

Benchmarking Symbolic and Neuro-Symbolic Description Logic Reasoners

Gunjan Singh¹

¹*Knowledgeable Computing and Reasoning Lab, IIT-Delhi, India*

Abstract

Ontologies are crucial in facilitating data sharing and integration across domains. The Web Ontology Language (OWL and its current version, OWL 2) is widely used to build expressive ontologies that capture complex relationships and semantics. However, the computational complexity of reasoning over OWL 2 ontologies increases with its expressive power. To advance the field of OWL reasoning, standardized benchmarks are needed to identify performance bottlenecks and evaluate reasoning systems. Existing real-world ontologies are limited in their coverage of OWL 2 constructs, necessitating the development of synthetic benchmarks that offer flexibility in testing various aspects of reasoning systems. Our work addresses this need by introducing benchmarks for three types of OWL 2 reasoning systems – symbolic reasoners on static data, stream reasoners, and neuro-symbolic reasoners. These benchmarks facilitate the evaluation and comparison of reasoners' performance, promoting the development of more efficient and effective reasoners.

Keywords

OWL 2, Reasoner, Benchmarking, Neuro-symbolic

1. Introduction

Ontologies enable the sharing and integration of data across different domains, such as health-care, geoscience, IoT, and e-commerce. Web Ontology Language (OWL, and its current version, OWL 2) [1] is a widely used W3C recommended standard for building ontologies that are expressive and can capture complex relationships and semantics. One of the benefits of OWL 2 is that it is based on Description Logics (DLs) [2], a family of logic-based knowledge representation formalisms, which provide a way to represent knowledge in a structured and precise manner¹. This enables automatic reasoning over ontologies in a computationally tractable way. OWL 2 has different profiles, each with varying expressive power, from the relatively simple and tractable profiles such as OWL 2 EL, OWL 2 QL, and OWL 2 RL to the more expressive OWL 2 DL. Expressive ontologies are required to capture complex relationships. However, there is a tradeoff between OWL 2's expressive power and the computational complexity of reasoning. Despite efforts to optimize reasoning methods, current reasoners face challenges in handling large and expressive ontologies effectively [3]. Therefore, there is a need for more advanced

Doctoral Consortium at ISWC 2023 co-located with 22nd International Semantic Web Conference (ISWC 2023)

✉ gunjans@iiitd.ac.in (G. Singh)

🌐 <https://gunjansingh1.github.io/> (G. Singh)

🆔 0000-0003-3171-9088 (G. Singh)



© 2023 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

CEUR Workshop Proceedings (CEUR-WS.org)

¹Since there is a one-to-one correspondence between OWL and DLs, we use those two terms interchangeably.

and efficient reasoning techniques.

1.1. Motivation

To advance the field of ontology reasoning, it is important to evaluate the reasoners on different ontologies and find their performance bottlenecks and improve on them. Benchmarks that can comprehensively evaluate the reasoners can do this job. Several thousands of ontologies that belong to different OWL 2 profiles are available in repositories such as the NCBO Bioportal², AgroPortal³, and the ORE dataset [4]. Although these ontologies can be used to benchmark the reasoners, they do not quite test the limits of the reasoners because, in most cases, they do not involve all the possible OWL 2 constructs and are either not large enough or complex enough. Without scalable and efficient reasoners, ontology developers will not build large and complex ontologies. Without these ontologies, it will be hard to test the performance and scalability of the reasoners. So a synthetic benchmark addresses this chicken-and-egg problem by offering the flexibility to test various aspects of the reasoners by changing the configuration parameters (such as size and complexity). Thus the strengths and weaknesses of different reasoning approaches can be identified. This, in turn, enables the development of more efficient and effective reasoning algorithms.

1.2. Problem Statement

We discuss the requirements for benchmarks in three different types of reasoning systems.

