The OWL Reasoner Evaluation (ORE) 2015 Resources

Bijan Parsia[1], Nicolas Matentzoglu[1], Rafael S. Gonçalves[2], Birte Glimm[3], and Andreas Steigmiller[3]

[1] Information Management Group, University of Manchester, United Kingdom
[2] Stanford Center for Biomedical Informatics Research, Stanford University, USA
[3] Institute of Artificial Intelligence, University of Ulm, Germany

Abstract. The OWL Reasoner Evaluation (ORE) Competition is an annual competition (with an associated workshop) which pits OWL 2 compliant reasoners against each other on various standard reasoning tasks over naturally occurring problems. The 2015 competition was the third of its sort and had 14 reasoners competing in six tracks comprising three tasks (consistency, classification, and realisation) over two profiles (OWL 2 DL and EL). In this paper, we outline the design of the competition and present the infrastructure used for its execution: the corpora of ontologies, the competition framework, and the submitted systems. All resources are publicly available on the Web, allowing users to easily re-run the 2015 competition, or reuse any of the ORE infrastructure for reasoner experiments or ontology analysis.

Keywords: OWL, ontologies, reasoning

1 Introduction

The Web Ontology Language (OWL) is in its second iteration (OWL 2) and has seen significant adoption especially in Health Care and Life Sciences. OWL 2 DL can be seen as a variant of the description logic $\mathcal{SROIQ}$, with the various other profiles being either subsets (e.g., OWL 2 EL) or extensions (e.g., OWL 2 Full). Description logics generally are designed to be *computationally practical* so that, even if they do not have tractable worst-case complexity for key services, they nevertheless admit implementations which seem to work well in practice [2]. Unlike in the early days of description logics or even of the direct precursors of OWL (DAML+OIL), the reasoner landscape for OWL is rich, diverse, and highly compliant with a common, detailed specification. Thus, we have a large number of high performance, production-quality reasoners with similar core capacities (with respect to language features and standard inference tasks).

Research on optimising OWL reasoning continues apace, though empirical work still lags both theoretical and engineering work in breadth, depth, and sophistication. There is, in general, a lack of shared understanding of test cases, test scenarios, infrastructure, and experiment design. A common strategy in research communities to help address these issues is to hold competitions, that is, experiments designed and hosted by third parties on an independent (often

constrained, but sometimes expanded) infrastructure. Such competitions (in contrast to published benchmarks) typically do not directly provide strong empirical evidence about the competing tools. Instead, they serve two key functions: 1) they provide a clear, motivating event that helps drive tool development (e.g., for correctness or performance) and 2) *components* of the competition are useful for subsequent research. Finally, competitions can be great fun and help foster a strong community. They can be especially useful for newcomers by providing a simple way to gain some prima facie validation of their tools without the burden of designing and executing complex experiments themselves.

Toward these ends, we have been running a competition for OWL reasoners (with an associated workshop): The OWL Reasoner Evaluation (ORE) competition. ORE has been running, in substantively its current form, for three years, and this year it was held in conjunction with the 28th International Description Logic Workshop (DL 2015)[4] in June 2015. A report on the ORE 2015 competition results and analysis is under submsission [15]. In this paper we focus on the elements of the ORE 2015 that are reusable by the general public. To that end, we describe the design of the 2015 competition, which provides a reasonable default structure for reasoner comparison. We also describe the competition infrastructure: the corpora of ontologies, the competition framework, and the submitted systems. All resources are publicly available on the Web, allowing users to easily re-run the 2015 competition, or reuse any of the ORE infrastructure for reasoner experiments, benchmarks, debugging, or improvement, or for ontology analysis.

2 Competition Design

The ORE competition is inspired by and modeled on the CADE ATP System Competition (CASC) [16,23] which has been running for 25 years and has been heavily influential in the automated theorem proving community[5] (especially for first order logic). The key common elements between ORE and CASC are:

1. A number of distinct tracks/divisions/disciplines characterised by problem type (e.g., "effectively propositional" or "OWL 2 EL ontology").
2. The test problems are derived from a large, neutral, updated yearly set of problems (e.g., for CASC, the TPTP library [22]).
3. Reasoners compete (primarily) on number of problems solved with a tight per problem timeout.

The last point is worth some comment. Most evaluations of reasoner performance in the literature use some form of time in their evaluations (e.g., CPU or wall clock time). This has several advantages, including capturing the primary quantity of interest for most users in most situations. However, it is vulnerable

[4] The websites for DL2015 and ORE2015 are archived at `http://dl.kr.org/dl2015/` and `https://www.w3.org/community/owled/ore-2015-workshop/` respectively.

