Scalop at ICCMA'25

Jean-Marie Lagniez
CRIL
Univ. Artois & CNRS
Lens, France
lagniez@cril.fr

Emmanuel Lonca
CRIL
Univ. Artois & CNRS
Lens, France
lagniez@cril.fr

Jean-Guy Mailly
IRIT
Université Toulouse Capitole
Toulouse, France
jean-guy.mailly@irit.fr

***Abstract*—SCALOP takes over from CRUSTABRI, which was itself a rewriting of COQUIAAS. It participates in the same tracks as CRUSTABRI was involved in, but also in the heuristics track and, without restriction, in the ABA track. It can use IPASIR dynamic libraries, which allows it to participate in the fixed-SAT solver comparison.**

***Index Terms*—argumentation, assumption, sat, solver.**

I. INTRODUCTION

SCALOP takes over from CRUSTABRI [1], which was itself a rewriting of COQUIAAS [2]. It participates in the same tracks that CRUSTABRI was involved in. Regarding the main track, it remains quite similar to what our previous solver proposed. Some shortcuts were found to reduce the number of calls to the SAT solvers for the DS-PR and SE-ID problems. While these changes improve the solver's performance on toy examples, it is unclear if they will help with competition benchmarks.

Regarding the dynamic track, nothing changed except for bug fixes that prevented the solver from being used on some MacOS and Windows machines.

Unlike its predecessor, Scalop enters the heuristics and the ABA tracks (without limitations). It is also designed to use any SAT solver compatible with the IPASIR library, allowing it to participate in the fixed-SAT solver comparison.

II. THE ABA TRACK

The 2023 edition of the competition established ASPFORABA [3] (and, to a lesser extent, ACBAR [4]) as the clear winners in all categories. CRUSTABRI was not sufficiently prepared and suffered from limitations and performance issues. The most significant addition to SCALOP is its ABA solver, which employs a different approach than ASPFORABA and ACBAR to solve problems.

ASPFORABA and ACBAR translate the input problems into an intermediate problem, that is then translated into a CNF formula used to feed a SAT solver. However, this translation can produce a large CNF formula, which makes it difficult for the SAT solver to determine its satisfiability. Concerning ASPFORABA, the weak point (which did not prevent it from beating its opponents handily in the last competition) could be the grounding step of the ASP solver. Regarding ACBAR, its preprocessing step, which computes a logically equivalent but acyclic framework, may produce quadratic growth of the problem.

SCALOP proceeds in a lazy manner using a counterexample-guided abstraction refinement (CEGAR) algorithm. The algorithm begins with a CNF formula that encodes a superset of the extensions. This formula has the advantage of being linear in size with respect to the number of atoms and rules. When the SAT solver returns a solution, we verify its correctness. If it is incorrect, we add clauses such that a set of models that do not encode the correct extensions (including the one returned by the solver) are falsified in the formula. This process continues until a model corresponding to an extension is returned, allowing us to learn the correct CNF step by step.

III. THE HEURISTICS TRACK

SCALOP enters the heuristics track as a dummy shell script that invokes the regular solver with a timeout. If it fails to find a solution, it returns the default value for the current combination of semantics and query.

IV. INTERFACE WITH IPASIR LIBRARIES

SCALOP is shipped with the same version of CADICAL [5] that was included in CRUSTABRI. However, this SAT engine can be replaced at execution time by providing to SCALOP a dynamic library of a SAT solver compatible with the IPASIR interface.

REFERENCES

[1] J. Lagniez, E. Lonca, and J. Mailly, "A sat-based approach for argumentation dynamics," in *Proceedings of the 23rd International Conference on Autonomous Agents and Multiagent Systems, AAMAS 2024, Auckland, New Zealand, May 6-10, 2024*, M. Dastani, J. S. Sichman, N. Alechina, and V. Dignum, Eds. International Foundation for Autonomous Agents and Multiagent Systems / ACM, 2024, pp. 2351–2353. [Online]. Available: https://dl.acm.org/doi/10.5555/3635637.3663157

[2] ——, "Coquiaas: A constraint-based quick abstract argumentation solver," in *27th IEEE International Conference on Tools with Artificial Intelligence, ICTAI 2015, Vietri sul Mare, Italy, November 9-11, 2015*. IEEE Computer Society, 2015, pp. 928–935. [Online]. Available: https://doi.org/10.1109/ICTAI.2015.134

[3] T. Lehtonen, J. P. Wallner, and M. Järvisalo, "Declarative algorithms and complexity results for assumption-based argumentation," *J. Artif. Intell. Res.*, vol. 71, pp. 265–318, 2021. [Online]. Available: https://doi.org/10.1613/jair.1.12479

[4] T. Lehtonen, A. Rapberger, M. Ulbricht, and J. P. Wallner, "Argumentation frameworks induced by assumption-based argumentation: Relating size and complexity," in *International Conference on Principles of Knowledge Representation and Reasoning*. International Joint Conferences on Artificial Intelligence, 2023, pp. 440–450.

[5] A. Fleury and M. Heisinger, "Cadical, kissat, paracooba, plingeling and treengeling entering the sat competition 2020," *Sat Competition*, vol. 2020, p. 50, 2020.