1. **Symbolic Static Data Reasoners.** These are the conventional description logic reasoners that work on static data (ontologies) and focus on reasoning tasks such as consistency checking, classification, and realization. Several reasoners⁴, such as Konclude [5], Openlet⁵, and HermiT [6], have been developed to reason over ontologies efficiently. The performance of these reasoners is usually evaluated in terms of reasoning time taken and memory consumed. The benchmarks that can comprehensively evaluate these reasoners should have the following characteristics.
 - a) Varying size of TBox and ABox axioms. This helps determine the reasoner's ability to handle large ontologies and identify its limits.
 - b) Varying number and type of language constructs. This enables us to understand how different language constructs impact reasoning performance.
 - c) Different combinations of language constructs. This facilitates evaluating the performance of reasoners with different sets of language features, such as ones found in different OWL 2 profiles.
2. **Symbolic Streaming Data Reasoners.** There are several streaming data reasoners such as RSP4J [7] and StreamQR [8] for the real-time processing of streaming data. The key performance indicators for these reasoners typically include latency, throughput, memory usage, completeness, and correctness. A benchmark with the following characteristics,

²<https://bioportal.bioontology.org/>

³<http://agroportal.lirmm.fr/>

⁴<http://owl.cs.manchester.ac.uk/tools/list-of-reasoners/>

⁵<https://github.com/Galigator/openlet>

in addition to the ones discussed for static data reasoners, is required to evaluate their performance.

- a) Realistic streaming data generator. The stream rate should be controllable and inspired by real-world scenarios so that the testing reflects the system's performance in practical settings. It should be possible to vary the number of parallel streams and their rates of frequencies.
 - b) Continuous queries. The queries should be designed to require continuous query answering over streaming data. They should be evaluated for different window sizes and multiple parallel queries. To answer each query, reasoning involving different OWL 2 language constructs is required.
3. **Neuro-Symbolic Reasoners.** With the advancements in automated knowledge base construction⁶, building large and expressive ontologies has become relatively easy. However, these ontologies often suffer from noise and inconsistency, posing challenges for conventional ontology reasoners [9]. To address this, hybrid systems that combine symbolic reasoning with neural networks are being developed. This integration aims to enhance the performance of reasoning systems and tackle challenges such as reasoning with incomplete or uncertain information. Neuro-symbolic reasoning systems can vary in support for different OWL 2 profiles, subsets of description logics, and reasoning tasks (classification, realization, consistency checking, class membership, class subsumption, axiom completion). Although significant developments have taken place in the field of neuro-symbolic reasoning space [10, 11] and advancements are ongoing, there is a need for a common infrastructure and experiment design that will enable developers to evaluate the performance of their systems and compare them with existing systems using standardized performance metrics.

2. Related Work

The benchmarking of OWL 2 reasoners has seen limited development. Prominent benchmarks such as LUBM (Lehigh University Benchmark) [12] and UOBM (University Ontology Benchmark) [13] do not support OWL 2 profiles. OntoBench [14] covers all OWL 2 constructs and profiles but focuses on reasoner coverage rather than scalability. Other than the aforementioned benchmarks, there also exists an open-source java-based ORE benchmark framework⁷ which was a part of OWL Reasoner Evaluation (ORE) Competition [4]. The competition was held to evaluate the performance of OWL 2 complaint reasoners but did not consider performance evaluation in the context of varying ontology sizes or the evaluation of SPARQL query engines with OWL 2 reasoning support. To address these gaps, OWL2Bench [3] was proposed, which is an extension of UOBM, enabling benchmarking of reasoners for different OWL 2 profiles, ABox scalability, and query performance. However, OWL2Bench does not offer scalable TBox and support for customized ontologies. An extension of OWL2Bench [15] describes an ongoing effort towards building such a customizable ontology benchmark for OWL 2 reasoners.

