The OWL Reasoner Evaluation (ORE) 2015 Competition Report

Bijan Parsia, Nicolas Matentzoglu, Rafael Gonçalves, Birte Glimm, and Andreas Steigmiller

{bijan.parsia, nicolas.matentzoglu}@manchester.ac.uk, rafael.goncalves@stanford.edu, {birte.glimm, andreas.steigmiller}@uni-ulm.de

Abstract. The OWL Reasoner Evaluation (ORE) Competition is an annual competition (with associated) workshop which pits OWL 2 compliant reasoners against each other on various standard reasoning tasks against corpora. The 2015 competition was the third of its sort and had 14 reasoners competing in 6 tracks comprising 3 tasks (consistency, classification, and realisation) over two profiles (OWL 2 DL and EL). In this paper, we discuss the design, execution and results of the 2015 competition with particular attention to lessons learned for benchmarking, comparative experiments, and future competitions.

1 Introduction

The Web Ontology Language (OWL) is in its second iteration (OWL 2) and has seen significant adoption especially in the 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[1] 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 the early days of description logics or even of the direct precursors of OWL (DAML+OIL), the reasoner landscape [17, 8] for OWL is rich, diverse, and highly compliant with the OWL spec. 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 theoretical and engineering work in breath, depth, and sophistication. There is, in general, a lack of shared understanding of test cases, test scenarios, infrastructure, or 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

[1] Some related standardised logics are subsets and extensions (e.g., RDFS) which are proper subsets of OWL 2 Full.

about the competing tools. Instead, they serve two key functions: 1) they provide a clear, motivating event that helps drive tools 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 in this paper we describe the 2015 competition (held in conjunction with the 28th International Description Logic Workshop (DL2015)[2] in June 2015).

2 Competition Design

The ORE competition is inspired by and modelled on the CADE ATP System Competition (CASC) [14, 22] which has been running for 25 years and been heavily influential in the automated theorem proving community[3] (esp. for first order logic).

Key common elements:

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 [21]).
3. Reasoners compete (primarily) on number of problems solved with a tight per problem timeout.

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). Other CASC inspired elements:

1. ORE 2015 consisted entirely of a "live" competition run during the DL workshop.
2. There was a secondary competition among DL attendees to predict the results for various reasoners.
3. Competitors and organisers were given custom designed t-shirts.

We observe that central to such competitions is participation, thus various incentives to participate are critical especially in the early years of the competition as it is trying to get established. Hence the importance of "fun" elements, incentives (e.g., prizes, bragging rights), as well as a reasonable chance of winning at least *something*.

[2] 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.

[3] 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//`

2.1 Tracks

ORE 2015 had 6 tracks based on three central reasoning services (consistency, classification, and realisation) and two OWL profiles (OWL 2 DL and EL). Classification is, almost certainly, the most common and important reasoning service for ontologies to date. Consistency is, in some sense, the most fundamental service. Realisation gets us at least a minimal form of instance reasoning. These services are not ubiquitously supported, with realisation not handled by some reasoners. These have standard definitions (though any consequence equivalent definition would do):

- An ontology $\mathcal{O}$ is *consistent* if $\mathcal{O} \not\models \top \sqsubseteq \bot$ and inconsistent otherwise.
- The classification of an ontology $\mathcal{O}$ ($Cl(\mathcal{O})$) is $\{\alpha | \alpha = A \sqsubseteq B; A, B \in N_c \cup \{\bot, \top\}; \mathcal{O} \models \alpha\}$ where N_c is the set of class names in $\mathcal{O}$.
- The realisation of an ontology $\mathcal{O}$ ($Rl(\mathcal{O})$) is $\{\alpha | \alpha = A(x); x \in N_i, A \in N_c; \mathcal{O} \models \alpha\}$ where N_c is the set of class names, N_i is the set of individual names in $\mathcal{O}$.

We split out a track into a tractable profile when we have enough participants which are specifically tuned for that profile. In prior years we have had an RL and QL track, but the number of RL and QL specific reasoners is very low. We believe this is, in part, due to the fact that RL and QL users tend to be conjunctive query oriented. We hope to introduce a conjunctive query track in future years, but see the discussion below for some of the challenges there. All reasoners purporting to handle the entirety of OWL 2 DL are entered in all tracks. Thus we have specialised EL reasoners competing against complete OWL DL reasoners.

