Project Proposal: SMT Instantiations Via GNNs

Jelle Piepenbrock[1,2], Mikoláš Janota[2], Jan Jakubuv[2], and Bartosz Piotrowski[2]

[1] Radboud University Nijmegen, The Netherlands
[2] CIIRC, Czech Technical University in Prague, Czechia

Abstract

Instantiation of quantifiers is a challenging problem within the field of SMT solvers. We will present ongoing work integrating graph neural networks into CVC5's instantiation strategies. The main idea is to predict which terms and combinations of terms will be useful for the solver to instantiate with.

1 Introduction

In satisfiability modulo theories (SMT) solvers, the ground reasoning is handled by highly optimized theory-specific solvers. For example, linear integer arithmetic or boolean vector calculations can be handled by specific routines that exploit domain knowledge.

The non-ground reasoning can be handled via quantifier instantiation. In existing SMT solvers, such as CVC5 [1], there are already several strategies implemented to choose how to instantiate quantifiers. For example, there is enumerative instantiation, in which the solver prefers tuples of terms that contain terms created earlier. There is also E-matching, which uses a specific pattern matching algorithm to choose terms to instantiate with. Another possibility is to make use of the propositional model, like in *model-based quantifier instantiation*.

As the space of possible instantiations is difficult to navigate, a logical step is to use machine learning techniques to learn a heuristic that will determine which instantiations are preferred by the solver.

2 Earlier Work

There has been earlier work on using machine learning to guide the instantiation process in SMT solvers. For example, the SMT solvers VeriT [2, 3] and CVC5 [5] have been extended with machine learning techniques. In the CVC5 case, a gradient boosted tree was used to predict term rankings for each quantifier, building from CVC5's *enumerative instantiation* procedure.

3 Instantiation Process

In Figure 1, we show a general overview of the enumerative instantiation procedure. The basic idea of [5] was to reorder the term rankings based on some features of the terms, so that more promising tuples (i.e. the ones that are judged by the machine learning predictor to contain terms likely to be in proof instantiations), are tried earlier in the solving process.

While the previous work did show some performance improvements, the gradient boosted tree had a very limited feature representation of each term to base its predictions on. We expect that a machine learning predictor that can use information about the entire formula, including previous instantiations made in a particular solving run, as well as the terms that are available, would be able to make better prediction than the predictor with a limited representation. This is why we are working on integrating a graph neural network to predict the term scores.

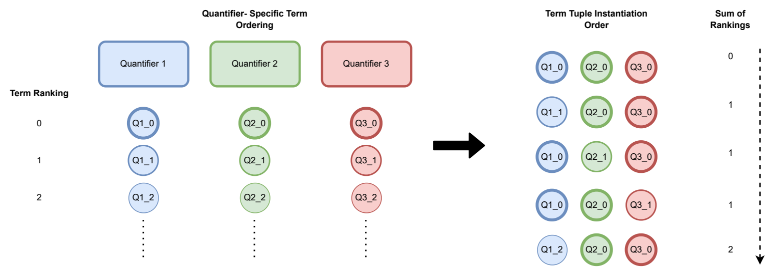

Figure 1: Schematic of enumerative instantiation as implemented in CVC5. The earlier a term appears in the ranking, the more likely it is to end up in a tuple of terms that is actually used in an instantiation step.

4 GNNs

Graph neural networks (GNNs) are a type of neural network that can process any structure that can be described in terms of nodes and an adjacency matrix that details how the nodes are connected. This type of machine learning model has become been used in several works to represent mathematical data [4].

Specifically, we plan to use a simple convolutional neural network to learn representations of variable nodes and term nodes. The representations of these nodes can then be used to compute scalar products of pairs of variable and term vectors to determine which terms should be ranked higher. The higher this product score between a variable and a term vector, the higher in the ranking we put the term.

5 Plans

Currently, the integration of a graph neural network into CVC5 is mostly resolved. We plan to do the following experiments:

1. Train the predictor on proofs extracted from CVC5 runs on the SMTLib benchmarks that contain quantifers.

2. Evaluate the performance within CVC5 and determine how destructive the slowdown from the calls to the predictor is in terms of solver performance.

References

[1] Haniel Barbosa, Clark W. Barrett, Martin Brain, Gereon Kremer, Hanna Lachnitt, Makai Mann, Abdalrhman Mohamed, Mudathir Mohamed, Aina Niemetz, Andres Nötzli, Alex Ozdemir, Mathias Preiner, Andrew Reynolds, Ying Sheng, Cesare Tinelli, and Yoni Zohar. "cvc5: A Versatile and Industrial-Strength SMT Solver". In: *Tools and Algorithms for the Construction and Analysis of Systems - 28th International Conference, TACAS 2022, Held as Part of the European Joint Conferences on Theory and Practice of Software, ETAPS 2022, Munich, Germany, April 2-7, 2022, Proceedings, Part I.* Ed. by Dana Fisman and Grigore Rosu. Vol. 13243. Lecture Notes in Computer Science. Springer, 2022, pp. 415–442.

DOI: 10.1007/978-3-030-99524-9_24. URL: https://doi.org/10.1007/978-3-030-99524-9_24.

[2] Jasmin Christian Blanchette, Daniel El Ouraoui, Pascal Fontaine, and Cezary Kaliszyk. "Machine Learning for Instance Selection in SMT Solving". In: *AITP 2019 - 4th Conference on Artificial Intelligence and Theorem Proving*. Obergurgl, Austria, Apr. 2019. URL: https://hal.archives-ouvertes.fr/hal-02381430.

[3] Daniel El Ouraoui. "Méthodes pour le raisonnement d'ordre supérieur dans SMT, Chapter 5." PhD thesis. Université de Lorraine, 2021.

[4] Zarathustra A Goertzel, Jan Jakubv, Cezary Kaliszyk, Miroslav Olšák, Jelle Piepenbrock, and Josef Urban. "The Isabelle ENIGMA". In: *13th International Conference on Interactive Theorem Proving*. 2022.

[5] Mikolás Janota, Jelle Piepenbrock, and Bartosz Piotrowski. "Towards Learning Quantifier Instantiation in SMT". In: *25th International Conference on Theory and Applications of Satisfiability Testing, SAT 2022, August 2-5, 2022, Haifa, Israel*. Ed. by Kuldeep S. Meel and Ofer Strichman. Vol. 236. LIPIcs. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2022, 7:1–7:18. DOI: 10.4230/LIPIcs.SAT.2022.7. URL: https://doi.org/10.4230/LIPIcs.SAT.2022.7.