⁶<https://www.akbc.ws/>

⁷<https://github.com/ykazakov/ore-2015-competition-framework>

The existing benchmarks mentioned above are primarily designed for evaluating the performance of static reasoning systems and are not well-suited for stream reasoning systems that handle dynamic data streams. That led to the development of benchmarks in the stream reasoning domain, such as LSBench [16], SRBench [17], CSRBench [18], SLUBM [19], YABench [20], CityBench [21], LASS 1.0 [22], and OWL2Streams [23]. Each of these benchmarks differs in expressivity, supported features, and the types of datasets they use. However, out of these, only LASS 1.0 and OWL2Streams focus on the reasoning tasks. The remaining benchmarks only address continuous query answering under the RDFS entailment regime. However, both LASS 1.0 and OWL2Streams are also limited in scope. LASS 1.0 includes a limited number of OWL 2 RL language constructs, limiting its scope. On the other hand, OWL2Streams proposes three different scenarios for the streaming domain, each focusing on specific requirements but lacking a comprehensive coverage of a knowledge heavy domain that involves modeling extensive knowledge using various OWL 2 constructs, realistic streaming data, and queries. For example, the university domain scenario, adapted from OWL2Bench, lacks highly frequent streams. The Smart City scenario, based on the extension of CityBench, is not knowledge-heavy. The third scenario, based on the Smart Building Covid scenario, is neither expressive nor allows for extensive ABox data. To fill this gap, one possibility is to extend the existing benchmarks with expressive OWL 2 constructs. However, the existing benchmarks are not knowledge heavy. Therefore, there is a need for a benchmark that effectively simulates real-world streaming scenarios, stress tests existing stream reasoners, and incorporates expressive OWL 2 constructs.

To the best of our knowledge, no benchmarks or evaluation frameworks have been designed explicitly to evaluate and compare neuro-symbolic reasoning systems. Most reasoner evaluations are performed on different publicly available ontologies. To evaluate neuro-symbolic reasoners effectively, a benchmark is needed to generate ontologies tailored to specific tasks, allowing for comparing performance and scalability.

3. Contributions and Next Steps

3.1. OWL2Bench

OWL2Bench is an extension of the well-known University Ontology Benchmark (UOBM) [13]. It consists of three major components – a TBox for each OWL 2 profile (EL, QL, RL, and DL), an ABox generator that can generate ABox of varying sizes for the corresponding TBox, and 22 SPARQL queries that involve reasoning. Thus, it allows users to benchmark three aspects of the reasoners – support for different OWL 2 profiles, scalability in terms of ABox size, and query performance. Moreover, the SPARQL queries also enable benchmarking SPARQL query engines that support OWL 2 reasoning. The TBox for each profile was created by enriching UOBM’s university ontology with the supported constructs. Two user inputs are required to generate varying size ABox, the number of universities, and the OWL 2 profile (EL, QL, RL, or DL) of interest. The generated instance data complies with the schema defined in the TBox of the selected profile, and the size depends on the number of universities. For one university, by default, approximately 50,000 ABox axioms are generated.

To demonstrate the utility of OWL2Bench, we ran our benchmark on six reasoners, ELK [24],

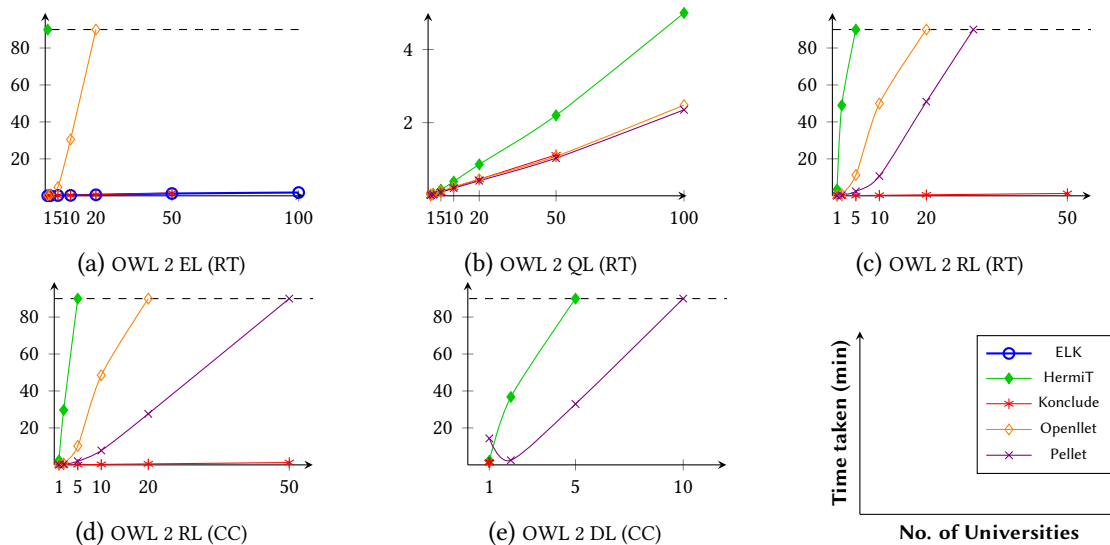


Figure 1: Time taken in minutes (Y-axis) by reasoners on ontologies with varying size represented by the number of universities (X-axis) is given here. The reasoning tasks are Consistency Checking (CC) and Realisation (RT). The dashed line, parallel to X-axis represents the time-out (90 min).