[5] See the CASC website for details on past competitions: `http://www.cs.miami.edu/~tptp/CASC/`. Also of interest, though not directly inspirational for ORE, is the SAT competition `http://www.satcompetition.org/`

to a number of problems esp. as one starts testing on large numbers of diverse ontologies. For example, one reasoner might perform very well on a large number of small ontologies and comparatively poorly on a few larger ones, whereas another might lose on all the small ones (due to start up overhead) and do much better on the larger ones. Yet, their averages and even medians might be similar.

More critically, timeouts severely distort aggregate statistics about time. If we include timeouts in the statistics, they *crop* times. That is, given a timeout of two hours, we cannot distinguish between a reasoner that would take two hours and one minute from one that would take days. Reasoner errors can cause similar issues. If we drop those times, buggy reasoners seem to do better. Even if we include them, less buggy reasoners that time-out, or just take longer to correctly finish than the buggy reasoners take to hit an error, will be penalised. Measuring problems solved does not fully ameliorate these problems and introduces some new ones, but it seems more robust for simple comparisons. So it serves as a better default experiment design.

As description logics have a varied set of core inference services supported by essentially all reasoners, ORE also has track distinctions based on task (e.g., classification or realisation). ORE 2015 had both a live as well as an offline competition. The offline competition is executed with more relaxed time constraints against user-submitted ontologies, while the live competition is executed with a tight timeout against a corpus of ontologies we constructed.

3 Ontology Corpora

In the following sections we present the publicly available corpora of ontologies constructed for the live competition and the user-submitted ontologies. Ontology pre-processing was done using the OWL API (v3.5.1) [4].

3.1 Live Competition Corpus

The full live competition corpus contains 1,920 ontologies, sampled from three source corpora: A January 2015 snapshot of Bioportal [12] containing 330 biomedical ontologies, the Oxford Ontology Library[6] with 793 ontologies that were collected for the purpose of ontology-related tool evaluation, and MOWLCorp [6], a corpus based on a 2014 snapshot of a Web crawl containing around 21,000 unique ontologies. As a first step, the ontologies of all three source corpora were collected and serialised into OWL/XML with their imports closure merged into a single ontology. The merging is, from a competition perspective, necessary to mitigate the bottleneck of loading potentially large imports repeatedly over the network, and because the hosts of frequently imported ontologies sometimes impose restrictions on the number of simultaneous accesses.[7] After the collection, the entire pool of ontologies is divided into three groups: (1) Ontologies with less

[6] `http://www.cs.ox.ac.uk/isg/ontologies/`

[7] Which may be exceeded considering that all reasoners in the competition run in parallel.

than 50 axioms, (2) OWL 2 DL ontologies, and (3) OWL 2 Full ontologies. The first group was removed from the pool.

As reasoner developers could tune their reasoners towards the ontologies in the three publicly available source corpora, we included a number of approximations into our pool. The entire set of OWL 2 Full ontologies was approximated into OWL 2 DL, i.e., we used a (slightly modified) version of the OWL API profile checker to drop DL profile-violating axioms so that the remainder is in OWL 2 DL [8]. Because of some imperfections in the "DLification" process, this process had to be performed twice. For example, in the first round, the DL expressivity checker may have noted a missing declaration *and* an illegal punning. Fixing this would result in dropping the axiom(s) causing the illegal punning *as well as* injecting the declaration—which could result again in an illegal punning.

The OWL 2 DL group was then approximated into OWL 2 EL using the approximation method employed by TrOWL [17]. As some ontologies are included in more than one of the source corpora, we excluded at this point (as a last preprocessing step) all duplicates[8] from the entire pool of ontologies, and removed ontologies with TBoxes containing less than 50 axioms. This left us with the full competition dataset of 1,920 unique OWL 2 DL ontologies. The full competition corpus can be obtained from Zenodo [9].

3.2 User Submitted Ontologies

We had four user submissions to ORE 2015, consisting of a total of 7 ontologies. The user submissions underwent the same pre-processing procedures as the corpus (Section 3.1). This had occasionally large consequences on the ontologies, most importantly with respect to rules (they were stripped out) and any axiom beyond OWL 2 DL (for example, axioms redefining built-in vocabulary or violating the global constraints on role hierarchies, see [8]).

