CEDAR: a Fast Taxonomic Reasoner Based on Lattice Operations

Samir Amir and Hassan Aït-Kaci

Université Claude Bernard Lyon 1, France

{samir.amir,hassan.ait-kaci}@univ-lyon1.fr

Abstract. Taxonomy classification and query answering are the core reasoning services provided by most of the Semantic Web (SW) reasoners. However, the algorithms used by those reasoners are based on Tableau method or Rules. These well-known methods in the literature have already shown their limitations for large-scale reasoning. In this demonstration, we shall present the CEDAR system for classifying and reasoning on very large taxonomies using a technique based on lattice operations. This technique makes the CEDAR reasoner perform on par with the best systems for concept classification and several orders-of-magnitude more efficiently in terms of response time for query-answering. The experiments were carried out using very large taxonomies (Wikipedia: 111599 sorts, MESH: 286381 sorts, NCBI: 903617 sorts and Biomodels: 182651 sorts).[1] The results achieved by CEDAR were compared to those obtained by well-known Semantic Web reasoners, namely FaCT++, Pellet, HermiT, TrOWL, SnoRocket and RacerPro.

Keywords: Semantic Reasoning, Lattice Operations, Partial-Order Encoding

1 Introduction

The demo will demonstrate how an implementation of a system based on lattice operations can be used for taxonomic reasoning in a robust and scalable way. Indeed, this challenge was defined on the frame of CEDAR project. [2] CEDAR system is a Boolean reasoner that can support a huge amount of sorts without any noticeable degradation of query evaluation performance. The essential technique we used for implementing the CEDAR reasoner is based on bit-vector encoding. This method was proposed over 20 years ago for implementing efficient lattice operations [1]. Since the common aspect of all Semantic Web reasoning systems is the representation and processing of taxonomic data, we implemented a taxonomic concept classification and Boolean query-answering system based

[1] We use "sort" as a synonym of atomic "class" or "concept." In other words, sorts are partially ordered symbols.

[2] Constraint Event-Driven Automated Reasoning—http://cedar.liris.cnrs.fr

on the method described above. We made measurements over several very large taxonomies under the exact same conditions for so-called TBox reasoning. A comparative evaluation was conducted to assess the performance of CEDAR over the mentioned systems which have been implemented by using OWL-API. [3] In terms of query-answering response time, CEDAR is several orders-of-magnitude more efficient than that of the best existing SW reasoning systems.

2 Lattice Operations for Taxonomic Reasoning

The CEDAR reasoner is an implementation in Java of the technique described as bottom-up transitive-closure encoding in [1]. This technique consists in representing the elements of a taxonomy (an arbitrary poset) as bit vectors. Thus, each element has a code (a bit vector) carrying a "`1`" in the position corresponding to the index of any other elements that it subsumes. In this manner, the three Boolean operations on sorts are readily and efficiently performed as their corresponding operations on bit-vectors. This is possible if the bit-vectors encoding the sorts comprising a taxonomy are obtained by computing the reflexive-transitive closure of the "is-a" relation derived from the subsort declarations.

3 Demonstration

The demo will show how CEDAR differs from existing and known reasoners in terms of classification (Figures 1 and 2) and query answering (Figures 3 and 4) where it is several orders-of-magnitude more efficient than other reasoners. Developed software integrates six other reasoners to provide a comparison with CEDAR (HermiT [4], FaCT++ [7], RacerPro [2], TrOWL [6], Pellet [5] and SnoRocket [3] all of which use the OWL-API interface). The proposed structure of the demonstration is the following:

- Classification performance using very large taxonomies as Wikipedia[4] (111599 sorts), NCBI[5](903617 sorts), MESH[6] (286381 sorts) and Biomodels[7] (182651 sorts). The demo will show the results illustrated in Figures 1 and 2 where CEDAR is always among the best three out of six reasoners.
- Query Answering using boolean queries (`and`, `or` and `not`) involving a large number of concepts (up to 100 concepts). The obtained results will be compared with those of traditional reasoners as shown in Figures 3 and 4.
- Saving and reusing an encoded taxonomy. With CEDAR, there is no need to perform a classification each time. A classified taxonomy can be saved and reused.
- Detecting Cycles: We will show how to detect cycles in taxonomies, which are a particular case of inconsistency resulting from modeling errors.

[3] http://owlapi.sourceforge.net
[4] http://www.h-its.org/english/research/nlp/download/wikitaxonomy.php
[5] http://www.ncbi.nlm.nih.gov/Taxonomy/taxonomyhome.html/
[6] http://www.nlm.nih.gov/mesh/meshhome.html
[7] https://code.google.com/p/sbmlharvester/