For each track, we award prizes to the top three participants for a total of 18 possible winners.

2.2 Corpus

The full 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[4] with 793 ontologies that were collected for the purpose of ontology related tool evaluation and MOWLCorp [7], a corpus based on a 2014 snapshot of a Web-Crawl containing around 21K unique ontologies. Each competition comes with its own random stratified sample of ontologies from this base corpus - this means that not all 1,920 ontologies actually made it into the live competition. Ontology processing was done using the OWL API (3.5.1) [4].

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

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

because the hosts of frequently imported ontologies sometimes impose restrictions on the number of simultaneous accesses.[5] After the collection, the entire pool of ontologies is divided into three groups: (1) Ontologies with less than 50 axioms, (2) OWL 2 DL ontologies, (3) OWL 2 Full ontologies. The first group is removed from the pool. As reasoner developers may chose to 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 enough axioms so that the remainder is in OWL 2 DL. As some degree of OWL Fullness comes from illegal axiom interaction,[6] we repeated the "DLification" process twice. The OWL DL group was then approximated into OWL 2 EL and OWL 2 QL, using the approximation method employed by TrOWL [15]. As the only syntax that is uniformly supported by all reasoners participating the competition, we then serialised the current pool (including the original OWL 2 DL ontologies, the EL/QL-approximated ontologies and the "DLified" OWL 2 Full ontologies) into Functional Syntax, and gathered all relevant ontology metrics again. 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 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 random stratified sampling for the competition then was done as follows: All ontologies were binned by size into the following groups: Very small (50-99 axioms), small (100-999 axioms), medium (1,000-9,999 axioms), large (10,000-100,000 axioms) and very large (more than 100,000 axioms). From each group, we attempted to sample 60 original ontologies, and 15 approximated ones for each competition. For the OWL 2 EL related track, the ontologies had to fall under the OWL 2 EL profile, for the OWL 2 DL competition the ontologies had to fall under OWL 2 DL but *not* under any of the three OWL 2 profiles, and for the two realisation challenges we only considered those ontologies that had at least 100 ABox axioms. This process resulted in the following six live competition corpora: 109 for OWL 2 EL realisation, 298 for OWL 2 EL classification and consistency, 264 for DL realisation and 306 for DL consistency and classification.

Figures 1 and 2 show the ontology sizes in terms of axiom counts and the usage of constructs through the corpus.

The full competition corpus, and the execution order of the competition, can be obtained from Zenodo [9].

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

[6] For example, an added declaration might introduce an illegal punning.

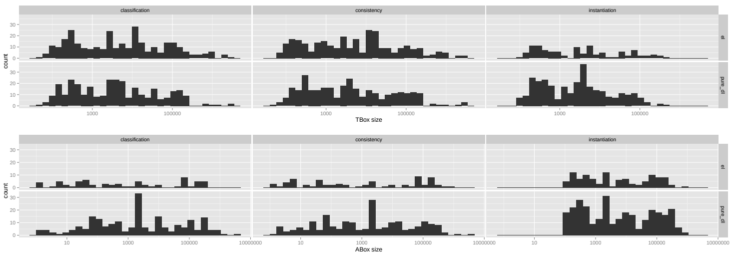

Fig. 1. Ontology size distribution of the full competition corpus.

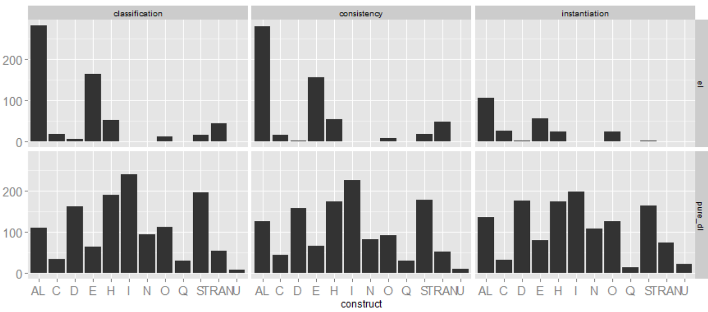

Fig. 2. Ontology size distribution of the full competition corpus.