HermiT, JFact⁸, Konclude, Openllet, and Pellet [25] for three reasoning tasks, i.e., consistency checking, classification, and realisation. We also evaluated two SPARQL query engines, Stardog⁹ and GraphDB¹⁰, on SPARQL queries in terms of their loading time and query response time. During our evaluation, we identified possible issues with these systems that need to be fixed and could pave the way for further research in developing reasoners and query engines. For example, there was a huge variation in run-time across the different runs on the same ontology. The inconsistency in the results was reported to the openllet support¹¹. The performance of the reasoners on OWL2Bench is shown in Figure 1. For our experiments, we set the heap space to 24 GB and the time-out to 90 minutes. Most reasoners timed out for even a few universities (except for the QL profile). Although Konclude is much faster, it requires a lot of memory and could not perform any reasoning task after 50 universities. For the EL profile, both Konclude and ELK performed exceptionally well in terms of time taken, but ELK needs lower memory for its computations. For the RL profile, most reasoners timed out on larger ontologies. In the case of OWL 2 DL, Konclude, HermiT, and Pellet were able to complete the consistency checking task only (for 1, 2, and 5 universities, respectively). However, we observed some inconsistency in the results of Pellet. Other evaluations were time-outs. More details about the benchmark and the results are available in the full version of our paper [3].

For the next steps, we are working towards an extension of OWL2Bench [15], which is a customizable benchmark that can generate ontologies based on user-provided inputs such as count and types of language constructs, as well as varying the size of TBox and ABox axioms.

⁸<http://jfact.sourceforge.net/>

⁹<https://www.stardog.com/>

¹⁰<http://graphdb.ontotext.com/>

¹¹<https://github.com/Galigator/openllet/issues/50>

3.2. OWL2StreamBench

We are working on OWL2StreamBench, a benchmark based on tweets from an academic conference event. To push stream reasoning systems to their limits, it is crucial to use diverse and scaled data that reflect real-world situations. Social media platforms like Twitter provide an excellent source of such data, as they generate data at varying frequencies, which is ideal for stream reasoning benchmarks. A wide range of topics and domains are discussed over tweets, allowing users to express their thoughts and opinions on various subjects. Some popular domains for which tweets are posted include news, personal life, and events. Unlike news-related tweets, event-related tweets tend to have a longer lifespan and generate engagement before and during the event. For instance, after the program is announced, there may be a peak of engagement as people register for the event and plan their participation. During the event, attendees may share their experiences, insights, and feedback on social media, generating another peak of engagement. In contrast, news-related tweets tend to generate a peak of engagement quickly after being posted, and the engagement may drop off rapidly. Therefore, we generated tweets related to an academic conference event (ACE) for ABox data. ACE is an ideal domain for benchmarking because it can produce a significant amount of data at varying frequencies, similar to real-world data. Furthermore, tweets generated around ACE also help design continuous queries suitable to replicate real-world scenarios. For example, one query could be continuously monitoring tweets related to papers published on trending academic topics like AI.

Our next steps involve evaluating state-of-the-art stream reasoners using OWL2StreamBench. However, one of the challenges we face that require careful consideration is determining how to evaluate the correctness of these reasoners within the context of stream reasoning. Unlike traditional reasoners, stream reasoners require a customized approach to assess correctness.

3.3. NeSyBench

We are also working on developing a benchmarking suite for neuro-symbolic reasoners. A challenge here is the variation in support for different OWL 2 profiles and reasoning tasks among the neuro-symbolic reasoners. Designing benchmarks that cover a wide range of profiles and tasks is crucial for effective evaluation and comparison. Another challenge is the diversity in techniques these reasoners use, such as neural language models for ontology completion [10] and deep learning for emulating deductive reasoning [11]. This involves considering appropriate evaluation metrics. Overcoming these challenges is essential for a robust and reliable evaluation framework that enables accurate assessments and comparisons of neuro-symbolic reasoners.