We make all user submitted ontologies for which we have permission to redistribute, redistributable. Occasionally, we have some user submitted ontologies which are proprietary and so cannot be redistributed. On the one hand, we prefer all ontologies be fully shareable. On the other, we want the widest reach possible. Currently, the number of "restricted" ontologies that have been submitted are very few, so it seems worth the outreach. We do work with submitters to make those ontologies as accessible as possible. Some basic metrics for the ontologies can be found in Table 1. The ontology archive is published on Zenodo, and linked to from `http://owl.cs.manchester.ac.uk/publications/supporting-material/ore-2015-report`.

[8] Duplicates are those that are *byte identical* after being "DLified" and serialised into Functional Syntax.

[9] `https://code.google.com/archive/p/dinto/`. Submitted by María Herrero, Computer Science Department, Univesidad Carlos III de Madrid. Leganés, Spain.

[10] `https://github.com/obophenotype/cell-ontology`. Submitted by Dr. David Osumi-Sutherland, GO Editorial Office, European Bioinformatics Institute, European Molecular Biology Laboratory, Cambridge, UK.

Table 1: Breakdown of user-submitted ontologies in the ORE 2015 corpus

Ontology name	#Axioms	OWL	Expressivity
Drug-Drug Interactions Ontology (DINTO)[9]	123,930	Pure DL	$\mathcal{ALCRIQ}(\mathcal{D})$
Cell Ontology (CO)[10]	7,527	Pure DL	$\mathcal{SRI}$
Drosophila Phenotype Ontology (DPO)[11] [13]	917	Pure DL	$\mathcal{SRIF}(\mathcal{D})$
Gene Ontology Plus (GO+)[12]	150,955	Pure DL	$\mathcal{SRI}$
Virtual Fly Brain Ontologies (VFB-KB)[13]	168,183	Pure DL	$\mathcal{SRI}$
Virtual Fly Brain Ontologies Ext. 1 (VFB-EPNT)	96,907	Pure DL	$\mathcal{SRI}$
Virtual Fly Brain Ontologies Ext. 2 (VFB-NCT)	96,907	Pure DL	$\mathcal{SRI}$

4 Competing Systems

There were 14 reasoners submitted with 11 purporting to cover OWL 2 DL, and 3 being OWL 2 EL specific. The set of competing systems (as submitted) is available on the Web.[14] In Table 2, we briefly summarize the participating reasoning systems. More detailed information about each reasoner can be found online[15] as well as in our recently conducted OWL reasoner survey [7]. The version information reflect the state of the system as it was submitted to ORE.

Table 2: Reasoners submitted to the ORE 2015 competition.

Reasoner	Version	Language	Maintained by
Chainsaw [26]	1.0	OWL 2 DL	University of Manchester, UK
ELepHant [19]	0.5.7	OWL 2 EL	Not given
ELK [5]	0.5.0	OWL 2 EL	University of Ulm, Germany
FaCT++ [25]	1.6.4	OWL 2 DL	University of Manchester, UK
HermiT [1]	1.3.8.5	OWL 2 DL	University of Oxford, UK
jcel [10]	0.21.0	OWL 2 EL	Technische Universität Dresden, Germany
Jfact [14]	4.0.1	OWL 2 DL	University of Manchester, UK
Konclude [21]	0.6.1	OWL 2 DL	University of Ulm, derivo GmbH, Germany
MORe [18]	0.1.6	OWL 2 DL	University of Oxford, UK
PAGOdA [27]	-	OWL 2 DL	University of Oxford, UK
Pellet (OA4) [20]	2.4.0	OWL 2 DL	Complexible (Original version)
Racer [3]	2.0	OWL 2 DL	Concordia University, Montreal, Canada
TrOWL [24]	1.5	OWL 2 DL	University of Aberdeen, UK

5 Test Framework

The test framework used in ORE 2015 is a slightly modified version of the one used for ORE 2014, which is implemented in Java, open sourced under the LGPL license, and versioned and distributed on Github.[16]

[11] `https://github.com/FlyBase/flybase-controlled-vocabulary`.
[12] `http://bioportal.bioontology.org/ontologies/GO-PLUS`.
[13] `https://github.com/VirtualFlyBrain`.
[14] See links in supplementary materials website: `http://owl.cs.manchester.ac.uk/publications/supporting-material/ore-2015-report`.
[15] `http://owl.cs.manchester.ac.uk/tools/list-of-reasoners/`.
[16] `https://github.com/andreas-steigmiller/ore-competition-framework/`.

