OWL Reasoner Evaluation (ORE) Workshop 2013 Results: Short Report

Rafael S. Gonçalves[1], Samantha Bail[1], Ernesto Jimenez-Ruiz[2], Nicolas Matentzoglu[1], Bijan Parsia[1], Birte Glimm[3], and Yevgeny Kazakov[3]

[1] School of Computer Science, The University of Manchester, UK
[2] Department of Computer Science, University of Oxford, UK
[3] Institut für Künstliche Intelligenz, Ulm University, Germany

Abstract. The OWL reasoner evaluation (ORE) workshop brings together reasoner developers and ontology engineers in order to discuss and evaluate the performance and robustness of modern reasoners on OWL ontologies. In addition to paper submissions, the workshop featured a live and offline reasoner competition where standard reasoning tasks were tested: classification, consistency, and concept satisfiability. The reasoner competition is performed on several large corpora of real-life OWL ontologies obtained from the web, as well as user-submitted ontologies which were found to be challenging for reasoners. Overall there were 14 reasoner submissions for the competition, some of which dedicated to certain subsets or profiles of OWL 2, and implementing different algorithms and optimisations. In this report, we give an overview of the competition methodology and present a summary of its results, divided into the respective categories based on OWL 2 profiles and test corpora.

1 Introduction

The OWL Reasoner Evaluation Workshop (ORE) aims at being an international venue for the annual systematic evaluation of reasoners for (subsets of) the Web Ontology Language OWL [9,3] and bringing together both users and developers of such reasoners. The first ORE workshop was organized in 2012 as a satellite event[4] of the IJCAR conference [10], and started as an initiative in the context of the SEALS (Semantic Evaluation At Large Scale) project [29]. In 2013 the ORE workshop was organized together with the Description Logic (DL) workshop.

This report summarizes the results of the ORE 2013 reasoner competition. All test data, results, and further information about the competition are available online: `http://ore2013.cs.manchester.ac.uk`.

The remainder of the report is organized as follows. In Section 2, we present the methodology of the competition. Section 3 provides a brief description of each participating OWL reasoner. The results of the offline and live competitions are shown in Sections 4 and 5, respectively. In Section 6 we present the results for the user-submitted ontologies. Finally, Section 7 provides a summary of the competition results.

[4] `http://www.cs.ox.ac.uk/isg/conferences/ORE2012/`

2 Methodology

We start by describing the reasoning tasks considered in the competition, followed by the presentation of the benchmark framework created for ORE 2013. Subsequently, we report on the hardware and ontology corpora used.

2.1 Reasoning tasks

The competition was based on three standard reasoning tasks, namely classification, consistency checking, and concept satisfiability checking. The call for submissions also included query answering, but there were no reasoners submitted for this task.

2.1.1 Ontology classification The classification task was chosen as the most complex of the three tasks. Given an ontology, the reasoners were asked to return an ontology file (parseable by the OWL API) in OWL functional syntax, containing a set of `SubClassOf` axioms of the form $\alpha := A \sqsubseteq B$, for named concepts A, $B \in sig(\mathcal{O})$, where $\mathcal{O} \models \alpha$ according to the following specifications:

1. Non-tautology:
   - $A \in sig(\mathcal{O}) \cup \top$
   - $B \in sig(\mathcal{O}) \cup \bot$
   - $A \neq B$
2. Directness: there exists no named concept $C \in sig(\mathcal{O})$ s.t. $\mathcal{O} \models A \sqsubseteq C$ and $C \sqsubseteq B$, where C is not equivalent to A, B, or $\bot$.
3. Conciseness: if $\mathcal{O} \models A \equiv \bot$, the only axiom with A on the left-hand side is $A \sqsubseteq \bot$.
4. Consistency: if the given ontology is inconsistent, the only output is the axiom $\top \sqsubseteq \bot$.
5. Non-strictness: if $\mathcal{O} \models A \equiv B$, output $A \sqsubseteq B$ and $B \sqsubseteq A$.

These specifications were selected in order to obtain a set of `SubClassOf` axioms that would represent all subsumptions between named classes, while omitting irrelevant information.

2.1.2 Ontology consistency For this task, the reasoner was asked to test the consistency of the ontology (i.e. whether $\mathcal{O} \models \top \sqsubseteq \bot$), and return ‘true’ or ‘false’, respectively.

2.1.3 Concept satisfiability This task was performed by randomly selecting ten concepts from each ontology in the respective corpus, giving precedence to unsatisfiable concepts where possible. The reasoner was then asked to test the satisfiability of the concept, i.e. whether $\mathcal{O} \models A \equiv \bot$ for a named concept A, and return ‘true’ or ‘false’, respectively.

2.2 Benchmark framework

2.2.1 Implementation The aim of the benchmarking framework is to work with as many different reasoner configurations as possible, without the need to interfere with reasoner internals. We therefore asked the system developers to write a simple executable wrapper for their reasoner which would accept input arguments (ontology file name, reasoning task, output directory, concept name) and output results according to our specification (a valid OWL file with the class hierarchy, 'true'/'false' for the consistency and satisfiability tasks, as well as the time taken for the task, or a separate file with an error trace).

The time measured is the wall-clock time (in milliseconds) elapsed from the moment preceding reasoner creation (e.g. before the call to `ReasonerFactory.createReasoner(ontology)` in the OWL API [8] where the ontology has already been parsed into an OWL object) to the completion of the given task, i.e. it includes the loading and possibly pre-processing time required by the *reasoner*, but excludes time taken for file I/O. While measuring CPU time would be more accurate, it comes with added complexity for concurrent implementations – for instance, in Java, one would have to aggregate the run times of each thread. The reasoners are also asked to enforce a five minute timeout, that is, if the measured time exceeds 5 minutes then the reasoner should stop the ongoing operation, and terminate itself. Failure to do so will trigger a kill command sent to the running process after another minute in order to give enough time for the process to terminate; i.e. the hard timeout is six minutes.

While one might argue that leaving the reporting of operation times to the reasoners may be error-prone, we believe that letting reasoner developers themselves handle the input and output of their system, as well as the time measurement, is the most straightforward way to include as many systems as possible; regardless of their implementation programming language, whether they use the OWL API, employ concurrent implementations, and so on. The large number of reasoners that was submitted to the competition shows that writing this simple wrapper script lowered the barrier for participation, and despite some difficulties with non-standard output, most reasoners adhered to the specifications closely enough for us to analyse their outputs.

