic Web – ISWC 2014. vol. 8796, pp. 584–600. Springer, Riva del Garda, Italy (Oct 2014)
4. Bischof, S., Krötzsch, M., Polleres, A., Rudolph, S.: Schema-agnostic query rewriting in SPARQL 1.1: Technical report. http://stefanbischof.at/publications/iswc14/ (2014)
5. Bischof, S., Krötzsch, M., Polleres, A., Rudolph, S.: Schema-agnostic query rewriting for OWL QL. In: Calvanese, D., Konev, B. (eds.) Proceedings of the 28th International Workshop on Description Logics (DL'15). CEUR Workshop Proceedings, vol. 1350. CEUR-WS.org, Athens, Greece (Jun 2015)
6. Bischof, S., Martin, C., Polleres, A., Schneider, P.: Open city data pipeline: Collecting, integrating, and predicting open city data. In: Völker, J., Paulheim, H., Lehmann, J., Svate, V. (eds.) Proceedings of the 4th Workshop on Knowledge Discovery and Data Mining Meets Linked Open Data co-located with 12th Extended Semantic Web Conference (ESWC'15). CEUR Workshop Proceedings, vol. 1365. CEUR-WS.org, Portoroz, Slovenia (May 2015)
7. Bischof, S., Polleres, A.: RDFS with attribute equations via SPARQL rewriting. In: Cimiano, P., Corcho, O., Presutti, V., Hollink, L., Rudolph, S. (eds.) The Semantic Web: Semantics and Big Data – Proceedings of the 10th ESWC (ESWC'13). vol. 7882, pp. 335–350. Springer Berlin Heidelberg, Montpellier, France (May 2013)
8. Bischof, S., Polleres, A., Sperl, S.: City data pipeline - a system for making open data useful for cities. In: Lohmann, S. (ed.) Proceedings of the I-SEMANTICS 2013 Posters & Demonstrations Track. CEUR Workshop Proceedings, vol. 1026, pp. 45–49. CEUR-WS.org, Graz, Austria (Sep 2013)
9. Calvanese, D., De Giacomo, G., Lembo, D., Lenzerini, M., Poggi, A., Rodríguez-Muro, M., Rosati, R., Ruzzi, M., Savo, D.F.: The MASTRO system for ontology-based data access. Semantic Web Journal 2(1), 43–53 (Jan 2011)
10. Calvanese, D., Giacomo, G.D., Lembo, D., Lenzerini, M., Rosati, R.: Tractable reasoning and efficient query answering in description logics: The DL-Lite family. J. Automated Reasoning 39(3), 385–429 (2007)
11. Gottlob, G., Orsi, G., Pieris, A.: Ontological queries: Rewriting and optimization. In: Abiteboul, S., Böhm, K., Koch, C., Tan, K.L. (eds.) Proc. 27th Int. Conf. on Data Engineering (ICDE'11). pp. 2–13. IEEE Computer Society (2011)
12. Guo, Y., Pan, Z., Heflin, J.: LUBM: A benchmark for OWL knowledge base systems. Web Semantics: Science, Services and Agents on the World Wide Web 3(2–3), 158–182 (2005)
13. Haarslev, V., Möller, R.: Racer system description. In: Gor, R., Leitsch, A., Nipkow, T. (eds.) Proc. 1st Int. Joint Conf. on Automated Reasoning (IJCAR'01). LNCS, vol. 2083, pp. 701–705. Springer (2001)
14. Haarslev, V., Möller, R.: Description logic systems with concrete domains: Applications for the semantic web. In: Proc. 10th Int. Workshop on Knowledge Representation meets Databases (KRDB'03) (2003)
15. Horrocks, I., Patel-Schneider, P.F.: A proposal for an OWL rules language. In: Feldman, S.I., Uretsky, M., Najork, M., Wills, C.E. (eds.) Proc. 13th Int. Conf. on World Wide Web (WWW'04). pp. 723–731. ACM (2004)
16. Horrocks, I., Patel-Schneider, P.F., Boley, H., Tabet, S., Grosof, B.N., Dean, M.: SWRL: A Semantic Web Rule Language. W3C Member Submission (21 May 2004), available at http://www.w3.org/Submission/SWRL/

