FARGO-LIMITED V1.1.1

Matthias Thimm
Artificial Intelligence Group
University of Hagen
Germany
matthias.thimm@fernuni-hagen.de

***Abstract*—We present FARGO-LIMITED V1.1.1, a solver for approximate reasoning for various tasks in abstract argumentation. The solver relies on a DPLL-approach to exhaustive search for extensions, but is constrained in the search space by a bounded depth.**

I. 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 $\mathsf{R} \subseteq \mathsf{A} \times \mathsf{A}$ [3]. 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 [1] can be phrased by imposing constraints on admissible sets. In particular, a 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 a *semi-stable* (SST) extension iff it is complete and $E \cup E^+$ is maximal.
- is a *stage* (STG) extension iff it is conflict-free and $E \cup E^+$ is 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 $\sigma \in \{CO, GR, ST, PR, SST, STG, ID\}$ we are interested in the following computational problems [4], [5]:

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

DS-σ : For a given 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 [1]. For these, we will only consider DS-σ.

The FARGO-LIMITED V1.1.1 solver supports solving the above-mentioned computational problems wrt. to all $\sigma \in \{CO, GR, ST, PR, SST, STG, ID\}$. In the remainder of this system description, we give a brief overview on the architecture of FARGO-LIMITED V1.1.1 (Section II) and conclude in Section III.

II. ARCHITECTURE

The core of the solver lies in an algorithm for approximately determining whether an argument is contained in an admissible set.[1] For $\sigma \in \{CO, ST, PR, SST, STG, ID\}$ we approximate the answer to a DC-σ query by a positive answer to such a test. For DS-σ, we additionally check whether any attacker of the query argument is (approximately) in an admissible set. If the query argument is (approximately) contained in an admissible set and no attacker of the query argument is (approximately) contained in an admissible set, the answer to DS-σ is positive.

The general algorithm for checking whether a given argument a is contained in an admissible set is given in Algorithm 1. This algorithm is a variant of the standard DPLL-search algorithm [2], where the search direction is influenced by the attack directions. Moreover, the search is bounded by a given maximum depth $n \in \mathbb{N} \cup \{\infty\}$. More precisely, Algorithm 1 is initially called via $\mathsf{admSuperSet}(\mathsf{AF}, \{a\}, n)$. If $S = \{a\}$ is already admissible, we terminate with a positive answer in line 2. As long as the maximum search depth is not reached (lines 3–4), we iterate over all arguments b that attack the current set S and are not defended against (line 6). If there is no possible defender c that can be added to S without violating conflict-freeness, we terminate with a negative answer (lines 6–7). Otherwise, we recursively call the algorithm again with the defender c added to S and the adapted maximum search depth (lines 9–10). Note that the algorithm is complete if the maximum search depth is unbounded, i. e., iff $n = \infty$. If the search depth n is finite, it may happen that the answer is FALSE although a is contained in an admissible set (which could not be found due to the limited search depth). However, if the algorithm's answer is TRUE, this is always

[1] Exceptions are problems DC-GR, DS-GR, DS-CO, which are directly solved by an algorithm running in polynomial time.

the correct answer, since an admissible set has been found. FARGO-LIMITED V1.1.1 is written in C++ and relies on no specific libraries other than the C++ standard libraries.

Algorithm 1 (Approximately) verifying whether a given subset can be extended to an admissible set

Input: $\mathsf{AF} = (\mathsf{A}, \mathsf{R})$, $S \subseteq \mathsf{A}$, $n \in \mathbb{N} \cup \{\infty\}$
Output: TRUE if there is admissible S' with $S \subseteq S'$.

admSuperSet(AF,S,n)
if S is admissible **then**
    return TRUE
if $n \leq 0$ **then**
    return FALSE
for $b \in S^- \setminus S^+$ **do**
    if $b^- \setminus (S^- \cup S^+) = \emptyset$ **then**
        return FALSE
    for $c \in b^- \setminus (S^- \cup S^+)$ **do**
        if admSuperSet($\mathsf{AF}, S \cup \{c\}, n-1$) **then**
            return TRUE
return FALSE

III. SUMMARY

We presented FARGO-LIMITED V1.1.1, an approximate solver for various problems in abstract argumentation. The solver relies on a variant of the DPLL-algorithm for searching for admissible sets and includes a maximum search depth. The source code of FARGO-LIMITED V1.1.1 is available at https://github.com/aig-hagen/taas-fargo.

REFERENCES

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

[2] 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.

[3] 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.

[4] 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.

[5] Matthias Thimm and Serena Villata. The first international competition on computational models of argumentation: Results and analysis. *Artificial Intelligence*, 252:267–294, August 2017.