

A Benchmark for OWL 2 DL Reasoners

Gunjan Singh

gunjans@iiitd.ac.in

Knowledgeable Computing and Reasoning Lab, IIIT-Delhi, India

Sumit Bhatia

sumitbhatia@in.ibm.com

IBM Research AI, New Delhi, India

Raghava Mutharaju

raghava.mutharaju@iiitd.ac.in

Knowledgeable Computing and Reasoning Lab, IIIT-Delhi, India

1 INTRODUCTION

An Ontology is a sharable, explicit and formal description of knowledge represented as a set of concepts within a domain and the relationships that hold between them. Web Ontology Language (OWL) is a W3C recommended standard, designed to facilitate ontology development. OWL 2 [2], an extension and revision of the OWL is becoming increasingly popular because of its high level of expressivity. OWL 2 provides several profiles such as OWL 2 Full, OWL 2 DL, OWL 2 EL, OWL 2 QL, and OWL 2 RL. They vary in terms of expressivity and reasoning complexity.

A reasoner is a software that is used to derive facts which are not expressed in the existing ontology explicitly. Typical reasoners provide automated support for reasoning tasks such as ontology consistency, classification, instance checking and query answering. In order to help users select the most suitable reasoner for their domain, it is important to evaluate available reasoners on the basis of their reasoning capability, scalability, efficiency, and storage methods.

With the increasing availability of reasoners for OWL, a number of benchmarks for evaluating OWL reasoners have been developed. LUBM [1] and UOBM [4] are well-known benchmarks that are based on synthetically generated instance data of varying size for university ontology. While LUBM supports only a subset of OWL Lite, UOBM, an extension of LUBM, is OWL DL complete and generates more realistic instance data for scalability testing by providing interlinks between the generated instances. While being widely used, neither of these two benchmarks provide support for OWL 2 semantics that is the need of advanced applications due to its additional expressiveness as compared to OWL. One of the most recent benchmarks, OntoBench [3], generates ontologies on the basis of user selected OWL/OWL2 constructs. OntoBench does not generate instance data along with the ontology (which has only schema information) and thus does not provide support for scalability testing and performance evaluation of reasoners. It only tests the scope of reasoners in terms of coverage of OWL constructs. To this end, no benchmark has been developed that provides support for complete evaluation of reasoners in terms of OWL 2 coverage, scalability testing and performance evaluation.

We address this shortcoming by developing a new benchmark for the evaluation of several well-known OWL 2 DL reasoners such

as FaCT++, Hermit, Pellet etc. The focus of this work is OWL 2 DL, as it is the highest OWL 2 profile (in terms of expressivity) that the benchmark supports.

2 METHODOLOGY

The building blocks of a standard benchmark for OWL 2 DL based reasoners include structure of data, usage of constructs, generation of different sized data, set of queries to test on the generated data and performance metrics for quantitative evaluation of reasoners.

In this work, we extend UOBM so that we can have a standard benchmark for evaluating OWL 2 DL reasoners. First step is to enrich existing University Ontology [4] to support OWL 2 DL constructs, which is done by modifying some of the existing axioms in the ontology and also by adding new classes and properties for some constructs. We also add new classes and properties to provide more interlinks across the universities that helps to improve scalability tests. From this extended ontology we can generate instance data of varying sizes, where user specifies the size by providing the number of universities as input to the benchmark. The instance generator can also be configured to modify the density of each node in the generated graph for specific evaluation. Next step is to construct a standard set of queries to test the performance of reasoners on generated datasets. The queries should be designed in such a way that in order to answer them, the query engine needs to reason over the universities. Since we are focussing on the reasoner performance, each query should cover different type of OWL reasoning tasks such as classification, instance checking etc. The last step is to evaluate existing OWL 2 DL reasoners on generated data against the constructed queries for performance metrics. We have completed generating the instance data of varying size for the extended ontology. We will next be working on designing the queries and evaluating the existing reasoners.

After completing the remaining steps of the OWL 2 DL benchmark, we plan to make the benchmark more configurable by providing the users options to select the OWL constructs of their choice to generate the ontology. For these constructs, users can further specify the *hardness level* of the ontology that the benchmark should generate. Hardness of the ontology can be measured either in terms of the time the reasoner takes or the memory needed or both these factors can be considered.

REFERENCES

- [1] Y. Guo, Z. Pan, and J. Heflin. 2005. LUBM: A Benchmark for OWL Knowledge Base Systems. *Journal of Web Semantics* 3, 2-3 (Oct. 2005), 158–182.
- [2] P. Hitzler, M. Krötzsch, B. Parsia, P. F. Patel-Schneider, and S. Rudolph. 2012. OWL 2 Web Ontology Language Primer (Second Edition). <https://www.w3.org/TR/owl2-primer/>
- [3] V. Link, S. Lohmann, and Haag F. 2016. OntoBench: Generating Custom OWL 2 Benchmark Ontologies. *International Semantic Web Conference* (2016), 122–130.
- [4] L. Ma, Y. Yang, G. Qiu, Z. and Xie, Y. Pan, and S. Liu. 2006. Towards a complete OWL ontology benchmark. *European Semantic Web Conference* (2006), 125–139.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

CoDS COMAD 2020, January 5–7, 2020, Hyderabad, India

© 2020 Association for Computing Machinery.

ACM ISBN 978-1-4503-7738-6/20/01...\$15.00

<https://doi.org/10.1145/3371158.3371223>