2.3 Test Framework and Environment

The test framework used in ORE 2015 is a slightly modified version of the one used for ORE 2014 which is open sourced under the LGPL and available on Github.[7]

The framework takes a "script wrapper" approach to running 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 fairly common choice), makes it very easy for new reasoners to participate even if they are written in hard-to-integrate with the JVM languages. The OWL API also is a very rich and rather heavyweight framework that is not tightly integrated with most reasoners. For example, systems using the OWL API generally consume more memory because they maintain the OWL API level representation of the ontology and the reasoner internal one. Thus, avoiding the OWL API can help competition performance. However, there is a standard script for OWL API based reasoners so it is fairly trivial to prepare an OWL API wrapped reasoner for competition.

[7] `https://github.com/andreas-steigmiller/ore-2014-competition-framework/`. A detailed description of the framework and how to run it is available there.

However, 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. Note, unlike CASC, we do not require reasoners to produce proofs of their results as this is not a standard feature of description logic reasoners and for many services (such as classification) it may be impractical. We are however experimenting with a more satisfactory justification-based technique for disagreement resolution [6] for future competitions.

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.

The competition was run on a cluster of 19 machines: 1 master machine that dispatched reasoners with problems to the 18 client machine as well as collecting and serving up results to a live display. Each machine sported an Intel Xeon 4-Core L5410 running at 2.33GHz with 12GB of RAM, for which 2GB were reserved for the operating system (i.e., 10GB could be used by the reasoners). The operating system was Ubuntu 14.04.02 LTS and the Java version was OpenJDK v1.7.0 64-bit. The reasoner execution was limited to 180s for each ontology in each track, where only 150s were allowed for reasoning and 30s could additionally be used for parsing and writing results in order to reduce the penalisation of reasoners with slow parsers. Hence, only if the time reported by the reasoner exceeded 150s was it interpreted as a timeout.

3 Participants

There were 14 reasoners participating with 11 purporting to cover OWL 2 DL and 3 being OWL EL specific, see Table 1. There is no specific penalty or test for being incomplete with respect to a profile and, indeed, one reasoner, TrOWL is intentionally incomplete for performance reasons.

The number of participants is fairly stable over the past three years ranging from 11 to 14. There is a stable core of participants with some fluctuation on the margin. Some reasoners are not entered by their original developers (e.g., Pellet) and ORE currently has no policy against that. We anticipate in the future that more coalition reasoners will be made available, though currently only MORe and Chainsaw use component reasoners (ELK and HermiT the former, FaCT++ the latter) which are also competing. MORe's coalition involves partitioning the ontology into an EL and DL part, dispatching each part to the respective tuned reasoner, and combining the results [16, 25]. Coalition reasoners that do not transform the ontology in any relevant way will need special consideration if they arrive.

Given the presence of deliberately incomplete (with respect to their purported profile) reasoners we are considering whether to modify the voting procedure to discount those reasoners' votes in certain cases. A full break-down of all tracks and competing reasoners can be seen in Table 2.

Table 1. Participant list

Reasoner	Profile supported	New to ORE?
Chainsaw [25]	DL	No
ELepHant [18]	EL	No
ELK [5]	EL	No
FaCT++ [24]	DL	No
HermiT[8] [1]	DL	No
jcel [10]	EL	No
Jfact [13]	DL	No
Konclude [20]	DL	No
MORe [16]	DL	No
PAGOdA [26]	DL	Yes
Pellet [19]	DL	Yes
Racer [3]	DL	Yes
TrOWL [23]	DL	No

Table 2. Breakdown of the competition by track.

Task	Language	Competitors	Problems
Consistency	OWL 2 EL	13	298
	OWL 2 DL	10	306
Realisation	OWL 2 EL	12	109
	OWL 2 DL	10	264
Classification	OWL 2 EL	13	298
	OWL 2 DL	10	306

4 Results

Results, error reports and more details on the competition framework are available at `http://dl.kr.org/ore2015`. Figure 3 shows the results of all partici-

[8] HermiT was submitted with OWL API 3 and OWL API 4 bindings

pants in all tracks as displayed during the live competition. During the competition, these charts are dynamically updated as problems are being solved and reported.