The framework takes a "script wrapper" approach to execute reasoners, instead of, for example, requiring all reasoners to use (a specific version of) the OWL API. While this puts some extra burden on established reasoners with good OWL API bindings, this, combined with the requirement only to handle *some* OWL 2 standard syntax (with the very easy to parse and serialise functional syntax [11] as a common choice), makes it very easy for new reasoners to participate even if they are written in hard-to-integrate with the JVM languages. There is a standard script for OWL API based reasoners so it is fairly trivial to prepare an OWL API wrapped reasoner for competition. On the other hand, this is not necessarily a desirable outcome as encouraging reasoners to provide good OWL API support (thus supporting access to those reasoners by the plethora of tools which use the OWL API) is an outcome we want to encourage.

Reasoners report times, results, and any errors through the invocation script. Times are in wall clock time (CPU time is inappropriate because it will penalise parallel reasoners) and exclude "standard" parsing and loading of problems (i.e., without significant processing of the ontology). The framework enforces (configurable) timeouts for each reasoning problem. Results are validated by comparison between competitors with a majority vote/random tie-breaking fallback strategy.

The framework supports both serial and parallel execution of a competition. Parallel distributed mode is used for the live competition, but serial mode is sufficient for testing or offline experiments. The framework also logs sufficient information to allow "replaying" the competition, and includes scripts for a complete replay as well as jumping to the final results.

5.1 Usage

The framework project can be cloned with git,[17] after which users can use the *build-evaluator* script to build the project with Maven.[18] Subsequently, the configuration work will primarily focus on the *data* folder (the paragraphs below discuss folders within *data*), and minor changes to the scripts in the *scripts* folder may be necessary to modify the amount of memory allocated to the JVM.

Global settings The global settings of the framework are defined in a file in the *configs* folder. Possible settings include the timeout (in milliseconds) for the execution each reasoner, the processing timeout (used to cut the reported processing time of the reasoners for the evaluation), the memory limit, output options, and execution options such as forcing the competition to only take place if there exists one client-machine available for each reasoner.

Competition settings The competition settings are defined in files in the *competitions* folder. The key competition settings are its name, output folder, list of participating reasoners, query folder, and execution and processing timeouts.

[17] `https://git-scm.com/`
[18] `http://maven.com`

Reasoner settings Each reasoner under test needs to be accompanied with two elements: a starter script that the framework executes to start the reasoner, and a configuration file that defines the reasoner name, response output folder, starter script location, accepted ontology format, supported OWL 2 profiles, and whether the reasoner supports datatypes and rules. Multiple versions of the same reasoner can be benchmarked, so long as their respective configuration files differ in name and input-output information.

Inputs The inputs for evaluating a competition are: a corpus of ontologies (which should be placed in the *ontologies* folder), and a collection of reasoners configured as above (each of which should be a folder within the *reasoners* folder). Next, the framework needs the queries that are meant to be evaluated in the benchmark; these files specify the reasoning task, and the ontology location, its profile(s), and whether it contains rules or datatypes. Query files are generated by the framework using the appropriate *create-queries* scripts for the task.

There is further documentation on the framework's GitHub repository.

6 Conclusion

The ORE 2015 Reasoner Competition continues the success of its predecessors. And with it, the general public can benefit from the resources we presented here for their own experimentation. The ORE 2015 corpus, whether used with the ORE framework or in a custom test harness, is a significant and distinct corpus for reasoner experimentation. Developers can easily rerun this year's competition with new or updated reasoners to get a sense of their relative progress and we believe that solving all the problems in that corpus in similar or somewhat relaxed time constraints is a reliable indicator of a very high quality implementation.

Ideally, the ORE toolkit and corpora will serve as a nucleus for an infrastructure for common experimentation. To that end, every relevant resource (from corpora to test framework) has been appropriately published, and where appropriate versioned, on the Web. The test harness seems perfectly well suited for black box head-to-head comparisons, and we recommend experimenters consider it before writing a home grown one. This will improve the reliability of the test harness as well as reproducibility of experiments. Even for cases where more elaborate internal measurements are required, the ORE harness can serve as the command and control mechanism. For example, separating actual calculus activity from other behavior (e.g., parsing) requires a deep delve into the reasoner internals. However, given a set of reasoners that could separate out those timings, it would be a simple extension to the harness to accommodate them.

