κ-Solutions: A Solver for the Dynamic Track of ICCMA 2023

Christian Pasero
Johannes P. Wallner
Institute of Software Technology
Graz University of Technology
Graz, Austria

***Abstract*—In this system description we present κ-Solutions, a solver submission for the dynamic track of the fifth International Competition on Computational Models of Argumentation (ICCMA). Our approach is to compute k-many witnesses to problems posed in the dynamic track using the state-of-the-art SAT solver Z3. For instance, for credulous acceptance under admissibility, we compute (up to) k-many admissible sets containing a queried argument. After subsequent changes, we only re-compute if witnesses are not preserved under the modifications. Our solver can answer credulous acceptance under complete and stable semantics, and skeptical acceptance under stable semantics.**

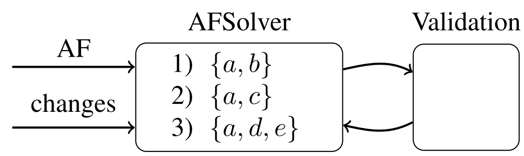

Fig. 1. System architecture of κ-Solutions

I. Introduction

In this system description we introduce κ-Solutions, a solver submission to the dynamic track of the fifth International Competition on Computational Models of Argumentation (ICCMA), which follows successful previous installments [1]–[4]. Our solver is based on computing several witnesses for answering the tasks of the dynamic track, and only re-computing if witnesses cease to remain witnesses after modifications. Following successful approach using Boolean Satisfiability (SAT) solvers for computing AF reasoning tasks [5], e.g., by the recent solver μ-toksia [6], we make use of Boolean encodings of argumentation semantics [7] and the Z3 SAT solver [8].

In the dynamic track of this edition of ICCMA, the tasks are to perform argumentative reasoning on a dynamically adapting argumentation framework (AF) [9]. An AF represents abstract arguments and attacks between these arguments as directed edges.

II. Background

We briefly recall argumentation semantics focused on in this work.

Given an AF $F = (A, R)$, with A a set of (abstract) arguments and $R \subseteq A \times A$ an attack relation, a set of arguments $S \subseteq A$ is called conflict-free in F if $\nexists a, b \in S$ with $(a, b) \in R$. An argument $a \in A$ attacks $b \in A$ in F if $(a, b) \in R$. A subset $S \subseteq A$ of arguments attacks an argument $a \in A$ in F if there is some $b \in S$ with $(b, a) \in R$. The set S defends an argument $c \in A$ if for each b with $(b, c) \in R$ we find an $a \in S$ with $(a, b) \in R$.

A conflict-free set of arguments S in F is called admissible in F if S defends each $a \in S$. An admissible set is called a complete extension in F if each argument defended by S is in S. Moreover, a conflict-free set S is a stable extension in F if each $a \in A \setminus S$ is attacked by S. For reasons of uniformity, we call admissible sets also admissible extensions.

Two reasoning tasks are credulous and skeptical acceptance of arguments w.r.t. an argumentation semantics $\sigma \in \{adm, com, stb\}$ with adm, com, and stb standing in for admissible, complete, or stable, respectively. Given an AF $F = (A, R)$ and an argument $a \in A$ it holds that a is credulously accepted under σ if there is a σ-extension S in F with $a \in S$. Skeptical acceptance under σ holds in F if for each σ-extension S we find that $a \in S$.

In the dynamic track the AF in question dynamically changes: arguments and attacks may be added or removed, and after each modification a query regarding credulous (or skeptical) acceptance of an argument may be performed.

III. System Architecture

Our approach for solving the dynamic reasoning tasks are, in brief terms, to compute k-many witnesses for credulous or skeptical (non-)acceptance. After each modification, we perform (direct) checks whether the stored witnesses are not witnesses anymore, and only re-compute when faced with no more witnesses.

Our solver, we call κ-Solutions, is outlined in Figure 1. The solver consists of two main components, implemented in a python library: the AFSolver and the Validation components. The AFSolver provides methods for parsing an AF and for changing arguments and attacks. When given a query for an argument a, e.g., credulous acceptance under admissibility, the AFSolver computes k-many admissible sets containing a, with $k \geq 1$. The positive integer k is pre-set to a constant. After initial testing, we set it in our current submission to $k = 3$. If there are no admissible sets containing a, AFSolver returns that the argument a is not credulously accepted under admissibility. If witnesses of credulous acceptance for a under admissibility can be found, AFSolver computes up to k-many, and less than k is there are less admissible sets containing a (then the maximum number of such sets are computed.