Discipline: OWL EL Realisation (finished)

Rank	Reasoner	Progress	Score	!	Time
1	Konclude		104 / 109	5	229.9 s
2	ELK		102 / 109	7	277.8 s
3	TrOWL		86 / 109	23	242.3 s
4	PAGOdA		86 / 109	23	1,771.7 s
5	ELepHant		84 / 109	25	424.8 s
6	FaCT++		79 / 109	30	354.2 s
7	JFact		63 / 109	46	280.7 s
8	Pellet-OA4		60 / 109	49	1,154.3 s
9	HermiT		57 / 109	52	905.1 s
10	HermiT-OA4		57 / 109	52	934.4 s
11	Chainsaw		43 / 109	66	251.9 s
12	Racer		32 / 109	77	518.8 s

Discipline: OWL DL Realisation (finished)

Rank	Reasoner	Progress	Score	!	Time
1	Konclude		247 / 264	17	739.3 s
2	FaCT++		172 / 264	92	1,111.3 s
3	HermiT		163 / 264	101	2,934.9 s
4	HermiT-OA4		162 / 264	102	3,022.5 s
5	TrOWL		150 / 264	114	503.5 s
6	Pellet-OA4		136 / 264	128	1,434.2 s
7	JFact		109 / 264	155	1,252.6 s
8	PAGOdA		104 / 264	160	3,437.5 s
9	Chainsaw		79 / 264	185	1,067.6 s
10	Racer		46 / 264	218	294.8 s

Discipline: OWL EL Classification (finished)

Rank	Reasoner	Progress	Score	!	Time
1	ELK		298 / 298	0	674.1 s
2	Konclude		294 / 298	4	622.3 s
3	MORe		294 / 298	4	1,685.1 s
4	ELepHant		291 / 298	7	957.0 s
5	TrOWL		275 / 298	23	767.4 s
6	HermiT		272 / 298	26	2,012.9 s
7	HermiT-OA4		272 / 298	26	2,068.6 s
8	Pellet-OA4		261 / 298	37	2,169.5 s
9	FaCT++		244 / 298	54	2,671.9 s
10	Racer		237 / 298	61	1,322.2 s
11	Chainsaw		191 / 298	107	1,587.4 s
12	JFact		189 / 298	109	2,404.3 s
13	jcel		133 / 298	165	98.4 s

Discipline: OWL DL Classification (finished)

Rank	Reasoner	Progress	Score	!	Time
1	Konclude		288 / 306	18	1,308.9 s
2	MORe		247 / 306	59	2,143.0 s
3	HermiT-OA4		237 / 306	69	5,808.2 s
4	HermiT		236 / 306	70	5,416.4 s
5	TrOWL		201 / 306	105	971.1 s
6	FaCT++		200 / 306	106	1,361.3 s
7	Pellet-OA4		187 / 306	119	2,179.3 s
8	Racer		164 / 306	142	1,103.8 s
9	JFact		128 / 306	178	889.5 s
10	Chainsaw		119 / 306	187	1,709.0 s

Discipline: OWL EL Consistency (finished)

Rank	Reasoner	Progress	Score	!	Time
1	ELK		298 / 298	0	425.1 s
2	Konclude		298 / 298	0	1,050.4 s
3	ELepHant		296 / 298	2	312.7 s
4	MORe		295 / 298	3	1,290.5 s
5	Pellet-OA4		285 / 298	13	2,018.6 s
6	HermiT		282 / 298	16	846.6 s
7	HermiT-OA4		282 / 298	16	874.7 s
8	Chainsaw		276 / 298	22	473.6 s
9	TrOWL		273 / 298	25	699.3 s
10	FaCT++		270 / 298	28	636.3 s
11	jcel		261 / 298	37	1,465.4 s
12	Racer		256 / 298	42	1,211.4 s
13	JFact		227 / 298	71	2,307.6 s

Discipline: OWL DL Consistency (finished)

Rank	Reasoner	Progress	Score	!	Time
1	Konclude		303 / 306	3	1,341.9 s
2	HermiT		294 / 306	12	1,449.6 s
3	HermiT-OA4		293 / 306	13	1,549.4 s
4	Chainsaw		291 / 306	15	1,100.5 s
5	Pellet-OA4		278 / 306	28	1,194.1 s
6	FaCT++		276 / 306	30	1,341.2 s
7	TrOWL		266 / 306	40	1,089.3 s
8	MORe		253 / 306	53	1,823.0 s
9	Racer		239 / 306	67	2,604.3 s
10	JFact		166 / 306	140	1,469.9 s

Fig. 3. Results of the competition by track as displayed from the live competition display. Score indicates the number of problems solved out of total problems for that track. The number of unsolved problems (whether by timeout, crash, or "wrong" results) are displayed in the next column. Time indicates the time actually taken to complete *solved* problems. Time is used to resolve ties in problems solved.