17. Jena, A.: Reasoners and rule engines: Jena inference support. https://jena.apache.org/documentation/inference/, accessed on May 24th, 2015
18. Kontchakov, R., Lutz, C., Toman, D., Wolter, F., Zakharyaschev, M.: The combined approach to ontology-based data access. In: Walsh, T. (ed.) Proc. 22nd Int. Joint Conf. on Artificial Intelligence (IJCAI'11). pp. 2656–2661. AAAI Press (2011)
19. Lanti, D., Rezk, M., Xiao, G., Calvanese, D.: The NPD benchmark: Reality check for OBDA systems. In: Alonso, G., Geerts, F., Popa, L., Barceló, P., Teubner, J., Ugarte, M., den Bussche, J.V., Paredaens, J. (eds.) Proceedings of the 18th International Conference on Extending Database Technology, EDBT'15. pp. 617–628. OpenProceedings.org, Brussels, Belgium (Mar 2015)
20. Lutz, C., Seylan, I., Toman, D., Wolter, F.: The combined approach to OBDA: Taming role hierarchies using filters. In: Alani, H., Kagal, L., Fokoue, A., Groth, P., Biemann, C., Parreira, J.X., Aroyo, L., Noy, N., Welty, C., Janowicz, K. (eds.) The Semantic Web – ISWC 2013, LNCS, vol. 8218, pp. 314–330. Springer Berlin Heidelberg (2013)
21. Motik, B., Sattler, U., Studer, R.: Query answering for OWL DL with rules. J. Web Semantics 3(1), 41–60 (2005)
22. Pérez-Urbina, H., Motik, B., Horrocks, I.: A comparison of query rewriting techniques for DL-lite. In: Cuenca Grau, B., Horrocks, I., Motik, B., Sattler, U. (eds.) Proc. 22nd Int. Workshop on Description Logics (DL'09). CEUR Workshop Proceedings, vol. 477. CEUR-WS.org (2009)
23. Pérez-Urbina, H., Motik, B., Horrocks, I.: Tractable query answering and rewriting under description logic constraints. J. Applied Logic 8(2), 186–209 (2010)
24. Poggi, A., Lembo, D., Calvanese, D., De Giacomo, G., Lenzerini, M., Rosati, R.: Linking data to ontologies. In: Spaccapietra, S. (ed.) Journal on Data Semantics X, LNCS, vol. 4900, pp. 133–173. Springer Berlin Heidelberg (2008)
25. Polleres, A., Bischof, S., Schreiner, H.: City data pipeline – a report about experiences from using open data to gather indicators of city performance. In: European Data Forum 2014 (May 2014)
26. Rodríguez-Muro, M., Calvanese, D.: High performance query answering over DL-Lite ontologies. In: Proceedings of the 13th International Conference on the Principles of Knowledge Representation and Reasoning (KR'12). pp. 308–318 (Jun 2012)
27. Rodríguez-Muro, M., Kontchakov, R., Zakharyaschev, M.: Ontology-based data access: Ontop of databases. In: Alani, H., Kagal, L., Fokoue, A., Groth, P., Biemann, C., Parreira, J.X., Aroyo, L., Noy, N., Welty, C., Janowicz, K. (eds.) The Semantic Web – ISWC 2013. LNCS, vol. 8218, pp. 558–573. Springer Berlin Heidelberg (2013)
28. Rosati, R., Almatelli, A.: Improving query answering over DL-Lite ontologies. In: Proceedings of the 12th Conference on Principles of Knowledge Representation and Reasoning Conference (KR'10) (May 2010)
29. Sirin, E., Parsia, B., Grau, B.C., Kalyanpur, A., Katz, Y.: Pellet: A practical OWL-DL reasoner. J. Web Semantics 5(2), 51–53 (2007)
30. SPARQL Working Group, W. (ed.): SPARQL 1.1 Overview. W3C Recommendation (21 March 2013), available at http://www.w3.org/TR/sparql11-overview/
31. Stoilos, G., Grau, B.C., Horrocks, I.: How incomplete is your semantic web reasoner? In: Proceedings of the Twenty-Fourth AAAI Conference on Artificial Intelligence (AAAI'10). pp. 1431–1436. AAAI Press (2010)