The proposed steps for NeSyBench involve examining the current state-of-the-art in neuro-symbolic reasoning and existing benchmarks to identify desired evaluation features. These will include generating profile-based axioms, addressing dataset biases, and representing input axioms for neural network architectures. The performance evaluation will focus on assessing the support for expressive logic, transferability, reasoning time, memory consumption, accuracy, soundness, and completeness, in comparison to symbolic reasoning counterparts. Additionally, the capabilities of neuro-symbolic reasoners to handle noisy and inconsistent ontologies will be evaluated.

4. Conclusion

We discussed the need for benchmarks for three types of OWL 2 reasoners – conventional static data reasoning, stream reasoning, and neuro-symbolic reasoning. We provided an overview of our ongoing efforts in developing three benchmark frameworks – OWL2Bench, OWL2StreamBench, and NeSyBench. Through these benchmarks, we aim to facilitate the evaluation, comparison, and enhancement of reasoning systems. This, in turn, facilitates the progress of ontological reasoning research.

Acknowledgements. I am grateful to my supervisors, Dr. Raghava Mutharaju and Dr. Sumit Bhatia, for their guidance and support. I would also like to acknowledge the partial support of the Infosys Center for Artificial Intelligence (CAI), IIIT-Delhi.

References

- [1] B. C. Grau, I. Horrocks, B. Motik, B. Parsia, P. F. Patel-Schneider, U. Sattler, OWL 2: The next step for OWL, *J. Web Semant.* 6 (2008) 309–322.
- [2] M. Krötzsch, F. Simancik, I. Horrocks, A Description Logic Primer, CoRR abs/1201.4089 (2012). URL: <http://arxiv.org/abs/1201.4089>. arXiv:1201.4089.
- [3] G. Singh, S. Bhatia, R. Mutharaju, OWL2Bench: A Benchmark for OWL 2 Reasoners, in: *The Semantic Web - ISWC 2020 - 19th International Semantic Web Conference*, volume 12507 of *Lecture Notes in Computer Science*, Springer, 2020, pp. 81–96.
- [4] B. Parsia, N. Matentzoglou, R. S. Gonçalves, B. Glimm, A. Steigmiller, The OWL reasoner evaluation (ORE) 2015 competition report, *J. Autom. Reason.* 59 (2017) 455–482.
- [5] A. Steigmiller, T. Liebig, B. Glimm, Konclude: System description, *Journal of Web Semantics.* 27 (2014) 78–85.
- [6] B. Glimm, I. Horrocks, S. Motik, B., Z. G., Wang, Hermit: An OWL 2 Reasoner, *Journal of Automated Reasoning.* 53 (2014) 245–269.
- [7] R. Tommasini, P. Bonte, F. Ongenaes, E. D. Valle, RSP4J: an API for RDF stream processing, in: *18th International Conference, ESWC 2021, Virtual Event, June 6-10, 2021, Proceedings*, volume 12731 of *Lecture Notes in Computer Science*, Springer, 2021, pp. 565–581.
- [8] K. Mamouras, M. Raghothaman, R. Alur, Z. G. Ives, S. Khanna, Streamqre: modular specification and efficient evaluation of quantitative queries over streaming data, in: *Proceedings of the 38th ACM SIGPLAN Conference on Programming Language Design and Implementation, PLDI 2017, ACM, 2017*, pp. 693–708.
- [9] M. K. Sarker, L. Zhou, A. Eberhart, P. Hitzler, Neuro-Symbolic Artificial Intelligence: Current Trends, CoRR abs/2105.05330 (2021). arXiv:2105.05330.
- [10] J. Chen, P. Hu, E. Jiménez-Ruiz, O. M. Holter, D. Antonyrajah, I. Horrocks, OWL2Vec*: Embedding of OWL ontologies, *Machine Learning* 110 (2021) 1813–1845.
- [11] A. Eberhart, M. Ebrahimi, L. Zhou, C. Shimizu, P. Hitzler, Completion Reasoning Emulation for the Description Logic \mathcal{EL}^+ , in: *Proceedings of the AAAI 2020 Spring Symposium on Combining Machine Learning and Knowledge Engineering in Practice, AAAI-MAKE 2020, Volume I*, volume 2600 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2020.