Out of the 6 competitions, 4 were won by the new hybrid reasoner Konclude [20], and two (EL-consistency and EL-classification) were won by ELK [5]. Figures 4 and 5 show how well the winning reasoners did in terms of reasoning time. There are a couple of observations to be made here. First, Konclude, the winner of all three DL disciplines, is doing consistently better on the majority of the easier ontologies, but towards the harder end on the right, other reasoners catch up.

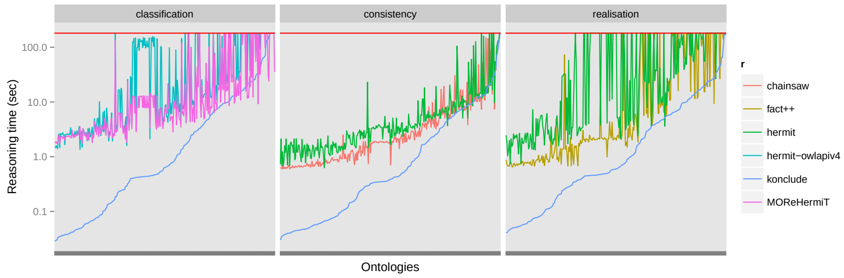

Fig. 4. Reasoning time of the three winning reasoners in each category: DL profile. Red line: Timeout. Ordered by speed of winning reasoner.

This is particularly obvious for the EL-classification competition. Up until a certain point, Konclude is doing much (sometime up to an order of magntitude) better than ELK (the winner of the discipline), but towards the harder end, ELK overtakes Konclude. Some of this may be due to JVM overhead for ELK and our "fire and forget" execution strategy. If we had a long running server based approach it might be that the JVM overhead for easy cases would be effectively amortised. Another interesting observation is the performance of ELepHants [18] consistency check, which regularly outperforms both ELK and Konclude. We speculate that this is due to differences in whether parsing time is incorporated in the reported time (e.g., ELK does this for all tasks and Konclude does this for consistency checking).

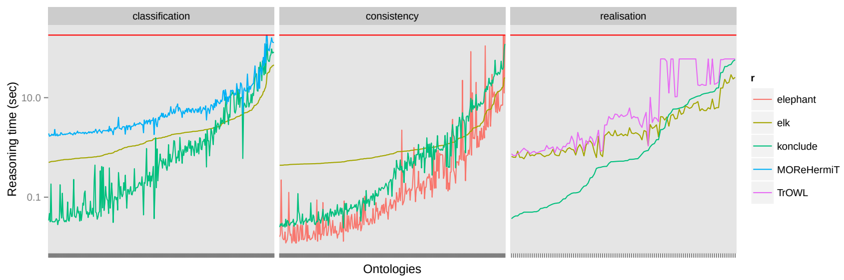

Fig. 5. Reasoning time of the three winning reasoners in each category: EL profile. Red line: Timeout. Ordered by speed of winning reasoner.

A full break-down for all reasoners by competition can be seen in Table 3.

The competition is reasonably challenging: In only two tracks (EL consistency and EL classification) did any reasoner solve all the problems in competition conditions. Figure 6 shows a detailed breakdown of how many problems were solved by how many reasoners (in percent).

