Concordia UNIVERSITY

A Resolution Based Framework to Explain Reasoning in Description Logics

Xi Deng and Volker Haarslev and Nematollaah Shiri

Dept. of Computer Science & Software Engineering, Montreal, Quebec, Canada

Introduction

The family of Description Logics (DLs) has played an important role in application areas such as domain modelling and the Semantic Web. However, existing DL reasoners, such as Racer, do not provide users with explanation services; they merely answer "yes" or "no" to a satisfiability and/or inconsistency query with no details. In case of unsatisfiability / inconsistency queries, simply providing such answers is clearly of little help for users to identify sources of unsatisfiability / inconsistency. It is thus crucial to develop explanation services for DL reasoners.

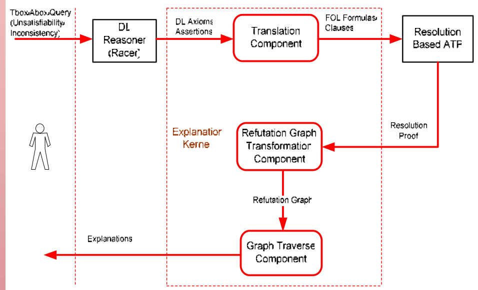
There are several proposals to provide explanations for DL reasoning. The earliest work is [1] which proposes to use inference rules to explain structural subsumption in CLASSIC. This work is extended in [2] by using modified sequent rules to explain subsumption in ALC. In contrast to these approaches, [3] provides algorithms to pinpoint unsatisfiable concepts and related axioms.

Our Proposal

We propose to use resolution proofs to construct explanations for **unsatisfiability and inconsistency queries w.r.t. TBoxes and ABoxes** in ALC. The main idea of our approach is that compared to natural deduction proofs, resolution proofs are usually simpler and provide more homogenous search space. Compared to tableau rules, resolution can deal more easily with ABoxes and global axioms.

Our explanation procedure consists of 3 components. Firstly, if the answer to a concept satisfiability or a TBox or Abox consistency query is "No", the DL reasoner will invoke the explanation module. The original DL axioms and assertions will be translated into first-order logic (FOL) formulas or clauses by the translation component. Secondly, the explanation module will call a resolution based automated theorem prover (ATP) to generate resolution proofs. Finally, the resolution proof is sent back to the explanation module to reconstruct expressions for better human understanding.

During the reconstruction procedure, our approach uses **refutation graphs** as a guidance, where the nodes are literals involved in a resolution proof, and the edges connect and highlight contradictory literals.



The various components in a system prototype.

Explanation Based on Labelled Refutation Graphs

We extend the definition of refutation graphs to labeled refutation graphs, in which a label associates each literal node with its originating DL axioms or FOL formulas.

The main idea of explanation through labeled refutation graphs is to start from literal nodes and traverse the graph. The traversal order is decided by predefined heuristic rules. For example, the axiom A⊆B is more appropriate to be used as an inference rule to explain B is the consequence of A than ¬A is the consequence of ¬B. Therefore, A should be traversed before B is traversed.

The simplified algorithm is as follows:

Step 1: Start from literal nodes and begin the traversal. For unsatisfiability, the unsatisfiable concept can act as the goal or the starting point of the traversal. For inconsistency queries, all the literal nodes involved in the first step of the resolution proof will be put into the start set.

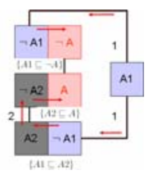
Step 2: When a literal node is being traversed, its label is added to the explanation list.

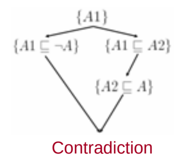
Step 3: After the traversal is completed, each label is translated into an entry in an explanation list consisting of its source axioms in DL or formulas in FOL.

Explaining Unsatisfiable Concepts

Consider the TBox shown on the right. Its corresponding labelled refutation graph is shown below on the left.

1. $A1 \sqsubseteq A2 \sqcap \neg A \sqcap A3$
2. $A2 \sqsubseteq A \sqcap A4$





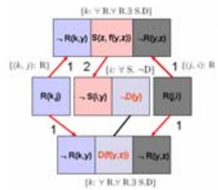
Since A1 is unsatisfiable, the corresponding literal node of A1 is the starting point of the traversal. The traversal order is shown above on the left, with the number on the link denoting the traversal order, the arrow indicating the traversal direction, and the axiom below each clause node indicating its label. The graph on the right shows the labels organized as explanations after the traversal.

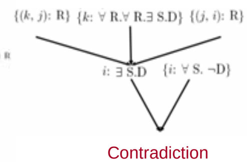
Explaining Inconsistent TBox/ABox

Consider the following more complex ABox:

1. i: $\forall$ S. ¬D
2. (i, j): R
3. (j, i): R
4. (j, k): R
5. (k, j): R
6. k: $\forall$ R.$\forall$ R.$\exists$ S.D

The corresponding labeled refutation graph and explanations are as follows:





We can see that the contradiction is due to the fact that i is an instance of ∀**S.¬D** as a known fact, but at the same time i belongs to ∃**S.D**.

Conclusion

We proposed a resolution proof based approach to explain DL reasoning. As a first step, we have focused on explaining unsatisfiability and inconsistency queries. We have developed the basic algorithms to generate explanations, details of which can be found in [4]. An implementation of the proposed framework is underway.

Open issues:

1. Showing the practical advantage of using resolution proofs by completing the prototype system.
2. Extending the expressivity of the underlying DL language.
3. Explaining satisfiability and subsumption queries.

References

1. McGuinness, D.L., and Borgida, A. 1995. Explaining subsumption in Description Logics. In Proceedings of the 10th International Joint Conference on Artificial Intelligence.
2. Borgida, A; Franconi, E.; Horrocks, I.; McGuinness, D.L.; and Patel-Schneider, P.F. 1999. Explaining ALC subsumption. In 1999 International Workshop on Description Logics.
3. Schlobach, S., and Cornet, R. 2003. Non-standard reasoning services for the debugging of description logic terminologies. In Proceedings of the 18th International Joint Conference on Artificial Intelligence.
4. Deng, X; Haarslev, V.; and Shiri. N. Explanation of Reasoning in Description Logics. To appear. In Proc. of AAAI Int'l Symp. on Explanation-Aware Computing (ExaCT2005).

Acknowledgement

This work was supported in part by Natural Sciences and Engineering Research Council (NSERC) of Canada and by ENCS, Concordia University.