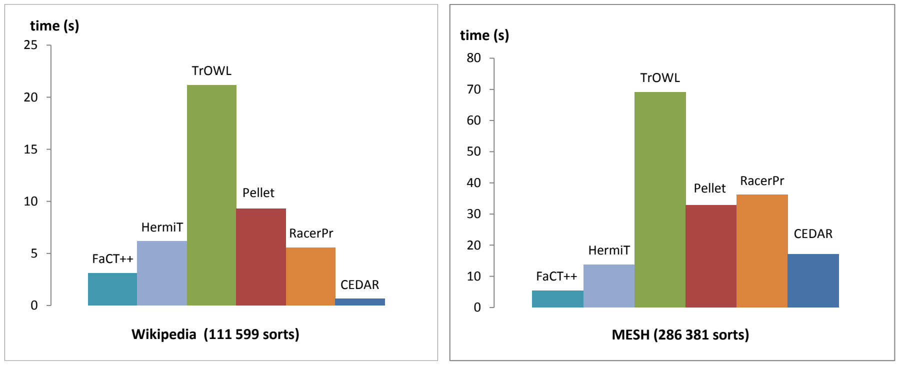

Fig. 1. Classification time per reasoner for the Wikipedia and MeSH taxonomies

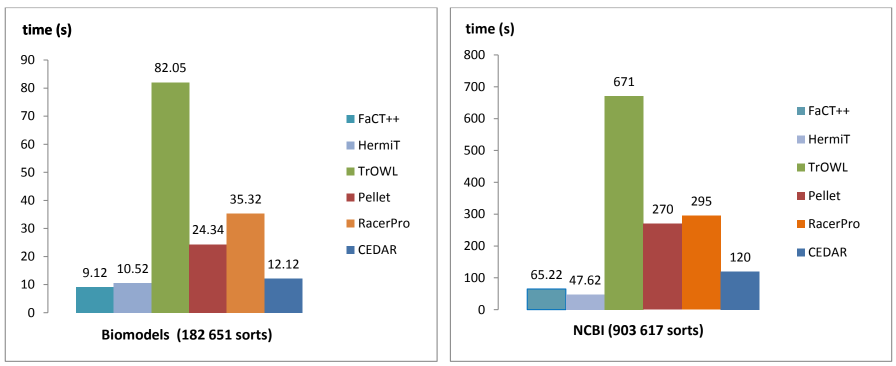

Fig. 2. Classification time per reasoner for the Biomodels and NCBI taxonomies

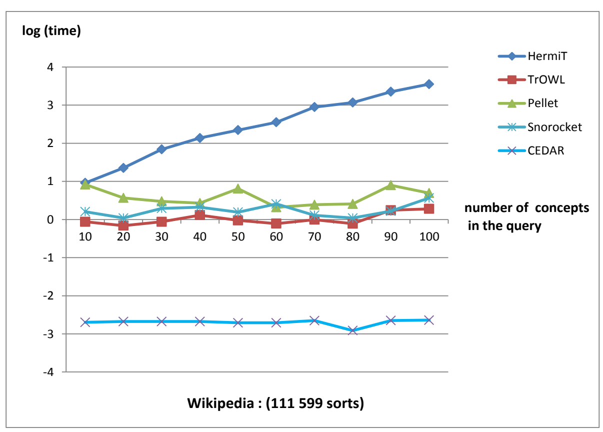

Fig. 3. Query response time per reasoner for the Wikipedia taxonomy

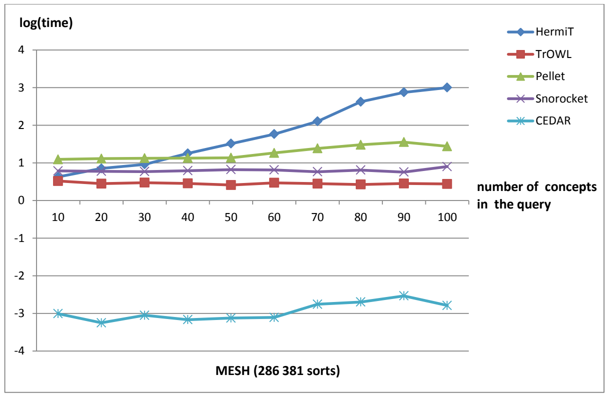

Fig. 4. Query response time per reasoner for the MESH taxonomy

4 Acknowledgements

The authors wish to thank Mohand-Saïd Hacid for his comments. This work was carried out as part of the CEDAR Project (Constraint Event-Driven Automated Reasoning) under the *Agence Nationale de la Recherche* (ANR) Chair of Excellence grant N^o **ANR-12-CHEX-0003-01** at the *Université Claude Bernard Lyon 1*.

References

1. Aït-Kaci, H., Boyer, R., Lincoln, P., and Nasr, R. Efficient implementation of lattice operations. *ACM Transactions on Programming Languages and Systems 11*, 1 (January 1989), 115–146.
2. Haarslev, V., Hidde, K., Möller, R., and Wessel, M. The RacerPro knowledge representation and reasoning system. *Semantic Web Journal 1* (March 2011), 1–11.
3. Lawley, M. J., and Bousquet, C. Fast classification in Protégé: Snorocket as an OWL 2 EL reasoner. In *Proceedings of the 2nd Australasian Ontology Workshop: Advances in Ontologies* (Adelaide, Australia, December 2010), T. Meyer, M. A. Orgun, and K. Taylor, Eds., AOW'10, ACS, pp. 45–50.
4. Shearer, R., Motik, B., and Horrocks, I. HermiT: A highly-efficient OWL reasoner. In *Proceedings of the 5th International Workshop on OWL Experiences and Directions* (Karlsruhe, Germany, October 2008), U. Sattler and C. Dolbear, Eds., OWLED'08, CEUR Workshop Proceedings.
5. Sirin, E., Parsia, B., Grau, B. C., Kalyanpur, A., and Katz, Y. Pellet: A practical OWL-DL reasoner. *Journal of Web Semantics 5*, 2 (June 2007), 51–53.
6. Thomas, E., Pan, J. Z., and Ren, Y. TrOWL: Tractable OWL 2 reasoning infrastructure. In *Proceedings of the 7th Extended Semantic Web Conference* (Heraklion, Greece, May-June 2010), ESWC'10, Springer-Verlag, pp. 431–435.
7. Tsarkov, D., and Horrocks, I. FaCT++ description logic reasoner: System description. In *Proceedings of the 3rd International Joint conference on Automated Reasoning* (Seattle, WA, USA, August 2006), U. Furbach and N. Shankar, Eds., IJCAR'06, Springer-Verlag, pp. 292–297.