Additionally, it is clear that reasoners which do *not* implement the five minute timeout, but rather rely on the kill signal after the six minute timeout sent by the benchmark framework, *could* potentially gain a slight advantage through this additional minute. However, not only is the number of successfully completed tasks between the five and six minute marks negligible, but also we have automatically induced a timeout for those reasoners that exceeded a runtime of five minutes for some input.

2.2.2 Correctness check The reasoner output was checked for correctness by a majority vote, i.e. the result returned by the most reasoners was considered to be correct.[5] Since the ontologies in the test corpus were 'real-life' ontologies,

[5] Unless most reasoners return an empty OWL file, in which case the majority vote is taken based on those reasoners which output a non-empty file.

this was the most straightforward way to automatically determine correctness without manual inspection or artificially generating test cases.

In the case of the consistency and satisfiability challenges the output was a simple unambiguous 'true' or 'false', so any incorrect results were unlikely to be caused by erroneous output from a sound reasoner; however, for ontology classification, the reasoners output an ontology file containing OWL `SubClassOf` axioms, which may lead to errors if the systems did not exactly follow the above specifications on which types of axioms to include or exclude. For the purpose of verifying correctness of the output we rely on an ontology *diff* to determine whether two given results are logically equivalent [6]. The diff is tuned to ignore (1) tautological axioms of the type $A \sqsubseteq \top$ for any named concept A, (2) axioms of the form $\bot \sqsubseteq B$ or $A \sqsubseteq B$, where A, B are named concepts and A is unsatisfiable, and (3) if two result files are not equivalent due to OWL `EquivalentClassOf` axioms, these axioms are ignored.[6]

2.2.3 Success and failure In the end, the outcome of a reasoning task on an ontology was either 'success' or 'fail'. A reasoner would pass the test ('solve the problem') successfully if it met the following three criteria:

- Process the ontology without throwing an error (e.g. parsing error, out of memory, unsupported OWL feature, etc.).
- Return a result within the allocated timeout.
- Return the *correct* result (based on the majority vote).

Likewise, a reasoner would *fail* a task if it did one of the following:

- Throw an error and abort the reasoning task.
- Return no result within the allocated time.
- Return an incorrect result (based on the majority vote).

Note that these criteria mean that a reasoner could successfully solve a task while being unsound or incomplete, or without completing the reasoning task within the allocated time. For example, for the classification task, if the reasoner has already found all required entailed atomic subsumptions without performing all possible entailment checks within the five minute time frame, it can simply output this 'intermediate' result before terminating the process. Since the correctness check is performed on whatever the reasoner returns within the timeout, the resulting output would be considered to be correct, despite the fact that the reasoner has not fully completed the task.

Likewise, a reasoner which does not support certain OWL 2 features, such as datatypes, might find (if there are *any to find*) the required atomic subsumptions via some other 'route' if there are several reasons why the entailment holds. In other words, if there exist multiple *justifications* (minimal entailing subsets) for a subsumption of which at least one only contains *supported* features, then the reasoner will still be able to find the subsumption without having to process the

[6] While the presence of equivalences in some result should not a problem when compared to a result with these equivalences in subsumption form, reasoners tend not to produce the latter because they are non-strict subsumptions, so we allowed equivalences and tuned our diff to ignore them where applicable.

unsupported feature. This is an issue we are planning to address with the next iteration of the benchmark framework by modifying ontologies (i.e. 'breaking' their justifications) in order to specifically test certain OWL 2 features.

2.3 Hardware

The experiments were run on a cluster of identical computers (one reasoner per computer) that were made available to us by Konstantin Korovin of the iProver project[7] at The University of Manchester, supported by the Royal Society grant RG080491. Each computer had the following configuration:

- Intel Xeon QuadCore CPU @2.33GHz
- 12GB RAM (8GB assigned to the process)
- Running the Fedora 12 operating system
- Java version 1.6

2.4 Test corpora

2.4.1 Main test corpus For each of the OWL 2 profiles [16] used in the competition (OWL 2 EL, RL, and DL ontologies which were not in any of the sub-profiles) we gathered a test set of up to 200 ontologies. The pool of ontologies we sampled from was composed of three corpora: *(i)* the NCBO BioPortal[8] corpus [17], *(ii)* the Oxford Ontology Library[9], and *(iii)* the corpus of ontologies from the Manchester Ontology Repository[10] [13]. The corpora were filtered to only include OWL 2 ontologies which had at least 100 axioms and 10 named concepts. Note that the sample ontology pool, composed of 2499 ontologies, does contain some degree of duplication due to the intersection of BioPortal, the Manchester OWL Repository, and the Oxford Ontology Library.

The ontologies were then binned by profile, i.e. one bin for each of the following: OWL 2 EL ontologies, OWL 2 RL, and OWL 2 DL. Regarding the latter, we chose to include here those ontologies that do not fall into any of the sub-profiles (i.e. OWL 2 EL, RL, or QL) in order to ensure that features outside the sub-profiles were tested. For each of these profile bins, a stratified random sample was drawn to obtain a set of 200 ontologies:

- 50 small ontologies (between 100 and 499 logical axioms)
- 100 medium sized ontologies (between 500 and 4,999 logical axioms)
- 50 large ontologies (5,000 and more logical axioms)

Note that these thresholds and weightings were chosen based on the distribution of ontology sizes we have found in several ontology corpora which follow (roughly) a normal distribution, with a large number of medium-sized ontologies and fewer small and large ontologies. While it would have been possible to select

[7] `http://www.cs.man.ac.uk/~korovink/iprover/`
[8] `http://bioportal.bioontology.org/`
[9] `http://www.cs.ox.ac.uk/isg/ontologies/`
[10] `http://owl.cs.manchester.ac.uk/owlcorpus`

exclusively medium-sized and large ontologies, we also expected some small ontologies to be fairly complex for the reasoners, which is why they were included in the test corpus.

In addition to the ontologies from BioPortal, the Oxford Library, the Manchester Repository, and user-submitted ontologies, the May 2013 version of the National Cancer Institute (NCI) Thesaurus (NCIt) [5], and the January 2011 version of the Systematized Nomenclature of Medicine (SNOMED) Clinical Terms (SNOMED CT) [21] were also added to the corpus, respectively to the DL and EL profile bins.