When the AF is modified, e.g., via adding attacks, the Validation component checks whether the current (up to k-many) witnesses still represent σ-extensions containing a. If not, the witness is discarded. If no witnesses remain, the AFSolver computes again k-many solutions. If one already-computed witness is still valid (i.e., still an admissible set containing a for admissibility and credulous acceptance), our solver directly answers the query.

For skeptical acceptance under stable semantics, a dual approach is used: we compute stable extensions not containing a queried argument.

For solving the underlying tasks of finding σ-extensions (that do or do not contain a queried argument) under a semantics σ, we make use of the Boolean encodings of argumentation semantics by Besnard and Doutre [7]. For the SAT solver, we make use of the well-performing Z3 solver [8].

For the Validation component, we also make use of Z3 to (in-)validate σ-extensions: we construct a Boolean formula that is satisfiable iff a given set of arguments is a σ-extension (for the current AF). More concretely, in the Validation component we construct a different Boolean formula, which sets its variables as defined by the given witness, enabling direct validation by the SAT solver.

IV. Supported Reasoning Tasks

Our solver κ-Solutions supports within the dynamic track the following reasoning tasks:

- credulous acceptance under complete extensions (and admissible sets) and
- credulous and skeptical acceptance under stable semantics.

V. Conclusions

We presented our solver κ-Solutions for the fifth editions of ICCMA. Our solver aims to save computation time after modifications of the original AF by computing several σ-extensions, and only discarding them if they are invalidated after a change.

Our solver κ-Solutions is available under the MIT license at

https://github.com/p4s3r0/argument_solver.

Acknowledgements

This research was supported by the Austrian Science Fund (FWF) P35632.

References

[1] M. Thimm and S. Villata, "The first international competition on computational models of argumentation: Results and analysis," *Artif. Intell.*, vol. 252, pp. 267–294, 2017.

[2] S. A. Gaggl, T. Linsbichler, M. Maratea, and S. Woltran, "Summary report of the second international competition on computational models of argumentation," *AI Mag.*, vol. 39, no. 4, pp. 77–79, 2018.

[3] S. Bistarelli, L. Kotthoff, F. Santini, and C. Taticchi, "Summary report for the third international competition on computational models of argumentation," *AI Mag.*, vol. 42, no. 3, pp. 70–73, 2021.

[4] J. Lagniez, E. Lonca, J. Mailly, and J. Rossit, "Introducing the fourth international competition on computational models of argumentation," in *Proc. SAFA*, ser. CEUR Workshop Proceedings, S. A. Gaggl, M. Thimm, and M. Vallati, Eds., vol. 2672. CEUR-WS.org, 2020, pp. 80–85.

[5] F. Cerutti, S. A. Gaggl, M. Thimm, and J. P. Wallner, "Foundations of implementations for formal argumentation," in *Handbook of Formal Argumentation*, P. Baroni, D. Gabbay, M. Giacomin, and L. van der Torre, Eds. College Publications, 2018, ch. 15, pp. 688–767.

[6] A. Niskanen and M. Järvisalo, "μ-toksia: An efficient abstract argumentation reasoner," in *Proc. KR*, D. Calvanese, E. Erdem, and M. Thielscher, Eds. IJCAI, 2020, pp. 800–804.

[7] P. Besnard and S. Doutre, "Checking the acceptability of a set of arguments," in *Proc. NMR*, J. P. Delgrande and T. Schaub, Eds., 2004, pp. 59–64.

[8] L. M. de Moura and N. S. Bjørner, "Z3: an efficient SMT solver," in *Proc. TACAS*, ser. Lecture Notes in Computer Science, C. R. Ramakrishnan and J. Rehof, Eds., vol. 4963. Springer, 2008, pp. 337–340.

[9] P. M. Dung, "On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games," *Artif. Intell.*, vol. 77, no. 2, pp. 321–358, 1995.