FUDGE: A LIGHT-WEIGHT SOLVER FOR ABSTRACT ARGUMENTATION BASED ON SAT REDUCTIONS

Matthias Thimm
Institute for Web Science and Technologies
University of Koblenz-Landau, Germany
thimm@uni-koblenz.de

Federico Cerutti
Department of Information Engineering
University of Brescia, Italy
federico.cerutti@unibs.it

Mauro Vallati
School of Computing and Engineering
University of Huddersfield, United Kingdom
m.vallati@hud.ac.uk

ABSTRACT

We present FUDGE, an abstract argumentation solver that tightly integrates satisfiability solving technology to solve a series of abstract argumentation problems. While most of the encodings used by FUDGE derive from standard translation approaches, FUDGE makes use of completely novel encodings to solve the skeptical reasoning problem wrt. preferred semantics and problems wrt. ideal semantics.

1 Introduction

An *abstract argumentation framework* AF is a tuple $\mathsf{AF} = (\mathsf{A}, \mathsf{R})$ where A is a (finite) set of arguments and R is a relation $\mathsf{R} \subseteq \mathsf{A} \times \mathsf{A}$ [1]. For two arguments $a, b \in \mathsf{A}$ the relation $a\mathsf{R}b$ means that argument a attacks argument b. For a set $S \subseteq \mathsf{A}$ we define

$$S^+ = \{a \in \mathsf{A} \mid \exists b \in S, b\mathsf{R}a\}$$
$$S^- = \{a \in \mathsf{A} \mid \exists b \in S, a\mathsf{R}b\}$$

We say that a set $S \subseteq \mathsf{A}$ is *conflict-free* if for all $a, b \in S$ it is not the case that $a\mathsf{R}b$. A set S *defends* an argument $b \in \mathsf{A}$ if for all a with $a\mathsf{R}b$ there is $c \in S$ with $c\mathsf{R}a$. A conflict-free set S is called *admissible* if S defends all $a \in S$.

Different semantics [2] can be phrased by imposing constraints on admissible sets. In particular, set E

- is a *complete* (CO) extension iff it is admissible and for all $a \in \mathsf{A}$, if E defends a then $a \in E$,
- is a *grounded* (GR) extension iff it is complete and minimal,
- is a *stable* (ST) extension iff it is conflict-free and $E \cup E^+ = \mathsf{A}$,
- is a *preferred* (PR) extension iff it is admissible and maximal.
- is an *ideal* (ID) extension iff $E \subseteq E'$ for each preferred extension E' and E is maximal.

All statements on minimality/maximality are meant to be with respect to set inclusion.

Given an abstract argumentation framework $\mathsf{AF} = (\mathsf{A}, \mathsf{R})$ and a semantics $\sigma \in \{CO, GR, ST, PR, ID\}$ we are interested in the following computational problems[1]:

SE-σ: For a given abstract argumentation framework AF, compute some σ-extension.

[1] http://argumentationcompetition.org/2021/SolverRequirements.pdf

CE-σ: For a given abstract argumentation framework AF, determine the number of all σ-extensions.

DC-σ: For a given abstract argumentation framework AF and an argument a, decide whether a is in at least one σ-extension of AF.

DS-σ: For a given abstract argumentation framework AF and an argument a, decide whether a is in all σ-extensions of AF.

Note that DC-σ and DS-σ are equivalent for $\sigma \in \{GR, ID\}$ as those extensions are uniquely defined [2]. For these, we will only consider DS-σ.

The FUDGE solver supports solving the above-mentioned computational problems wrt. all $\sigma \in \{CO, GR, ST, PR, ID\}$. In the remainder of this system description, we give a brief overview on the architecture of FUDGE (Section 2) and conclude in Section 3.

2 Architecture

FUDGE follows the standard reduction-based approach to solve the above-mentioned reasoning problems [3, 4] with the target formalism being the satisfiability problem SAT [5]. For example, given the problem SE-ST and an input argumentation framework $\mathsf{AF} = (\mathsf{A}, \mathsf{R})$, first, for each argument $a \in \mathsf{A}$, we create a propositional variable in_a, with the meaning that in_a is true in a satisfying assignment iff the argument a is in the stable extension to be found. Then conflict-freeness can be modelled by the formula

$$\Phi_1(\mathsf{AF}) = \bigwedge_{(a,b)\in\mathsf{R}} \neg(\mathrm{in}_a \wedge \mathrm{in}_b)$$

while the constraint that all arguments not included in the extension must be attacked can be modelled by

$$\Phi_2(\mathsf{AF}) = \bigwedge_{a\in\mathsf{A}} (\neg\mathrm{in}_a \Leftrightarrow \bigvee_{(b,a)\in R} \mathrm{in}_b)$$

Then the formula $\Phi_1(\mathsf{AF}) \wedge \Phi_2(\mathsf{AF})$ is satisfiable iff AF has a stable extension and a stable extension can be easily extracted from a satisfying assignment of $\Phi_1(\mathsf{AF}) \wedge \Phi_2(\mathsf{AF})$. All reasoning problems on the first level of the polynomial hierarchy [6] can be solved in a similar manner and their corresponding counting problems can be realised by iterative satisfiability tests to enumerate all extensions.