Reasoner	Task	Success		Timeout		Error	
		DL	EL	DL	EL	DL	EL
Chainsaw	CL	122	191	168	94	16	13
Chainsaw	CON	292	276	3	19	11	3
Chainsaw	REAL	82	44	166	63	16	2
ELepHant	CL	NA	293	NA	5	NA	0
ELepHant	CON	NA	296	NA	2	NA	0
ELepHant	REAL	NA	109	NA	0	NA	0
ELK	CL	NA	298	NA	0	NA	0
ELK	CON	NA	298	NA	0	NA	0
ELK	REAL	NA	109	NA	0	NA	0
FaCT++	CL	202	244	87	51	17	3
FaCT++	CON	279	270	14	22	13	6
FaCT++	REAL	183	79	56	27	25	3
HermiT	CL	241	273	63	25	2	0
HermiT	CON	296	282	7	16	3	0
HermiT	REAL	167	57	92	52	5	0
HermiT-4	CL	241	273	63	25	2	0
HermiT-4	CON	296	282	6	16	4	0
HermiT-4	REAL	165	57	93	52	6	0
jcel	CL	NA	134	NA	158	NA	6
jcel	CON	NA	262	NA	34	NA	2
Jfact	CL	143	208	104	88	59	2
Jfact	CON	174	229	80	69	52	0
Jfact	REAL	128	66	89	43	47	0
Konclude	CL	298	298	7	0	1	0
Konclude	REAL	261	109	2	0	1	0
Konclude	CON	305	298	1	0	0	0
MORe	CL	266	296	38	2	2	0
MORe	CON	264	295	40	3	2	0
Pagoda	REAL	120	96	49	13	95	0
Pellet-4	CL	188	261	104	28	14	9
Pellet-4	REAL	187	75	53	32	24	2
Pellet-4	CON	280	286	26	12	0	0
Racer	CL	218	260	86	38	2	0
Racer	CON	257	258	48	40	1	0
Racer	REAL	186	78	75	31	3	0
TrOWL	CL	271	275	0	0	35	23
TrOWL	CON	270	273	0	0	36	25
TrOWL	REAL	221	87	0	0	43	22

Table 3. Full break-down of solved problems by reasoner and task. Note that "HermiT-4" refers to the current version of HermiT wrapped in the OWL API version 4.

It is interesting to observe that the union of all reasoners successfully process all EL reasoning problems. As one might expect, realisation is still challenging for reasoners. But in all tracks, for the majority of reasoners, the ORE problems provide a good target for optimisation. We know, from the results of the competition, that these problems are (almost) all in principle solvable on a modest machine in around 3 minutes.

5 Discussion

The top slots in all tracks have been dominated by Konclude (and to a lesser extend by ELK) for two years now. Konclude is an highly optimised, very efficient reasoner whose developers continuously test it against a vast set of available ontologies. Even so, there is interesting jockeying around second and third place

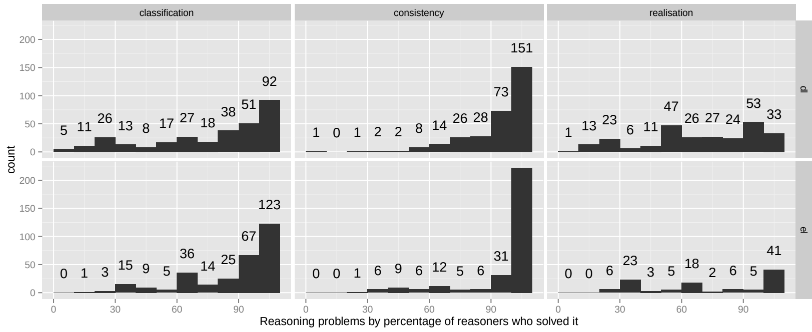

Fig. 6. Number of reasoning problems by percentage of reasoners solving them. For example, 5 DL-classification tasks where not solved by any reasoner, and 123 EL-classification tasks were solved by all reasoners.

for all tracks, and we were impressed with how well older reasoners, which have not been updated recently (notably Pellet and Racer), fared.

The robustness experiments in [2] used a much longer timeout (up to 2 hours per test), though the analysis clustered results by subdivisions of the timeout period. That suggests that a slightly longer timeout might significantly increase the total number of solved problems across reasoners. This needs to be balanced by the increased running time of the competition (which is bounded by the slowest reasoner). We prefer the bulk of the competition to be executed during a single day of the DL workshop to facilitate engagement which imposes fairly tight limits on the timeout and number of problems. (This year, due to technical issues, we were not able to do that.) Having a separate offline competition remains an option, but it is unclear that this extra significant effort produces much benefit.

