

A Preliminary Comparison of Forgetting Solutions Computed using SCAN, LETHE and FAME

Ruba Alassaf and Renate A. Schmidt

School of Computer Science, The University of Manchester, UK

Abstract. *Forgetting, in description logic, is a non-standard reasoning technique. The aim of this technique is to eliminate concept and role symbols from a knowledge base, whilst preserving all logical consequences up to the remaining symbols. In this research, three forgetting tools were used to understand the relationship between the approaches on which they are based: SCAN, LETHE and FAME. The approach applied LETHE to description logic-based ontologies to investigate the solutions being computed. The solutions were then compared with those being computed using the other forgetting tools, SCAN and FAME.*

Key words: Forgetting, Ontology, Resolution, Ackermann's Lemma.

1 Introduction

In previous research, a number of practical approaches have been developed to perform forgetting. Generally, these tools fall within two categories: those which use resolution and others which use methods that exploit Ackermann's Lemma. The aim of this research was to compare the solutions being computed between the different methods using real world ontologies. The three tools used to represent the different approaches analyzed in this research are SCAN [2, 5], LETHE [3, 4], and FAME [6].

The first tool, SCAN, computes forgetting solutions for knowledge-bases expressed in first-order logic using a resolution-based approach, namely the SCAN algorithm [2]. Since forgetting in first-order logic is not generally solvable, SCAN is not guaranteed to find a solution. SCAN uses Skolemization to eliminate existential quantifiers, and therefore Skolem terms may appear in SCAN's solutions.

Similarly, LETHE, the second tool, computes its forgetting solution for description logic-based ontologies using a resolution-based method. However, it has been proven that the approach used to develop LETHE will find forgetting solutions for the description logics it can handle [3]. Furthermore, a point that will be useful later is that the forgetting solutions computed using LETHE may involve definer symbols, which are not part of the original vocabulary. This is due to applying flattening techniques to the ontology and sometimes failing to eliminate the extra symbols introduced during the computation [3].

Copyright © 2017 by the paper's authors

In: P. Koopmann, S. Rudolph, R. Schmidt, C. Wernhard (eds.): *SOQE 2017 – Proceedings of the Workshop on Second-Order Quantifier Elimination and Related Topics, Dresden, Germany, December 6–8, 2017*, published at <http://ceur-ws.org>.

Finally, the last tool, FAME, computes its forgetting solutions for description logic-based ontologies using a method based on Ackermann’s Lemma. Similar to LETHE, the method is guaranteed to terminate. However, the method is not forgetting complete and may fail to eliminate some of the symbols [6]. The symbols it fails to eliminate are present in the result computed by the system.

The difference between the tools lie in the expressivity of the logics being supported, as well as the forgetting methodology being used. The aim of this research was to apply one of the forgetting tools, LETHE, to description logic-based ontologies to investigate the solutions computed, and compare them with those computed using SCAN and FAME. The focus was on concept forgetting in the description logic \mathcal{ALC} .

2 Our Tool

For the purposes of the comparison in this research, a tool has been developed to investigate the solutions computed using LETHE, which were then compared with those computed using SCAN and FAME. FAME and LETHE have the advantage of being able to process large ontologies, hence two separate versions of the system were created: one that compares LETHE and SCAN, and another that compares FAME and LETHE using larger ontologies. An effort has been made to maximize the size of the ontologies that SCAN can accept. However, the considered ontologies were still not as large as desired. This is due to boundaries in the static data structures of SCAN.

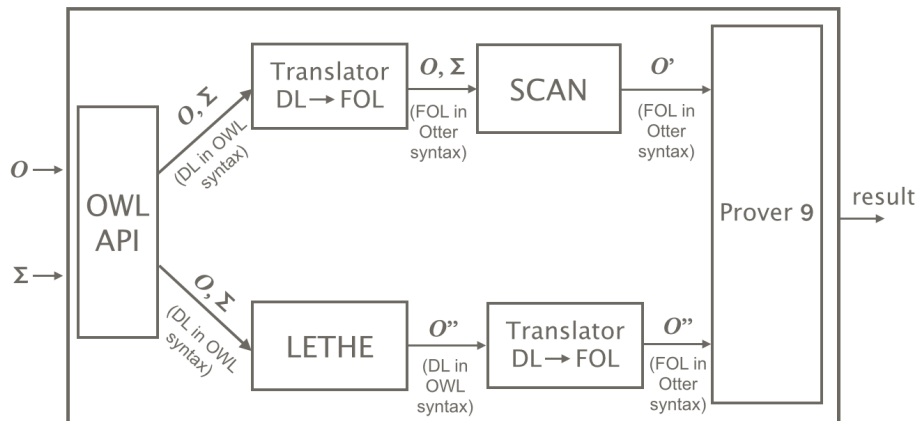


Fig. 1. An abstract design of our tool using SCAN and LETHE.