The experiments were run on the OWL functional syntax serialisations of the selected ontologies, except for one reasoner (Konclude) which currently only supports OWL/XML syntax. A number of ontologies serialised into functional syntax (55 across all the sets) turned out to be broken (they were correctly loaded and serialised, but the serialisation could not be parsed back by the OWL API), possibly due to problems with the respective serialiser in the OWL API (version 3.4.4). These were replaced by random selections for their respective bin. The same occurred for 12 ontologies serialised into OWL/XML.

The entire sampling process was performed twice in order to create two complete test sets: Set A for the offline competition, and Set B for the live competition. Note that some ontologies occurred in both Set A and B: 40 ontologies occurred in both Set A and B for the DL category, Set A and B were fully identical for the EL category, and 29 ontologies were shared between Set A and B in the RL category.

2.4.2 User-submitted ontologies In the call for submissions to the ORE 2013 workshop, we also included a call for 'hard' ontologies and potential reasoner benchmark suites. Several groups of ontology and reasoner developers submitted their ontologies, which were either newly developed OWL ontologies or modifications of existing ones. These included:

- C. M. Keet, A. Lawrynowicz, C. d'Amato, M. Hilario: the Data Mining OPtimization Ontology (DMOP) [12], a complex ontology with around 3,000 logical axioms in the $\mathcal{SROIQ}(D)$ description logic which makes uses of all OWL 2 DL features.
- M. Samwald: Genomic CDS [20], an $\mathcal{ALCQ}$ ontology containing around 4,000 logical axioms, which involves a high number of qualified number restrictions of the type 'exactly 2'.
- V. Chaudhri, M. Wessel: Bio KB 101 [2], a set of OWL approximations of the first-order logic representation of a biology textbook, which consists of 432 different approximations containing various OWL 2 features. Only 72 of these files were in the OWL 2 DL profile and thus used for the reasoner evaluation.
- W. Song, B. Spencer, W. Du: three ontology variants:
  - FMA-FNL, a variant of the FMA (Foundational Model of Anatomy) ontology [19], a large and highly cyclic $\mathcal{ALCOI}(D)$ ontology with over 120,000 locial axioms.

  - GALEN-FNL, a highly cyclic $\mathcal{ALCHOI}(D)$ variant of the well-known Galen ontology [18], which contains around 37,000 logical axioms and 951 object properties.
  - GALEN-Heart: a highly cyclic $\mathcal{ALCHOI}(D)$ ontology containing a module extracted from the Galen ontology with over 10,000 logical axioms.
- S. Croset: Functional Therapeutic Chemical Classification System (FTC)[11], a large ontology with nearly 300,000 logical axioms in the OWL 2 EL profile.

As mentioned above, some of the user-submitted ontologies (all except Bio KB and DMOP) were added to the set used in the competition. Additionally, we also performed a separate benchmark on *all* of the user-submitted ontologies.

3 Participating reasoners

3.1 OWL 2 DL reasoners

Chainsaw [28] is a 'metareasoner' which first computes modules for an ontology, then delegates the processing of those modules to an existing OWL 2 DL reasoner, e.g. FaCT++ in the current implementation.

FaCT++ [27] is a tableaux reasoner written in C++ which supports the full OWL 2 DL profile.

HermiT [4] is a Java-based OWL 2 DL reasoner implementing a hypertableau calculus.

JFact is a Java implementation of the FaCT++ reasoner with extended datatype support.[12]

Konclude is a C++ reasoner supporting the full OWL 2 DL profile except datatypes. It uses an optimised tableau algorithm which also supports parallelised processing of non-deterministic branches and the parallelisation of higher-level reasoning tasks, e.g. satisfiability and subsumption tests.[13]

MORe [1] is Java-based *modular* reasoner which integrates a fully-fledged (and slower) reasoner with a profile specific (and more efficient) reasoner. In the competition, MORe has integrated both HermiT and Pellet [24] as OWL 2 DL reasoners and ELK as the OWL 2 EL profile specific reasoner.

Treasoner [7] is a Java reasoner which implements a standard tableau algorithm for $\mathcal{SHIQ}$.

TrOWL [26] is an approximative OWL 2 DL reasoner. In particular, TrOWL utilises a semantic approximation to transform OWL 2 DL ontologies into OWL 2 QL for conjunctive query answering and a syntactic approximation from OWL 2 DL to OWL 2 EL for TBox and ABox reasoning.

WSClassifier [25] is a Java reasoner for the $\mathcal{ALCHOI}(D)$ fragment of OWL 2 DL, using a hybrid of the consequence based reasoner ConDOR [23] and hypertableau reasoner HermiT.

[11] `https://www.ebi.ac.uk/chembl/ftc/`

[12] `http://sourceforge.net/projects/jfact/`

[13] `http://www.derivo.de/en/produkte/konclude/`

3.2 OWL 2 EL reasoners

ELepHant [22] is a highly optimised consequence-based $\mathcal{EL}+$ reasoner written in C, which is aimed at platforms with limited memory and computing capabilities (e.g. embedded systems).

ELK [11] is a consequence-based Java reasoner which utilises multiple cores/processors by parallelising multiple threads.

jcel [14] uses a completion-based algorithm, which is a generalization of CEL's algorithm. It is a Java reasoner which supports ontologies in $\mathcal{EL}+$.

SnoRocket [15] is a Java reasoner developed for the efficient classification of the SNOMED CT ontology. It implements a multi-threaded saturation algorithm similar to that of ELK, thus support concurrent classification.

3.3 OWL 2 RL reasoners

BaseVISor is a Java-based forward-chaining inference engine which supports OWL 2 RL and XML Schema Datatypes.[14]

4 Results – Offline competition

4.1 OWL 2 DL results

Nine reasoners entered the OWL 2 DL category, although not all of them competed in the three reasoning tasks. MORe participated with both HermiT and Pellet as the internal DL reasoner. The results for the classification, consistency, and satisfiability tasks are shown in Figure 1.

In the classification task, HermiT performed best in terms of robustness with 147 out of 204 ontologies that were correctly processed within the timeout (at 12.3s per ontology), whereas MORe-Pellet achieved the smallest mean time (2.8s per ontology) for the 141 ontologies it processed correctly.