However, the ORE workshop solicits “challenge” ontologies from ontology developers partly in the hopes of directing reasoner developer attention to real user performance needs. Unfortunately, we have not yet managed to do a “user ontology” track, though we are hoping to do so as a satellite event at OWLED 2015. This will almost certainly have to be offline and, of course, many of the submitted ontologies are currently unsolved by current reasoners.

The most important next expansion of tracks is to conjunctive query answering (CQA). Setting up a meaningful CQA competition is significantly more difficult, because we do not only have to consider ontologies, but also queries and data. Gathering suitable (meaningful) queries is probably the most difficult hurdle to overcome. However, we made significant progress toward a reasonable design this year and hope to incorporate it in next year’s competition.

Another area of interest is application style benchmarks which would situate the reasoning task in the context of a pattern of use characteristic of a real or realistic application. This might include modification of the ontology or data during the competition run.

6 Conclusion

The ORE 2015 Reasoner Competition continues the success of its predecessors. Participants, workshop attendees, and interested bystanders all had fun, and 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 years 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.

Acknowledgments. The ORE competition has been the work of many people over the years and we would like to especially acknowledge the contributions of Ernesto Jiménez-Ruiz for running the very first primitive competition and being a PC chair for all workshops; Samantha Bail for being a PC chair as well as implementing the first "live results" screen; Ian Horrocks for helping getting the project started; and Yevgeny Kasakov for helpful discussions on the competition design as well as finding a critical bug *just before the competition started*. We also would like to acknowledge the generous support of B2i Healthcare[9] for their repeated donations of prize money and the DBOnto project[10] for funding compeition T-Shirts.

Finally, the competition would not have been possible without the donation of cluster time by Konstantin Korovin (funded by Royal Society research grant RG080491).

References

1. Birte Glimm, Ian Horrocks, Boris Motik, Giorgos Stoilos, and Zhe Wang. HermiT: An OWL 2 Reasoner. *J. Autom. Reasoning*, 53(3):245–269, 2014.
2. Rafael S. Gonçalves, Nicolas Matentzoglu, Bijan Parsia, and Uli Sattler. The Empirical Robustness of Description Logic Classification. In *Proceedings of the ISWC 2013 Posters & Demonstrations Track, Sydney, Australia, October 23, 2013*, pages 277–280, 2013.
3. Volker Haarslev, Kay Hidde, Ralf Möller, and Michael Wessel. The RacerPro knowledge representation and reasoning system. *Semantic Web*, 3(3):267–277, 2012.
4. Matthew Horridge and Sean Bechhofer. The OWL API: A Java API for OWL ontologies. *Semantic Web*, 2(1):11–21, 2011.
5. Yevgeny Kazakov, Markus Krötzsch, and Frantisek Simancik. The Incredible ELK - From Polynomial Procedures to Efficient Reasoning with EL Ontologies. *J. Autom. Reasoning*, 53(1):1–61, 2014.
6. Michael Lee, Nicolas Matentzoglu, Bijan Parsia, and Uli Sattler. A multi-reasoner, justification-based approach to reasoner correctness. In *International Semantic Web Conference*, 2015.

