Finding Justifications by Approximating Core for Large-scale Ontologies

Mengyu Gao[1] , **Yuxin Ye**[1,2*] , **Dantong Ouyang**[1,2] and **Bin Wang**[1]
[1]College of Computer Science and Technology, Jilin University, Changchun 130012
[2]Key Laboratory of Symbol Computation and Knowledge Engineering (Jilin University), Ministry of Education, Changchun 130012
gaomengyu_jlu@163.com, {yeyx, ouyd}@jlu.edu.cn, liren819@163.com

Abstract

Finding justifications for an entailment is one of the major missions in the field of ontology research. Recent advances on finding justifications w.r.t. the light-weight description logics focused on encoding this problem into a propositional formula, and using SAT-based techniques to enumerate all MUSes (minimally unsatisfiable subformulas). It's necessary to import more optimized techniques into finding justifications as emergence of large-scale real-world ontologies. In this paper, we propose a new strategy which introduces local search (in short, LS) technique to compute the approximating core before extracting an exact MUS. Although it is based on a heuristic and LS, such technique is complete in the sense that it always delivers a MUS for any unsatisfiable SAT instance. Our method will find the justifications for large-scale ontologies more effectively.

1 Introduction and Background

Recent years, automated reasoning techniques in description logics have been developed for computing not only concept subsumption, but also to find justifications causing each subsumption. In the early research stages, Schlobach et al. first proposed the glass-box approach based on the tableau extension strategy to find Minimal unsatisfiability-preserving subTBoxes(MUPS), which is actually a special justification, in any expressive description logic languages. The main idea of the glass-box is to use the rules of the tableau to expand. In the process of expansion, once a conflict occurs, the elements on the path that caused the conflict are the causes(may not be the minimal) of the inconsistency of the ontology. Certainly, there may be more than one such path. After that, Kalyanpur et al. extended the interpretation of the unsatisfiable concept to the interpretation of general axioms and named it as justification. They introduced a black-box method which could be easily and robustly implemented to debugging ontologies. The black-box algorithm uses the reasoner solely as a subroutine and the internals of the reasoner do not need to be modified.

*Contact Author

With the advent of a series of the light-weight ontology(such as the $\mathcal{EL}$ family, etc.), it began to consider novel strategies to find justifications for such ontologies. Since the language complexity of the ontology is reduced, the size of the ontology becomes larger, relatively. Considering the characteristics of the light-weight ontology, it is feasible to *encode* the classification of the ontology such as the $\mathcal{EL}$ family into the form of the Horn proposition formula, thus transforming finding justifications into computing all MUSes of the propositional formula. Baader used SAT based to find justifications for SNOMED-CT ontology in [Baader and Suntisrivaraporn, 2007]. The method built an exponential-size monotone propositional formula from a classification of terminology. [Sebastiani and Vescovi, 2009] used the boolean constraint propagation and conflict analysis of the SAT solver to compute the concept subsumptions and perform finding justifications. It's based on a polynomial-size Horn propositional formula. [Arif *et al.*, 2015] exploited an important relationship between minimal axiom sets and minimal unsatisfiable subformulas in the propositional domain. The encoding of this method also adopts a polynomial-size Horn propositional formula. In addition, [Ignatiev *et al.*, 2017] describes different improvements that have been considered since the translation was firstly proposed. Clauses of the encoded unsatisfiable formula are partitioned into groups. The recent work [Kazakov and Skocovský, 2018] is based on the extension of the resolution method with so-called answer literals. According to the experiment of the paper, it was able to compute all justifications for all direct subsumptions of SNOMED-CT in about 1.5 hour.

2 Related Work

Recent years, some of approximation methods to reduce the set of clauses that must be tested for exclusion from the MUS have been proposed. The following methods were compared in [Grégoire *et al.*, 2007]. **AMUSE** proposed the algorithm adapting the "learning process" of a modern SAT solver to identify unsatisfiable subformulas rather than search for satisfying assignments. The unsatisfiable cores obtained by this way could be helpful in diagnosing the causes of infeasibility in large systems. **Zcore** served to find a fixed point in the unsatisfiable sub-formulas computed by proof-producing SAT solvers, that is, it aimed to remove irrelevant clauses that can be included by SAT solvers while generating an unsatisfiable sub-formula. **AOMUS** associating a score with each clause was based on LS exploited the fact that clauses in MUSes are likely to be unsatisfied often during LS. Experimental results illustrated that **AOMUS** proved more competitive with respect to both the required run-times and the quality of the delivered approximations of MUSes, in the sense that smaller-sizes approximations were obtained most often. In this paper, we adopt LS technique to optimize finding justifications based on SAT encoding. It aims to pre-select a subset of axioms corresponding to an accurate justification as small as possible from a large-scale ontology.