Figure 1 and Figure 2 give an overview of the components of the system and show how they are used. As can be seen in Figure 1, the tool parses an ontology

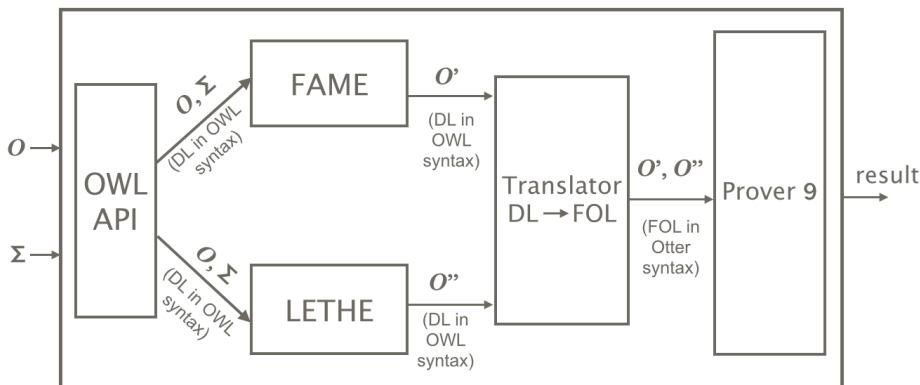


Fig. 2. An abstract design of our tool for comparing LETHE and FAME.

O expressed in description logic in the OWL syntax, using the OWL API, and a set of concept symbols Σ , that is referred to as the forgetting vocabulary. A description logic to first-order logic translator that uses the established relationship between them [1] has been built. The tool uses the translator to translate an ontology expressed in OWL syntax to an equivalent set of first-order formulae expressed in Otter syntax. It is used to translate the input ontology into an equivalent input for SCAN. After each of the two systems have computed its solution, a first-order logic theorem prover called Prover9¹ was used to check for entailments between the results. The translator was used again to translate LETHE's results to first-order logic as Prover9 uses the Otter syntax, too.

The system shown in Figure 2 is another version of the one used in Figure 1, except that the appropriate changes were made to use FAME instead of SCAN.

3 Results of Comparing SCAN and LETHE

In this section, the results of the comparison made between the solutions produced using SCAN and LETHE are presented. Following this, the results of the comparison made between the solutions produced using LETHE and FAME are presented in the next section.

In both comparisons, a set of randomly selected concepts was forgotten. The cardinality of the forgetting vocabulary varied depending on the number of concepts present in the ontology. However, for each ontology, an effort has been made to test and evaluate the forgetting solutions computed, as a result of forgetting sets of concepts of different sizes, ranging from low to high. In ontologies expressed in description logics that are more expressive than description logic \mathcal{ALC} , an \mathcal{ALC} fragment of the ontologies were used. The tests where LETHE

¹ <https://www.cs.unm.edu/~mccune/prover9/>

produces results that contain definer symbols were excluded as they are not part of the original vocabulary.

After testing and evaluating fragments of different ontologies from the OBO Foundry ontology repository², the following results were observed.

1. The solutions computed using SCAN always entail the solutions computed using LETHE.
2. The solutions computed using LETHE always entail the solutions computed using SCAN if no Skolem terms occur in SCAN's solutions. However, if Skolem terms do occur in the solution, the solution computed using LETHE entail the subset of clauses and formulas that do not contain Skolem terms.
3. In some cases, the subset of clauses and formulas in SCAN's solution, that do not contain Skolem terms, is sufficient to entail the forgetting solutions computed using LETHE. The exact reason behind this occurrence raises interesting new questions of research.
4. If SCAN and LETHE both terminate, the approach used to develop SCAN outperforms the approach used to develop LETHE. However, it must be noted that this is the average situation.
5. In some cases the two approaches agree on the difficulty of the forgetting problem. However, in other cases, it was observed that LETHE found some forgetting problems to be difficult, while SCAN found a solution relatively quickly. It was speculated that the reverse occurs as well, but currently there is not sufficient evidence to support it.

4 Results of Comparing LETHE and FAME

Based on the testing and evaluation performed using fragments of ontologies from the Oxford ontology repository³, the following results were observed:

1. If FAME successfully forgets all the symbols in the forgetting vocabulary and produces a solution that is expressed in description logic \mathcal{ALC} , rather than a more expressive description logic such as \mathcal{ALCI} , then the solutions produced by FAME and LETHE entail one another.
2. If FAME successfully forgets all the symbols in the forgetting vocabulary, but produces a solution in a more expressive description logic, namely \mathcal{ALCI} , then the solutions produced by FAME entail those produced by LETHE.
3. If FAME fails to forget a subset of the forgetting vocabulary and LETHE produces a solution that does not contain any unwanted symbols such as definer symbols, then the solutions produced by FAME entail those produced by LETHE. This case captures the strength of resolution as it shows that there exist cases where a forgetting problem can be solved using resolution but not using an Ackermann-based approach. Consequently, this opens the door to consider a hybrid technique that benefits from the speed of FAME and the power of LETHE.