References

1. Glimm, B., Horrocks, I., Motik, B., Stoilos, G., Wang, Z.: HermiT: An OWL 2 Reasoner. J. of Automated Reasoning 53(3), 245–269 (2014)
2. Gonçalves, R.S., Matentzoglu, N., Parsia, B., Sattler, U.: The empirical robustness of description logic classification. In: Proc. of ISWC (2013)

3. Haarslev, V., Hidde, K., Möller, R., Wessel, M.: The RacerPro knowledge representation and reasoning system. Semantic Web J. 3(3), 267–277 (2012)
4. Horridge, M., Bechhofer, S.: The OWL API: A Java API for OWL ontologies. Semantic Web J. 2(1), 11–21 (2011)
5. Kazakov, Y., Krötzsch, M., Simancik, F.: The incredible ELK - From polynomial procedures to efficient reasoning with EL ontologies. J. of Automated Reasoning 53(1), 1–61 (2014)
6. Matentzoglu, N., Bail, S., Parsia, B.: A Snapshot of the OWL Web. In: Proc. of ISWC (2013)
7. Matentzoglu, N., Leo, J., Hudhra, V., Sattler, U., Parsia, B.: A Survey of Current, Stand-alone OWL Reasoners. In: Proc. of ORE (2015)
8. Matentzoglu, N., Parsia, B.: The OWL Full/DL Gap in the Field. In: Proc. of OWLED (2014)
9. Matentzoglu, N., Parsia, B.: ORE 2015 reasoner competition dataset (2015), `http://dx.doi.org/10.5281/zenodo.18578`
10. Mendez, J.: jcel: A modular rule-based reasoner. In: Proc. of ORE (2012)
11. Motik, B., Patel-Schneider, P.F., Parsia, B.: OWL 2 Web Ontology Language: Structural specification and functional-style syntax. W3C recommendation (2009)
12. Noy, N.F., Shah, N.H., Whetzel, P.L., Dai, B., Dorf, M., Griffith, N., Jonquet, C., Rubin, D.L., Storey, M.A., Chute, C.G., Musen, M.A.: BioPortal: ontologies and integrated data resources at the click of a mouse. Nucleic Acids Research 37, 170–173 (2009)
13. Osumi-Sutherland, D., Marygold, S.J., Millburn, G.H., McQuilton, P., Ponting, L., Stefancsik, R., Falls, K., Brown, N.H., Gkoutos, G.V.: The drosophila phenotype ontology. J. of Biomedical Semantics (2013)
14. Palmisano, I.: JFact repository (2015), `https://github.com/owlcs/jfact`
15. Parsia, B., Matentzoglu, N., Gonçalves, R.S., Glimm, B., Steigmiller, A.: The OWL reasoner evaluation (ORE) 2015 competition report. J. of Automated Reasoning (2016), in submission
16. Pelletier, F., Sutcliffe, G., Suttner, C.: The Development of CASC. AIC 15(2-3), 79–90 (2002)
17. Ren, Y., Pan, J.Z., Zhao, Y.: Soundness preserving approximation for TBox reasoning. In: Proc. of AAAI (2010)
18. Romero, A.A., Cuenca Grau, B., Horrocks, I.: MORe: Modular combination of OWL reasoners for ontology classification. In: Proc. of ISWC (2012)
19. Sertkaya, B.: The ELepHant reasoner system description. In: Proc. of ORE (2013)
20. Sirin, E., Parsia, B., Cuenca Grau, B., Kalyanpur, A., Katz, Y.: Pellet: A practical OWL-DL reasoner. J. of Web Semantics 5(2), 51–53 (2007)
21. Steigmiller, A., Liebig, T., Glimm, B.: Konclude: System description. J. of Web Semantics 27, 78–85 (2014)
22. Sutcliffe, G.: The TPTP Problem Library and Associated Infrastructure: The FOF and CNF Parts, v3.5.0. J. of Automated Reasoning 43(4), 337–362 (2009)
23. Sutcliffe, G., Suttner, C.: The State of CASC. AIC 19(1), 35–48 (2006)
24. Thomas, E., Pan, J.Z., Ren, Y.: TrOWL: Tractable OWL 2 reasoning infrastructure. In: Proc. of ESWC (2010)
25. Tsarkov, D., Horrocks, I.: FaCT++ description logic reasoner: System description. In: Proc. of IJCAR (2006)
26. Tsarkov, D., Palmisano, I.: Chainsaw: a Metareasoner for Large Ontologies. In: Proc. of ORE (2012)
27. Zhou, Y., Nenov, Y., Grau, B.C., Horrocks, I.: Pay-as-you-go OWL query answering using a triple store. In: Proc. of AAAI (2014)