3 Research Goals and Expected Contributions

SAT solver is used to determine the satisfiability of formulas during finding justifications. It's complete but expensive. However, the computational cost based on incomplete LS technology is much lower. We use it instead of SAT solver to detect an approximating core in advance. The approximating core is identified by taking critical clause into account. Intuitively, a critical clause is a falsified clause that becomes true thanks to a local search flip only when some other clauses become false at the same time. It must be complete in the sense that it always delivers a MUS for any unsatisfiable SAT instance. By introducing this incomplete LS before the complete SAT solving, the scope of the axioms for finding justifications will be reduced to the single-digit scale. A justification will be computed by iterating process for an amount of time up-to-linear in the size of the first approximating core was found.

4 Preliminary Results

The experiment mainly compares the time and the size of clauses calculated with Zcore and LS. We modified the ontologies in the same way as [Sebastiani and Vescovi, 2009] in order to be compliant with the input grammar of EL+2SAT. It removed the comments and annotations of the ontologies and also rewrote the axioms with purely syntactic transformations. Table 1 records the detailed experimental results. We experimented with NOT-GALEN, the Gene Ontology(GO), NCI and FULL-GALEN. In the query file for the four ontologies, 20 queries were randomly selected for the experiment. #var represents the number of literals in the formula obtained by encoding, and #cla corresponds to the number of the clauses. It also records the computing time and the size of the approximating core for each query using LS and Zcore, respectively. Calculating in the approximating way can quickly locate the ontology used for finding justifications from the original scale in 4th power of 10 to the single digit level. LS always delivers an approximating core within a reasonable time less than Zcore, and delivered clauses are often similar to the clauses obtained by Zcore.

Ontologies	#var	#cla	Time(s) (Zcore/LS)	Size of Clauses (Zcore/LS)
NOT-GALEN	23398	26720	0.260 / 0.162	5.45 / 7.20
NCI	78454	46802	0.620 / 0.131	4.50 / 4.00
GO	64327	54230	0.42 / 0.123	3.90 / 4.45
FULL-GALEN	777445	1377320	12.85 / 9.32	10.25 / 11.33

Table 1: Partial experimental results

5 Conclusion and Future Work

Our approach is based on encoding the ontology classification procedure into Horn clauses. It introduces the process of computing the approximating core by LS before calculating a MUS. Experiments show that the approach proposed in this paper deserves further study. From the experimental data, the method can greatly reduce the scale of the axioms used for finding justifications. It is also proved that the larger the scale of the ontology, the more obvious the effect of LS. The work of calculating the approximating core by using LS has been completed. In the following, we will use the Insertion method to calculate the MUS, and then use the method proposed in [Ignatiev *et al.*, 2017] to calculate the MUSes. At last, MUSes are decoded into all justifications for given queries. We expect that the results of the experiment will be very substantial and of practical value.

References

[Arif *et al.*, 2015] M. Fareed Arif, Carlos Mencía, and João Marques-Silva. Efficient axiom pinpointing with EL2MCS. In *KI 2015: Advances in Artificial Intelligence - 38th Annual German Conference on AI, Dresden, Germany, September 21-25, 2015, Proceedings*, pages 225–233, 2015.

[Baader and Suntisrivaraporn, 2007] Franz Baader and Boontawee Suntisrivaraporn. Pinpointing in the description logic $\mathcal{EL}^+$. In *German Conference on Advances in Artificial Intelligence*, 2007.

[Grégoire *et al.*, 2007] Éric Grégoire, Bertrand Mazure, and Cédric Piette. Local-search extraction of muses. *Constraints*, 12(3):325–344, 2007.

[Ignatiev *et al.*, 2017] Alexey Ignatiev, João Marques-Silva, Carlos Mencía, and Rafael Peñaloza. Debugging EL+ ontologies through horn MUS enumeration. In *Proceedings of the 30th International Workshop on Description Logics, Montpellier, France, July 18-21, 2017.*, 2017.

[Kazakov and Skocovský, 2018] Yevgeny Kazakov and Peter Skocovský. Enumerating justifications using resolution. In *Automated Reasoning - 9th International Joint Conference, IJCAR 2018, Held as Part of the Federated Logic Conference, FloC 2018, Oxford, UK, July 14-17, 2018, Proceedings*, pages 609–626, 2018.

[Sebastiani and Vescovi, 2009] Roberto Sebastiani and Michele Vescovi. Axiom pinpointing in lightweight description logics via horn-sat encoding and conflict analysis. In *International Conference on Automated Deduction*, 2009.