In the consistency task, Konclude processed the highest number of ontologies correctly (186 out of 204), while also performing fastest on average with 1.7s per ontology; Konclude was also twice as fast as the second faster reasoner (HermiT).

Finally, for the DL satisfiability task, Konclude also processed the highest number of concepts correctly (1,929 out of 2,040) within the given timeout, while coming second after Chainsaw (1.3s) in terms of speed, with a mean time of 1.8s per ontology.

4.2 OWL 2 EL results

In addition to the EL-specific reasoners, all OWL 2 DL reasoners also participated in the EL category; the results for all participating reasoners on the three reasoning tasks in the EL profile are shown in Figure 2. In both the classification and consistency categories, ELK performed extremely well both in terms

[14] `http://vistology.com/basevisor/basevisor.html`

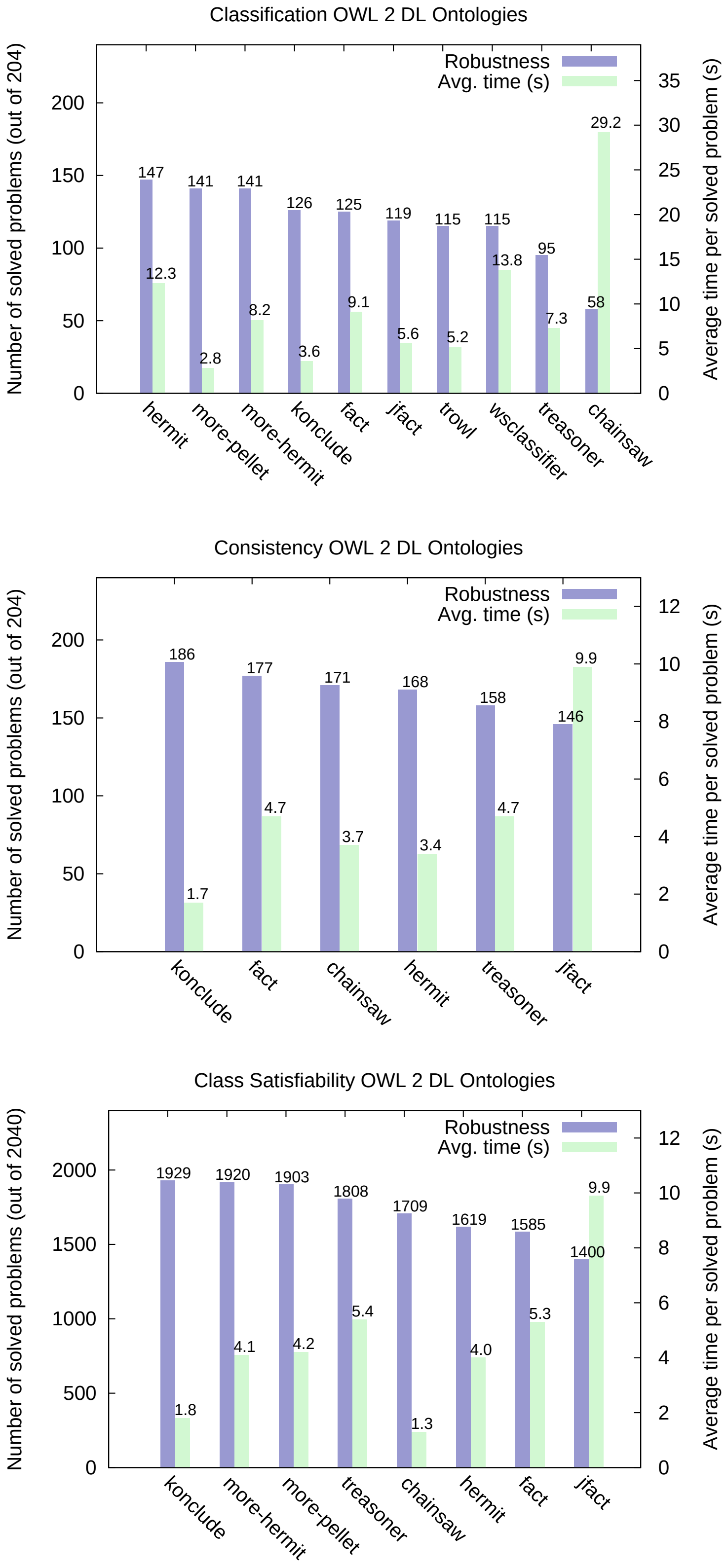

Fig. 1: Results (robustness/average time) for the OWL 2 **DL** category.

