Avoiding Subsumption Tests During Classification Using the Atomic Decomposition

Haoruo Zhao[1], Uli Sattler[1], and Bijan Parsia[1]

The University of Manchester, Oxford Road, Manchester, M13 9PL, UK
{haoruo.zhao,uli.sattler,bijan.parsia}@manchester.ac.uk

Classifying a TBox requires many subsumption tests, for example $\mathcal{O} \models^? Animal \sqsubseteq Person$. For n the number of *concept names* in $\mathcal{O}$, a naive algorithm involves a quadratic number of these problems, and various optimisations, called *Enhanced Traversal* algorithms, have been developed [1, 4]. In DLs, subsumption is decidable but possibly computationally expensive, e.g. for $\mathcal{SROIQ}(\mathcal{D})$, it is 2NExpTime [5]. In this paper, we try to modify the standard Enhanced Traversal algorithm to work over a set of *modules* [3, 9] to optimize this core reasoning task, classification. There are two potential benefits: to shorten the subsumption test times (i.e., *easification*) and to reduce the number of subsumption tests by avoiding some tests (i.e., *avoidance*).

In [7, 6], we investigated the potential sources of performance benefits for blackbox modular reasoning (with a single delegate reasoner). Such schemes have an enormous advantage of implementation flexibility and ease, but potentially have difficulty with overhead and missed opportunities. One insight from [7] is that easification (at least for ontologies which can be processed within a few hours) is not likely to be a source of dramatic performance gain. Worse, in a blackbox context, we face a trade-off of maximising potential avoidance (e.g., using smaller modules) and minimising redundancy (since small modules tend to overlap a lot, we end up *redoing* work). It was not clear that any blackbox approach would show reliable benefits. Existing systems based on modularity, in particular, MORe [8] and Chainsaw [11], do not provide reliable improvements over their non-modular component reasoners except (for MORe) in very particular circumstances. Moreover, where they do provide significant performance benefits, the *reason* for those benefits is unclear, i.e., whether they are due to easification, avoidance, or a mix.

In this paper, we explore whether modifying the core Traversal algorithm to be module aware is worth attempting. Since, touching the internals of complex and highly tuned systems such as modern description logic reasoners is a labor intensive and fraught affair, and since existing Enhanced Traversal algorithms are already highly optimised, our first aim is to gather data about the total amount of (further) avoidance possible for a given reasoner (Pellet [10]) over a significant corpus of ontologies (BioPortal). In other words, we want to verify whether "module aware subsumption avoidance" is meaningful. Hence we modify Pellet to be moderately module aware: we insert, between its Enhanced Traversal algorithm and its subsumption checker, a module aware component that simply determines whether the ontology's *Atomic Decomposition*[1] contains enough information to answer the subsumption test in question. In this way, we minimise the overhead of communicating between the core Pellet components and our avoidance checker and gain insight into the potential avoidance gains of module awareness for an existing Traversal algorithm.

We find that in BioPortal, on a significant number of ontologies, almost all subsumption tests performed by Pellet can be avoided–despite the fact that Pellet uses an Enhanced Traversal algorithm. In Figure 1, we see an overview of our results: each point represents an ontology, the X-axis the number of Subsumption Tests (STs) carried out by Pellet and the Y-axis the percentage of STs avoided.

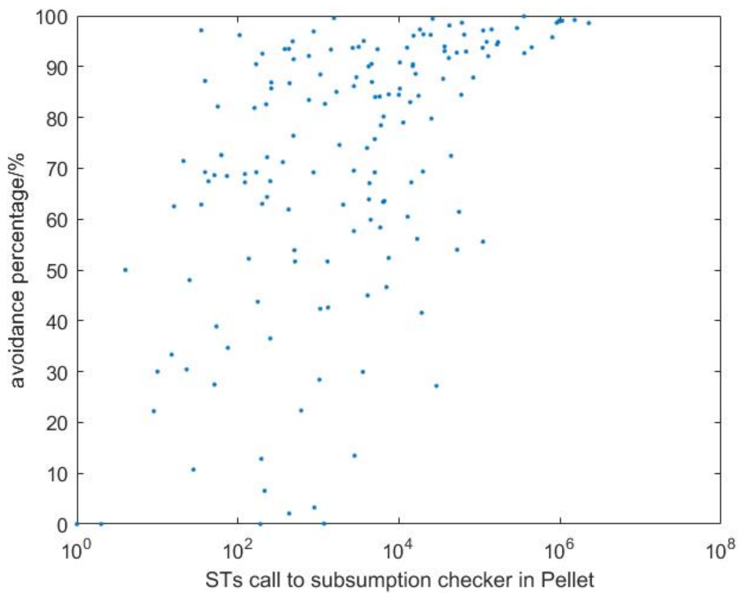

Fig. 1. Avoidance Percentage against calls to ST checker in Pellet on 166 ontologies

More precisely, we find that for a third of the ontologies, more than 90% of subsumption tests performed by Pellet during classification can be avoided by such a module aware system. For more than half of the ontologies, more than 70% can be avoided.

These results suggest that designing and implementing a full-blown, module aware traversal algorithm is likely to be net beneficial. We are currently investigating module-based avoidance into Traversal algorithm.

[1] A modular partition of an ontology computable in polynomial time [2] representing all locality-based modules.

References

1. Baader, F., Hollunder, B., Nebel, B., Profitlich, H.J., Franconi, E.: An empirical analysis of optimization techniques for terminological representation systems. In: Proc. of the 3rd Int. Conf. on the Principles of Knowledge Representation and Reasoning (KR92). pp. 270–281 (1992)
2. Del Vescovo, C.: The modular structure of an ontology: Atomic decomposition towards applications
3. Del Vescovo, C., Gessler, D.D., Klinov, P., Parsia, B., Sattler, U., Schneider, T., Winget, A.: Decomposition and modular structure of bioportal ontologies. In: International Semantic Web Conference. pp. 130–145. Springer (2011)
4. Glimm, B., Horrocks, I., Motik, B., Shearer, R., Stoilos, G.: A novel approach to ontology classification. Web Semantics: Science, Services and Agents on the World Wide Web **14**, 84–101 (2012)
5. Kazakov, Y.: $\mathcal{RIQ}$ and $\mathcal{SROIQ}$ are harder than $\mathcal{SHOIQ}$. International Joint-Conference on Automated Reasoning pp. 292–297 (2008)
6. Matentzoglu, N., Parsia, B., Sattler, U.: OWL reasoning: Subsumption test hardness and modularity. Journal of Automated Reasoning **60**(4), 385–419 (2018)
7. Matentzoglu, N.A.: Module-based classification of OWL ontologies. The University of Manchester (2016)
8. Romero, A.A., Grau, B.C., Horrocks, I.: More: Modular combination of OWL reasoners for ontology classification. In: International Semantic Web Conference. pp. 1–16. Springer (2012)
9. Romero, A.A., Kaminski, M., Grau, B.C., Horrocks, I.: Module extraction in expressive ontology languages via datalog reasoning. J. Artif. Intell. Res. **55**, 499–564 (2016). https://doi.org/10.1613/jair.4898, `https://doi.org/10.1613/jair.4898`
10. Sirin, E., Parsia, B., Grau, B.C., Kalyanpur, A., Katz, Y.: Pellet: A practical OWL-DL reasoner. Web Semantics: science, services and agents on the World Wide Web **5**(2), 51–53 (2007)
11. Tsarkov, D., Palmisano, I.: Chainsaw: a Meta Reasoner for large ontologies. In: OWL Reasoner Evaluation Workshop (ORE 2012). Citeseer (2012)