Particularly challenging problems are those wrt. preferred semantics as, in particular, DS-PR is Π_2^P-complete [6]. To solve that problem, we use the approach recently presented in [7]. This approach relies on the following observation[2]:

Theorem 1 ([7])**.** *$a \in \mathsf{A}$ is skeptically accepted wrt. preferred semantics iff*

1. *there is an admissible set S with $a \in S$ and*
2. *for every admissible set S with $a \in S$ and every admissible set S' with $S'\mathsf{R}S$, there is an admissible set S'' with $S' \cup \{a\} \subseteq S''$.*

The above theorem states that we can decide skeptical acceptance wrt. preferred semantics by considering only those admissible sets that attack an admissible set containing the argument in question. As an admissibility check can be solved by a satisfiability check, similarly as above, the above insight leads to an algorithm that can solve DS-PR without actually computing preferred extensions. The algorithm is presented in detail in [7] and experiments confirm a significant performance improvement compared to previous encoding approaches.

Coming to ideal semantics, it is worth recalling [8, Theorem 3.3].

Theorem 2 ([8, Theorem 3.3])**.** *An admissible set of arguments S is ideal iff for each argument a attacking S there exists no admissible set of arguments containing a.*

The interesting aspect of [8, Theorem 3.3] is that ideal semantics, although defined based on skeptical acceptance wrt. preferred semantics, does not rely on that notion. Moreover, in [7] we also prove that starting from the set of arguments which are not attacked by an admissible set, its largest admissible set is the ideal extension. We can therefore tweak the machinery we created for computing skeptical acceptance wrt. preferred semantics to compute the ideal extension too. The complete algorithm is presented in [7].

[2]Define $S'\mathsf{R}S$ iff there is $a \in S'$ and $b \in S$ with $(a, b) \in \mathsf{R}$.

FUDGE is written in C++ and uses the satisfiability solver CaDiCaL 1.3.1[3] via its C++ API. CaDiCaL supports *incremental* solving which provides a significant performance boost when successive satisfiability checks have to be made, in particular for counting problems.

3 Summary

We presented FUDGE a reduction-based solver for various problems in abstract argumentation. FUDGE leverages on a mix of standard and novel SAT encodings to solve reasoning problems, with the aim of avoiding the costly maximisation step that is characteristic of some of the abstract argumentation problems. The source code of FUDGE is available at `http://taas.tweetyproject.org`.

References

[1] Phan Minh Dung. On the Acceptability of Arguments and its Fundamental Role in Nonmonotonic Reasoning, Logic Programming and n-Person Games. *Artificial Intelligence*, 77(2):321–358, 1995.

[2] Pietro Baroni, Martin Caminada, and Massimiliano Giacomin. An introduction to argumentation semantics. *The Knowledge Engineering Review*, 26(4):365–410, 2011.

[3] Philippe Besnard, Sylvie Doutre, and Andreas Herzig. Encoding argument graphs in logic. In *International Conference on Information Processing and Management of Uncertainty in Knowledge-based Systems - IPMU 2014*, 2014.

[4] Federico Cerutti, Sarah A. Gaggl, Matthias Thimm, and Johannes P. Wallner. Foundations of implementations for formal argumentation. In Pietro Baroni, Dov Gabbay, Massimiliano Giacomin, and Leendert van der Torre, editors, *Handbook of Formal Argumentation*, chapter 15. College Publications, February 2018.

[5] Armin Biere, Marijn Heule, Hans van Maaren, and Toby Walsh, editors. *Handbook of Satisfiability*, volume 185 of *Frontiers in Artificial Intelligence and Applications*. IOS Press, 2009.

[6] Wolfgang Dvořák and Paul E. Dunne. Computational problems in formal argumentation and their complexity. In Pietro Baroni, Dov Gabbay, Massimiliano Giacomin, and Leendert van der Torre, editors, *Handbook of Formal Argumentation*, chapter 14. College Publications, February 2018.

[7] Matthias Thimm, Federico Cerutti, and Mauro Vallati. Skeptical reasoning with preferred semantics in abstract argumentation without computing preferred extensions. In *Proceedings of the 30th International Joint Conference on Artificial Intelligence (IJCAI'21)*, 2021.

[8] P. M. Dung, P. Mancarella, and F. Toni. Computing ideal sceptical argumentation. *Artificial Intelligence*, 171(10):642–674, 2007.

[3] `http://fmv.jku.at/cadical/`