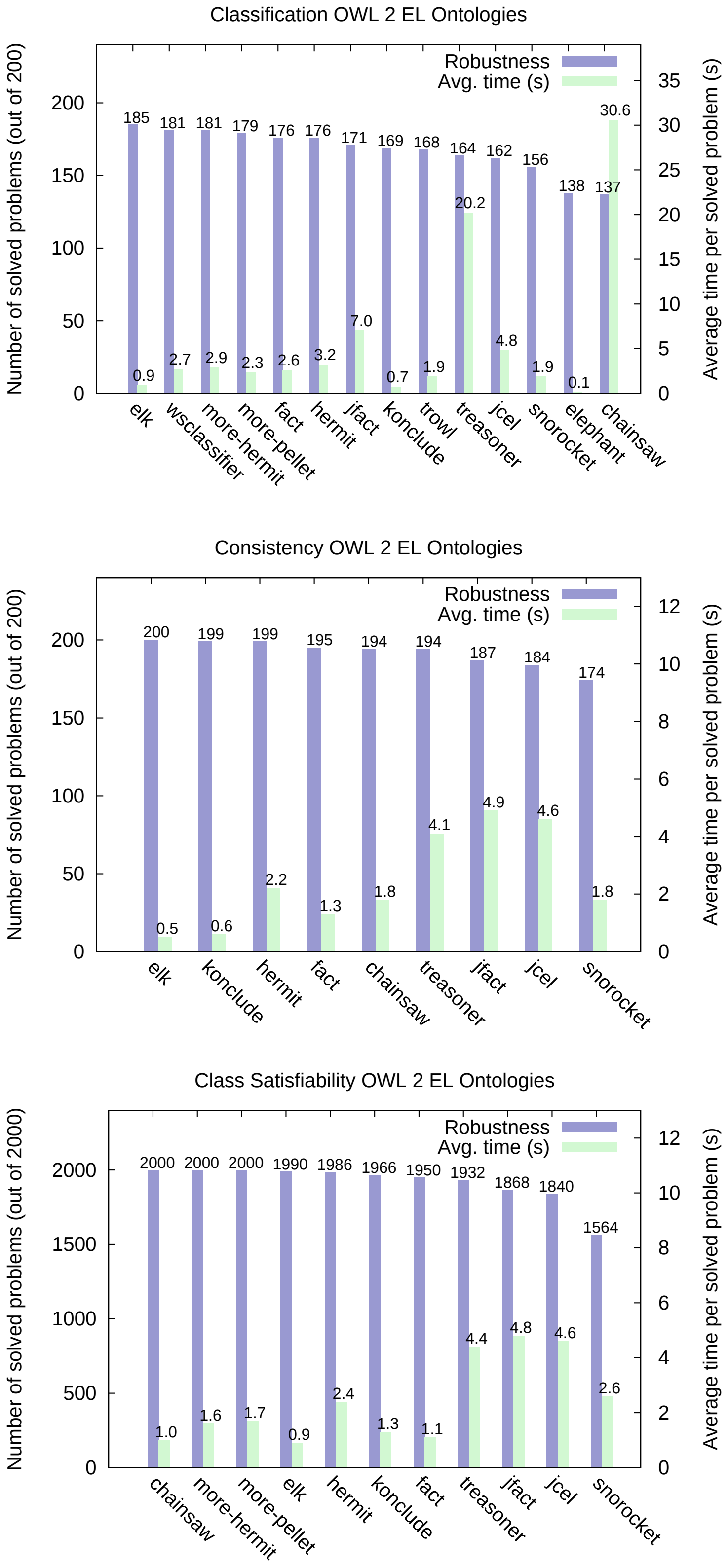

Fig. 2: Results (robustness/average time) for the OWL 2 **EL** category.

of robustness (185 and 200 out of 200 correctly processed ontologies) as well as average speed (0.9s for classification, 0.5s for consistency checking). Perhaps surprisingly, MORe with both HermiT and Pellet performed worse than ELK on robustness, as we expected its combination of ELK with a DL reasoner to handle *more* ontologies than the stand-alone version of ELK which does not support the full OWL 2 EL profile. However, it is possible that the DL reasoners in MORe got in fact 'held up' by those parts of the ontologies that the stand-alone ELK simply ignored, which may have caused a this slightly worse result.

Two of the EL-specific reasoners SnoRocket and ELepHant both performed comparatively fast on those ontologies they did successfully process, but failed to process a large number of ontologies (44 and 62, respectively). The remaining EL reasoner, jcel, was slower than most other reasoners, while also failing to process 38 of the 200 ontologies in the given time.

Finally, for the satisfiability checking task in the EL category, Chainsaw processed the highest number of concepts (all 2,000) correctly while also being second fastest with an average of 1s per concept. MORe with both Pellet and HermiT also completed all 2,000 concepts within the given timeouts, while ELK performed fastest on those 1,990 concepts it did process.

4.3 OWL 2 RL results

Only one profile-specific reasoner (BaseVISor) competed in the OWL 2 RL category. Figure 3 shows the results for the three challenges in the RL profile category. Out of the eleven competing reasoners, BaseVISor failed on a significantly large number of ontologies in the classification challenge and processed only 34 of the 197 ontologies correctly. 17 of these failures were due to parsing errors, ten were caused by timeouts that did not return any results, and the remaining failures were due to incorrect results (according to our correctness check). The winning reasoner here was TReasoner, which—despite being the second-slowest reasoner in the group—correctly classified 181 of the 197 ontologies, while most other reasoners correctly processed between 151 and 157 ontologies.

In the consistency checking task, Konclude correctly processed all 197 ontologies, while also performing significantly faster than the other reasoners. Finally, the RL satisfiability category was won by both MORe versions, which correctly processed all 1,970 concepts at an average speed of 0.7s per concept.

5 Results – Live competition

The live competition was performed using only the **classification** task in the OWL 2 DL and EL categories, since this is the task supported by most reasoners. The setup was slightly modified from that of the live competition: rather than running the reasoners until they had processed all ontologies in the corpus, we set a strict timeout of one hour for the EL classification task and two hours for the DL classification task, and measured how many ontologies the reasoners would successfully classify in the given time (applying the same five/six minute