² <http://obofoundry.org>

³ <http://www.cs.ox.ac.uk/isg/ontologies/>

4. One of the differences between FAME and LETHE is that LETHE introduces definer symbols using standard flattening techniques to structurally transform the ontology into what is called normal form, in order to simplify its computations. We found that if LETHE introduces definer symbols and fails to eliminate all of them, FAME fails to forget at least one symbol in the ontology. Consequently, if FAME fails to forget at least one symbol and LETHE fails to eliminate at least one definer symbol, then no entailments can be inferred. This is obvious because each tool produces solutions that contain symbols that are not present in the solutions of the other tool. However, interestingly, one of the flattening techniques used to produce definer symbols in LETHE is inspired by Ackermann's Lemma; instead of using the Lemma to eliminate a symbol, it is used to introduce one. Hence, this raises the question: what is the relationship between the definer symbols that LETHE failed to eliminate and the symbols FAME failed to forget? We expect that these are cases of symbols with cyclic dependencies, which remains to be checked.

5 Concluding Remarks

Parts of the theory were confirmed and verified using a tool that was developed for the purposes of this research. The tool aims to compare the solutions computed using the three forgetting tools: SCAN, LETHE and FAME. It provided insight into the relationship between the three forgetting tools in the absence of a theoretical comparison.

The possibilities of a hybrid system are currently being investigated in an attempt to improve the performance while exploiting the power of resolution. A naive implementation has shown interesting results, particularly, for one of the forgetting problems. This problem took the resolution rules, implemented in LETHE, more than 14 hours to compute a solution. On the other hand, FAME, which exploits Ackermann's Lemma, successfully forgets all the symbols in the forgetting problem, except for one, in two seconds. The hybrid method which first applies FAME and then LETHE to the result computed by FAME successfully forgets all the symbols in the forgetting problem in three seconds. It would be interesting to develop heuristics that select the approach to use for each symbol in the forgetting vocabulary.

Moreover, we are interested in investigating the relationships between the tools and the underlying approaches more deeply. For example, we have not yet found any counterexamples to the following statement: when FAME successfully forgets all the symbols in the forgetting vocabulary, but produces results in \mathcal{ALCI} , we have that the solutions produced by LETHE entail a subset of those produced by FAME, namely the subset of solutions that are expressed in description logic \mathcal{ALC} . Additionally, no cases where FAME successfully forgets all the symbols and LETHE fails to eliminate a definer symbol were found during this research. We would like to investigate if this can theoretically happen. For concept forgetting in \mathcal{ALC} , we expect that this cannot happen. Finally, we would

like to continue investigating the relationship between the approaches used in the various systems but for more expressive description logics.

References

1. F. Baader, D. Calvanese, D. L. McGuinness, D. Nardi, and P. F. Patel-Schneider, editors. *The Description Logic Handbook: Theory, Implementation and Applications*. University of Cambridge Press, Cambridge, UK, 2007.
2. D. M. Gabbay, R. A. Schmidt, and A. Szalas. *Second-Order Quantifier Elimination: Foundations, Computational Aspects and Applications*, volume 12 of *Studies in Logic: Mathematical Logic and Foundations*. College Publications, 2008.
3. P. Koopmann. *Practical Uniform Interpolation for Expressive Description Logics*. PhD thesis, University of Manchester, 2015.
4. P. Koopmann and R. A. Schmidt. Implementation and evaluation of forgetting in \mathcal{ALC} -ontologies. In *Proceedings of the 7th International Workshop on Modular Ontologies co-located with the 12th International Conference on Logic Programming and Non-monotonic Reasoning (LPNMR 2013), Corunna, Spain, September 15, 2013.*, 2013.
5. H. J. Ohlbach, T. Engel, R. A. Schmidt, and D. M. Gabbay. Scan: Home page. <http://www.mettel-prover.org/scan/index.html>. [Online; accessed 29-October-2017].
6. Y. Zhao and R. A. Schmidt. Concept forgetting in \mathcal{ALCOT} -ontologies using an Ackermann approach. In M. Arenas, O. Corcho, E. Simperl, M. Strohmaier, M. d'Aquin, K. Srinivas, P. T. Groth, M. Dumontier, J. Heflin, K. Thirunarayan, and S. Staab, editors, *The Semantic Web, 14th International Semantic Web Conference, ISWC 2015*, volume 9366 of *Lecture Notes in Computer Science*, pages 587–602. Springer, 2015.