- [12] Y. Guo, Z. Pan, J. Heflin, LUBM: A benchmark for OWL knowledge base systems, *J. Web Semant.* 3 (2005) 158–182.
- [13] L. Ma, Y. Yang, Z. Qiu, G. T. Xie, Y. Pan, S. Liu, Towards a complete OWL ontology benchmark, in: *The Semantic Web: Research and Applications, 3rd European Semantic Web Conference, ESWC 2006*, volume 4011 of *Lecture Notes in Computer Science*, Springer, 2006, pp. 125–139.
- [14] V. Link, S. Lohmann, F. Haag, Ontobench: Generating custom OWL 2 benchmark ontologies, in: *15th International Semantic Web Conference (ISWC), Proceedings, Part II*, volume 9982 of *Lecture Notes in Computer Science*, 2016, pp. 122–130.
- [15] G. Singh, A. Kumar, K. Bhagat, S. Bhatia, R. Mutharaju, OWL2Bench: Towards a Customizable Benchmark for OWL 2 Reasoners, in: *Proceedings of the ISWC 2020 Demos and Industry Tracks: 19th International Semantic Web Conference (ISWC 2020)*, volume 2721 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2020, pp. 344–349.
- [16] D. L. Phuoc, M. Dao-Tran, M. Pham, P. A. Boncz, T. Eiter, M. Fink, Linked stream data processing engines: Facts and figures, in: *11th International Semantic Web Conference (ISWC)*, volume 7650 of *Lecture Notes in Computer Science*, Springer, 2012, pp. 300–312.
- [17] Y. Zhang, M. Pham, Ó. Corcho, J. Calbimonte, Srbench: A streaming RDF/SPARQL benchmark, in: *The Semantic Web - ISWC 2012 - 11th International Semantic Web Conference*, volume 7649 of *Lecture Notes in Computer Science*, Springer, 2012, pp. 641–657.
- [18] D. Dell’Aglío, J. Calbimonte, M. Balduini, Ó. Corcho, E. D. Valle, On correctness in RDF stream processor benchmarking, in: *12th International Semantic Web Conference (ISWC)*, volume 8219 of *Lecture Notes in Computer Science*, Springer, 2013, pp. 326–342.
- [19] T. N. Nguyen, W. Siberski, SLUBM: an extended LUBM benchmark for stream reasoning, in: *Proceedings of the 2nd International Workshop on Ordering and Reasoning, OrdRing 2013, Co-located with the 12th International Semantic Web Conference (ISWC 2013)*, volume 1059 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2013, pp. 43–54.
- [20] M. Kolchin, P. Wetz, E. Kiesling, A. M. Tjoa, Yabench: A comprehensive framework for RDF stream processor correctness and performance assessment, in: *Web Engineering - 16th International Conference, ICWE 2016*, volume 9671 of *Lecture Notes in Computer Science*, Springer, 2016, pp. 280–298.
- [21] M. I. Ali, F. Gao, A. Mileo, Citybench: A configurable benchmark to evaluate RSP engines using smart city datasets, in: *14th International Semantic Web Conference (ISWC)*, volume 9367 of *Lecture Notes in Computer Science*, Springer, 2015, pp. 374–389.
- [22] R. Tommasini, M. Balduini, E. D. Valle, Towards a benchmark for expressive stream reasoning, in: *Joint Proceedings of the 2nd RDF Stream Processing (RSP 2017) and the Querying the Web of Data (QuWeDa 2017) Workshops co-located with 14th ESWC 2017 (ESWC 2017)*, volume 1870 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2017, pp. 26–36.
- [23] P. Bonte, F. D. Turck, F. Ongenaë, Towards an evaluation framework for expressive stream reasoning, in: *The Semantic Web: ESWC 2021 Satellite Events*, volume 12739 of *Lecture Notes in Computer Science*, Springer, 2021, pp. 76–81.
- [24] Y. Kazakov, M. Krötzsch, F. Simancik, The incredible ELK - from polynomial procedures to efficient reasoning with ontologies, *J. Autom. Reason.* 53 (2014) 1–61.
- [25] E. Sirin, B. Parsia, B. Cuenca Grau, A. Kalyanpur, Y. Katz, Pellet: A practical OWL-DL reasoner, *Journal of Web Semantics.* 5 (2007) 51–53.