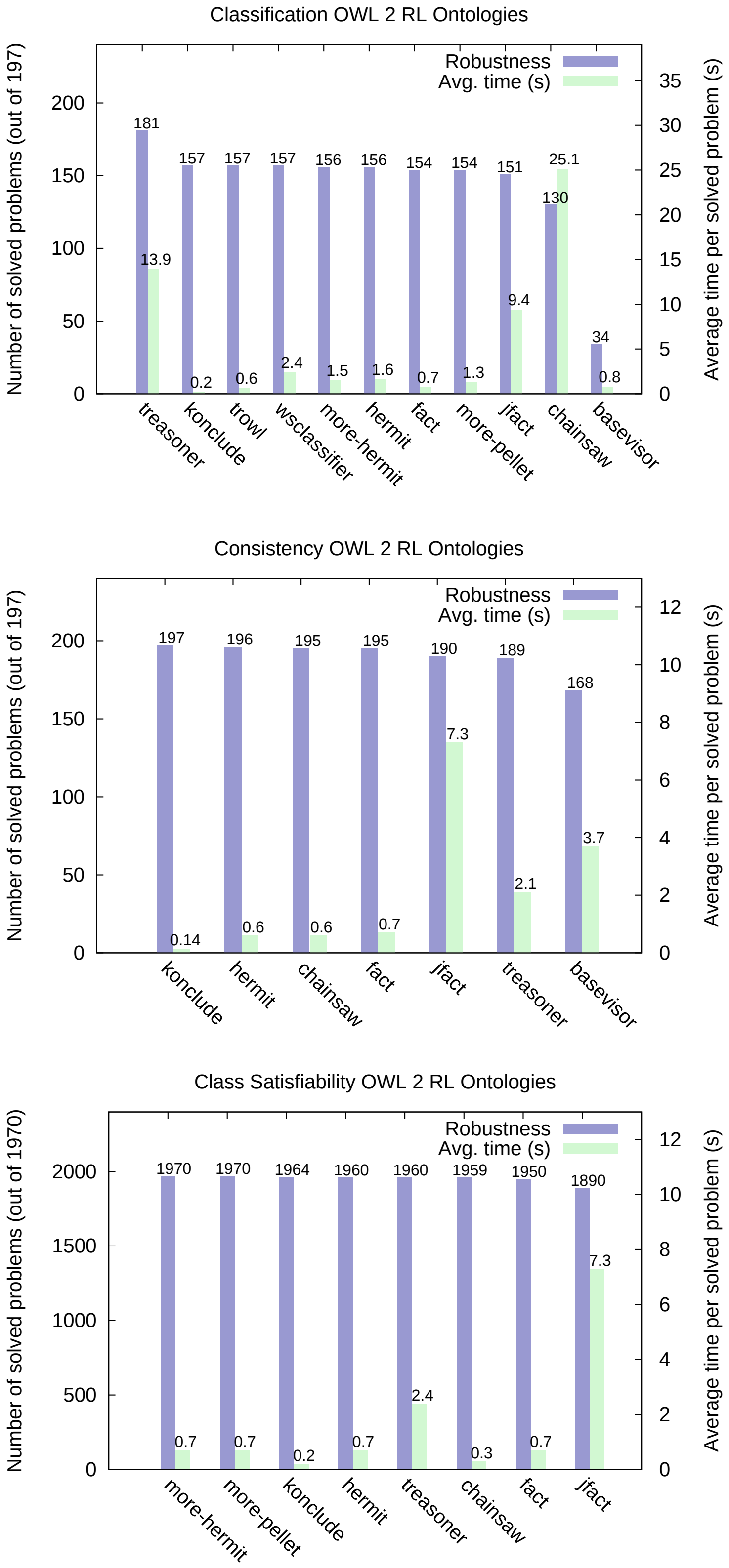

Fig. 3: Results (robustness/average time) for the OWL 2 **RL** category.

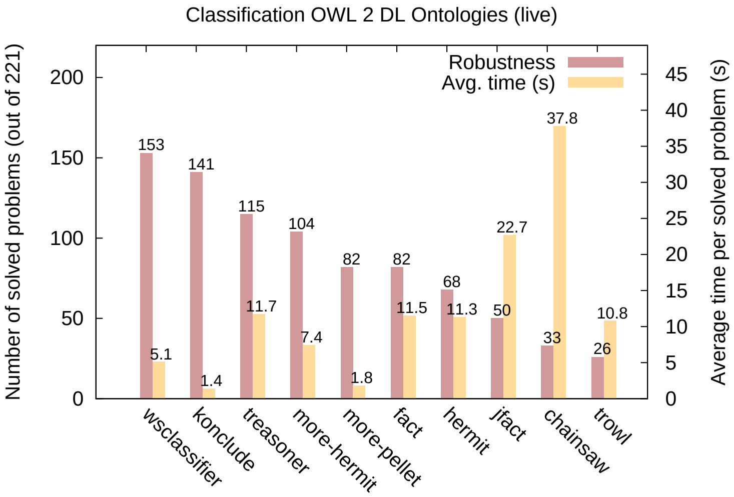

Fig. 4: Results (robustness/average time) for the live competition **DL** category.

timeout per ontology as in the offline competition). As mentioned above, the live competition was performed on Set B, which was entirely different for the DL category, but nearly identical (due to the small number of available EL ontologies) to Set A in the EL category. That is, we expected the results for the DL category to differ from the offline competition, while the results for the EL competition would be largely identical.

The live competition was held on the second day of the Description Logic 2013 workshop, allowing workshop participants to place bets on the reasoner performance, while the current status for each reasoner (number of attempted and number of successfully classified ontologies) was shown and continuously updated on a screen.

5.1 OWL 2 DL classification results

Due to the use of the different test corpus (Set B) in the live competition, we expected a slightly different outcome from the offline competition. And indeed, the winning reasoner (in terms of number of correctly processed ontologies) was WSClassifier, which had shown an average performance in the offline competition. WSClassifier processed 153 out of the 221 ontologies in the test corpus, with an average time of 5.1s per ontology, while the reasoner in second place was Konclude, with 141 ontologies and an average time of 1.4s per ontology. Figure 4 shows an overview of the number of processed ontologies and classification times in the DL category.

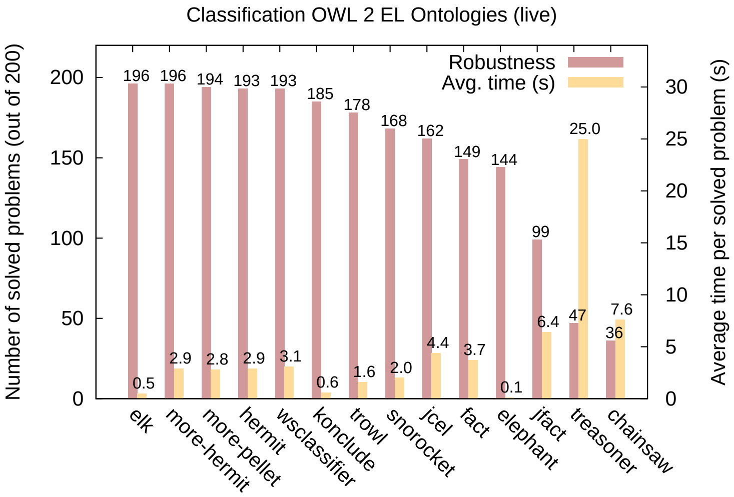

Fig. 5: Results (robustness/average time) for the live competition **EL** category.

5.2 OWL 2 EL classification results

The number of processed ontologies and mean classifications for all reasoners participating in the EL live competition can be found in Figure 5. Perhaps unsurprisingly, in the EL classification challenge the results were very similar to the offline challenge, with ELK classifying 196 out of the 200 ontologies at an average speed of 0.5s per ontology. Again, ELepHant was clearly the fastest reasoner with less than 0.1 seconds per ontology, but it also failed on 56 of the 200 ontologies.

6 Results – User-submitted ontologies

As with the live competition, the results for the user-submitted ontologies presented here are limited to the classification task, as we consider this to be the most relevant (TBox) reasoning task which is supported by all reasoners in the competition. Note that due to the high number of timeouts and errors on some of these ontologies, the correctness of the successful reasoners could not be determined.