[9] `http://b2i.sg`

[10] `http://www.cs.ox.ac.uk/projects/DBOnto/`

7. Nicolas Matentzoglu, Samantha Bail, and Bijan Parsia. A Snapshot of the OWL Web. In *The Semantic Web - ISWC 2013 - 12th International Semantic Web Conference, Sydney, NSW, Australia, October 21-25, 2013, Proceedings, Part I*, pages 331–346, 2013.
8. Nicolas Matentzoglu, Jared Leo, Valentino Hudhra, Uli Sattler, and Bijan Parsia. A survey of current, stand-alone owl reasoners. In Michel Dumontier, Birte Glimm, Rafael Gonçalves, Matthew Horridge, Ernesto Jiménez-Ruiz, Nicolas Matentzoglu, Bijan Parsia, Giorgos Stamou, and Giorgos Stoilos, editors, *Informal Proceedings of the 4th International Workshop on OWL Reasoner Evaluation*, volume 1387. CEUR-WS, 2015.
9. Nicolas Matentzoglu and Bijan Parsia. ORE 2015 reasoner competition dataset http://dx.doi.org/10.5281/zenodo.18578, June 2015.
10. Julian Mendez. jcel: A Modular Rule-based Reasoner. In *Proceedings of the 1st International Workshop on OWL Reasoner Evaluation (ORE-2012), Manchester, UK, July 1st, 2012*, 2012.
11. Boris Motik, Peter F Patel-Schneider, Bijan Parsia, Conrad Bock, Achille Fokoue, Peter Haase, Rinke Hoekstra, Ian Horrocks, Alan Ruttenberg, Uli Sattler, and Mike Smith. Owl 2 web ontology language: Structural specification and functional-style syntax. W3C, 2009.
12. Natalya Fridman Noy, Nigam H. Shah, Patricia L. Whetzel, Benjamin Dai, Michael Dorf, Nicholas Griffith, Clement Jonquet, Daniel L. Rubin, Margaret-Anne D. Storey, Christopher G. Chute, and Mark A. Musen. BioPortal: ontologies and integrated data resources at the click of a mouse. *Nucleic Acids Research*, 37(Web-Server-Issue):170–173, 2009.
13. Ignazio Palmisano. JFact repository, 2015.
14. F.J. Pelletier, G. Sutcliffe, and C.B. Suttner. The Development of CASC. *AI Communications*, 15(2-3):79–90, 2002.
15. Yuan Ren, Jeff Z. Pan, and Yuting Zhao. Soundness Preserving Approximation for TBox Reasoning. In *Proceedings of the Twenty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2010, Atlanta, Georgia, USA, July 11-15, 2010*, 2010.
16. Ana Armas Romero, Bernardo Cuenca Grau, and Ian Horrocks. MORe: Modular Combination of OWL Reasoners for Ontology Classification. In *The Semantic Web - ISWC 2012 - 11th International Semantic Web Conference, Boston, MA, USA, November 11-15, 2012, Proceedings, Part I*, pages 1–16, 2012.
17. Uli Sattler and Nicolas Matentzoglu. List of Reasoners (owl.cs) http://owl.cs.manchester.ac.uk/tools/list-of-reasoners/. Modified: 01/09/2014.
18. Baris Sertkaya. The ELepHant Reasoner System Description. In *Informal Proceedings of the 2nd International Workshop on OWL Reasoner Evaluation (ORE-2013), Ulm, Germany, July 22, 2013*, pages 87–93, 2013.
19. Evren Sirin, Bijan Parsia, Bernardo Cuenca Grau, Aditya Kalyanpur, and Yarden Katz. Pellet: A practical OWL-DL reasoner. *J. Web Sem.*, 5(2):51–53, 2007.
20. Andreas Steigmiller, Thorsten Liebig, and Birte Glimm. Konclude: System description. *J. Web Sem.*, 27:78–85, 2014.
21. G. Sutcliffe. The TPTP Problem Library and Associated Infrastructure: The FOF and CNF Parts, v3.5.0. *Journal of Automated Reasoning*, 43(4):337–362, 2009.
22. G. Sutcliffe and C. Suttner. The State of CASC. *AI Communications*, 19(1):35–48, 2006.
23. Edward Thomas, Jeff Z. Pan, and Yuan Ren. TrOWL: Tractable OWL 2 Reasoning Infrastructure. In *The Semantic Web: Research and Applications, 7th Extended Semantic Web Conference, ESWC 2010, Heraklion, Crete, Greece, May 30 - June 3, 2010, Proceedings, Part II*, pages 431–435, 2010.

24. Dmitry Tsarkov and Ian Horrocks. FaCT++ Description Logic Reasoner: System Description. In *Automated Reasoning, Third International Joint Conference, IJCAR 2006, Seattle, WA, USA, August 17-20, 2006, Proceedings*, pages 292–297, 2006.
25. Dmitry Tsarkov and Ignazio Palmisano. Chainsaw: a Metareasoner for Large Ontologies. In *Proceedings of the 1st International Workshop on OWL Reasoner Evaluation (ORE-2012), Manchester, UK, July 1st, 2012*, 2012.
26. Yujiao Zhou, Yavor Nenov, Bernardo Cuenca Grau, and Ian Horrocks. Pay-as-you-go OWL query answering using a triple store. In *Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence, July 27 -31, 2014, Québec City, Québec, Canada.*, pages 1142–1148, 2014.