6.1 OWL 2 DL classification results

In total, 66 user-submitted ontologies fell into the OWL 2 DL profile, which included the three modified versions of FMA and GALEN discussed above, two versions of the Genomic CDS knowledge base (CDS and CDS-demo), 58 different variants of the Bio KB 101 ontology (of which four were considered to be the most challenging by the ontology developers), and three of the DMOP ontologies.

Except for Chainsaw and TReasoner, all reasoners could successfully classify 54 of the Bio KB ontologies within the five minute timeout, while none of the reasoners processed any of the four 'hard' Bio KB ontologies within the timeout.

The only reasoners that could process both Genomic CDS ontologies were TrOWL and WSClassifier (at an average time of approximately 3 and 8 seconds), while FaCT++ also managed to classify the complete Genomic CDS in 100 seconds. Interestingly, HermiT was the only reasoner to report that the Genomic CDS ontology was inconsistent.

TrOWL and WSClassifier were also the only reasoners to classify the FMA and GALEN modifications within the timeout (perhaps unsurprisingly, since WSClassifier was tuned to work with these ontologies), while both Chainsaw and FaCT++ successfully processed the two GALEN versions, and MORe-Pellet processed the GALEN-FNL version in 14 seconds. For the remaining ontologies, all reasoners except FaCT++ and TrOWL reported datatype errors.

At an average of 0.17 seconds per processed ontology, Konclude was clearly fastest, while most other reasoners also managed average times of less than five seconds for the ontologies they processed correctly.

6.2 OWL 2 EL classification results

There were 19 user-submitted OWL 2 EL ontologies, 18 of which were variants of the Bio KB 101 ontology, and the FTC knowledge base. Neither Chainsaw nor ELepHant could process any of the Bio KB ontologies within the five minute timeout, while ELK reported a parsing error. The remaining reasoners, except Snorocket, processed all 18 Bio KB ontologies correctly within the timeout, with Konclude being fastest at 0.1 seconds per ontology.

ELK, Konclude, and WSClassifier all successfully processed the FTC KB, with ELK clearly being fastest at five seconds (it did, however, ignore three ObjectPropertyRange axioms which are outside the OWL 2 EL fragment supported by ELK), and the other two reasoners taking between 20 and 30 seconds. The remaining reasoners either timed out or reported an error for this ontology.

6.3 OWL 2 RL classification results

All 18 ontologies in the OWL 2 RL profile were variants of Bio KB 101. BaseVISor failed to parse the input on all files, while Chainsaw timed out on 15 of the ontologies. The remaining reasoners all classified the ontologies correctly within the five minute timeout, with Konclude processing the ontologies at an average of 0.15 seconds.

7 Summary

In this report we presented an overview of the methodology and results of the ORE reasoner competition for the different categories, OWL 2 profiles, and test corpora. There were a total of 14 OWL reasoners submitted for participation in the competition, which made it all the more successful. Out of these, 5 were profile specific reasoners (4 OWL 2 EL and 1 OWL 2 RL) while 9 were OWL 2 DL reasoners or supported a large fragment of $\mathcal{SROIQ}(D)$ not included within the OWL 2 EL, RL or QL profiles. The reasoners were evaluated with a random sample of ontologies from known repositories, on three standard reasoning tasks: classification, consistency checking, and concept satisfiability. In the competition we gave preference to how robust the systems were, that is, the number of tests correctly passed within the given timeout, rather than reasoning times alone. The top 3 reasoners for each category are listed below:

OWL 2 DL Ontologies

- *Classification:* (1) HermiT (2) MORe-HermiT/MORe-Pellet (3) Konclude
- *Consistency:* (1) Konclude (2) FaCT++ (3) Chainsaw
- *Satisfiability:* (1) Konclude (2) MORe-Pellet/MORe-HermiT (3) TReasoner
- *Classification (live):* (1) WSClassifier (2) Konclude (3) TReasoner

OWL 2 EL Ontologies

- *Classification:* (1) ELK (2) WSClassifier (3) MORe-HermiT
- *Consistency:* (1) ELK (2) HermiT (3) Konclude
- *Satisfiability:* (1) Chainsaw (2) MORe-Pellet (3) TrOWL
- *Classification (live):* (1) ELK (2) MORe-HermiT/MORe-Pellet (3) HermiT

OWL 2 DL Ontologies

- *Classification:* (1) TReasoner (2) Konclude (3) TrOWL
- *Consistency:* (1) Konclude (2) HermiT (3) Chainsaw
- *Satisfiability:* (1) MORe-HermiT/MORe-Pellet (2) Konclude (3) HermiT

Additionally, the MORe and ELepHant reasoners were also given a special recognition prize. MORe was selected as the *best newcomer reasoner* since it consistently performed well in terms of time and robustness. The ELepHant reasoner, although it struggled with a high number of errors, was incredibly fast for the ontologies that it was able to classify correctly, and so was awarded a *special mention*. We look forward to seeing the evolution of these novel reasoners.

Regarding the user-submitted ontologies, it is interesting to see that most reasoners could either process *all* or *none* of the Bio KB ontologies. When they did process them, the classification times were fairly uniform. The results for the GALEN and FMA modifications, which were specifically developed for testing with WSClassifier, confirmed the robustness of the reasoner on these ontologies; however, the other two reasoners which *could* process the GALEN modifications (Chainsaw and FaCT++) were significantly faster within the timeout. Our experiments on the Genomic CDS ontologies confirmed the reports of the ontology developer [20] who found that out of the now 'mainstream' reasoners, only TrOWL could process the ontology in reasonable time, while HermiT (falsely) reported an inconsistency error. While we have seen that WSClassifier could also process the ontology, the correctness of the classification result is unclear, since WSClassifier does not support qualified number restrictions which are heavily used in Genomic CDS.

Finally, we have only carried out our benchmark with a fixed timeout of five minutes in the main offline and live competitions, which may have been too short for some of these ontologies, e.g. the four 'challenging' Bio KB ontologies could not be processed by any of the reasoners Thus, we are planning to re-run these tests with longer timeouts in the near future.

Acknowledgements

We thank Konstantin Korovin (supported by the Royal Society grant RG080491) at the University of Manchester who kindly provided us with the PC cluster for the competition. We also thank the developers of the submitted reasoners and ontologies for their invaluable effort. We also gratefully acknowledge the support of the ORE workshop sponsor: B2i Healthcare (`https://www.b2international.com/`). Ernesto Jimenez-Ruiz was supported by the Seventh Framework Program (FP7) of the European Commission under Grant Agreement 318338, 'Optique', and the EPSRC projects Score!, ExODA and MaSI[3].

References

1. Armas Romero, A., Cuenca Grau, B., Horrocks, I., Jiménez-Ruiz, E.: MORe: a Modular OWL Reasoner for Ontology Classification. In: OWL Reasoning Evaluation Workshop (ORE) (2013)
2. Chaudhri, V.K., Wessel, M.A., Heymans, S.: KB_Bio_101: A Challenge for OWL Reasoners. In: 2nd OWL Reasoner Evaluation Workshop (ORE) (2013)
3. Cuenca Grau, B., Horrocks, I., Motik, B., Parsia, B., Patel-Schneider, P.F., Sattler, U.: OWL 2: The next step for OWL. J. Web Sem. 6(4), 309–322 (2008)
4. Glimm, B., Horrocks, I., Motik, B., Shearer, R., Stoilos, G.: A novel approach to ontology classification. J. of Web Semantics 10(1) (2011)
5. Golbeck, J., Fragoso, G., Hartel, F.W., Hendler, J.A., Oberthaler, J., Parsia, B.: The National Cancer Institute's Thésaurus and Ontology. J. Web Sem. 1(1), 75–80 (2003)
6. Gonçalves, R.S., Parsia, B., Sattler, U.: Categorising logical differences between OWL ontologies. In: ACM Conference on Information and Knowledge Management (CIKM) (2011)
7. Grigoryev, A., Ivashko, A.: TReasoner: System Description. In: 2nd OWL Reasoner Evaluation Workshop (ORE) (2013)
8. Horridge, M., Bechhofer, S.: The OWL API: A java api for owl ontologies. Semantic Web 2(1), 11–21 (2011)
9. Horrocks, I., Patel-Schneider, P.F., van Harmelen, F.: From $\mathcal{SHIQ}$ and RDF to OWL: the making of a web ontology language. J. Web Sem. 1(1), 7–26 (2003)
10. Horrocks, I., Yatskevich, M., Jiménez-Ruiz, E. (eds.): Proceedings of the 1st International Workshop on OWL Reasoner Evaluation (ORE), CEUR Workshop Proceedings, vol. 858. CEUR-WS.org (2012)
11. Kazakov, Y., Krötzsch, M., Simancik, F.: Concurrent classification of EL ontologies. In: International Semantic Web Conference (ISWC). pp. 305–320 (2011)

12. Keet, C.M., Lawrynowicz, A., d'Amato, C., Hilario, M.: Modeling issues and choices in the Data Mining OPtimization Ontology. In: OWL: Experiences and Directions (2013)
13. Matentzoglu, N., Bail, S., Parsia, B.: A corpus of owl dl ontologies. In: 26th International Workshop on Description Logics (DL). pp. 829–841 (2013)
14. Mendez, J.: jcel: A Modular Rule-based Reasoner. In: 1st OWL Reasoner Evaluation Workshop (ORE) (2012)
15. Metke Jimenez, A., John Lawley, M.: Snorocket 2.0: Concrete Domains and Concurrent Classification. In: 2nd OWL Reasoner Evaluation Workshop (ORE) (2013)
16. Motik, B., Cuenca Grau, B., Horrocks, I., Wu, Z., Fokoue, A., Lutz, C.: OWL 2 web ontology language profiles. W3C Recommendation (2009)
17. Noy, N.F., Shah, N.H., Whetzel, P.L., Dai, B., Dorf, M., Griffith, N., Jonquet, C., Rubin, D.L., Storey, M.A.D., Chute, C.G., Musen, M.A.: Bioportal: ontologies and integrated data resources at the click of a mouse. Nucleic Acids Research 37(Web-Server-Issue), 170–173 (2009)
18. Rector, A.L., Rogers, J.E., Zanstra, P.E., Van Der Haring, E., Openg: OpenGALEN: open source medical terminology and tools. AMIA Annu Symp Proc (2003)
19. Rosse, C., Mejino Jr., J.: A reference ontology for biomedical informatics: the Foundational Model of Anatomy. J. Biomed. Informatics 36(6), 478–500 (2003)
20. Samwald, M.: Genomic CDS: an example of a complex ontology for pharmacogenetics and clinical decision support. In: 2nd OWL Reasoner Evaluation Workshop (ORE) (2013)
21. Schulz, S., Cornet, R., Spackman, K.A.: Consolidating SNOMED CT's ontological commitment. Applied Ontology 6(1), 1–11 (2011)
22. Sertkaya, B.: The ELepHant Reasoner System Description. In: 2nd OWL Reasoner Evaluation Workshop (ORE) (2013)
23. Simancik, F., Kazakov, Y., Horrocks, I.: Consequence-Based Reasoning beyond Horn Ontologies. In: 22nd International Joint Conference on Artificial Intelligence (IJCAI). pp. 1093–1098 (2011)
24. 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)
25. Song, W., Spencer, B., Du, W.: A Transformation Approach for Classifying ALCHI(D) Ontologies with a Consequence-based ALCH Reasoner. In: 2nd OWL Reasoner Evaluation Workshop (ORE) (2013)
26. Thomas, E., Pan, J.Z., Ren, Y.: TrOWL: Tractable OWL 2 Reasoning Infrastructure. In: 7th Extended Semantic Web Conference (ESWC). pp. 431–435 (2010)
27. Tsarkov, D., Horrocks, I.: FaCT++ Description Logic Reasoner: System Description. In: Third International Joint Conference on Automated Reasoning (IJCAR). pp. 292–297 (2006)
28. Tsarkov, D., Palmisano, I.: Chainsaw: a Metareasoner for Large Ontologies. In: 1st OWL Reasoner Evaluation Workshop (ORE) (2012)
29. Wrigley, S.N., Garcia-Castro, R., Nixon, L.J.B.: Semantic Evaluation At Large Scale (SEALS). In: The 21st World Wide Web Conf., WWW (Companion Volume). pp. 299–302 (2012), http://www.seals-project.eu