

# ZJU-ARG: A Decomposition-Based Solver for Abstract Argumentation

Qianle Guo<sup>1</sup> and Beishui Liao<sup>1,2</sup>

<sup>1</sup> Center for the Study of Language and Cognition,  
Zhejiang University, Hangzhou 310028, P.R.China

<sup>2</sup> University of Luxembourg, Luxembourg

**Abstract.** This paper gives the system description of a solver for abstract argumentation ZJU-ARG submitted to the First International Competition on Computational Models of Argumentation (ICCMA'15). It adopts a decomposition-based approach to enumerate preferred extensions (labellings) of an abstract argumentation framework, and is implemented in Java. The efficiency of this solver highly depends on the topologies of argumentation frameworks.

## 1 System architecture

The solver ZJU-ARG is based on our previous work for efficient computation of argumentation semantics [1–6]. It enumerates all preferred labellings (or preferred extensions), and the grounded labelling (or the grounded extension) of an argumentation framework (AF). Given an AF, its extension(s) is(are) obtained by means of computing its labelling(s).

While the grounded labelling of an AF is computed directly by Modgil and Caminada's algorithm [7], we use a decomposition-based approach originally presented in [1], to enumerate all preferred labellings:

Given an AF, it is preprocessed by differentiating accepted/rejected arguments and undecided arguments under grounded semantics, by means of computing its grounded labelling. Then, a modified framework (denoted as AF') only containing the undecided arguments is decomposed into a set of sub-frameworks by exploiting its strongly connected components (SCCs). Here, sub-frameworks can be unconditioned and/or conditioned [6]. These sub-frameworks are organized into several layers conforming to the partial order of the SCCs of the AF'. Then, the preferred labellings of the AF' are computed and combined incrementally, from the lowest layer in which each sub-framework is not restricted by other sub-frameworks, to the highest layer in which each sub-framework is at most restricted by the sub-frameworks located in the lower layers. In this step, the algorithms for computing the preferred labellings of each sub-framework are either Modgil and Caminada's algorithm [7] or a revised version of this algorithm. Finally, each preferred labelling of AF' is revised by adding the accepted and rejected arguments identified in the preprocessing step, to form a preferred labelling of the AF. Readers are referred to [5–7] for detailed notions, algorithms and empirical results of this approach.

Compared to the SCC-Recursiveness schema [8], an important characteristic of ZJU-ARG is that the concept of modularity of argumentation (in terms of sub-frameworks) is proposed and exploited. For further development of this notion, please refer to [9].

Since ZJU-ARG solver adopts the divide and conquer strategy, but without using more efficient algorithms to compute the semantics of each sub-framework, its efficiency highly depends on the topologies of argumentation frameworks.

The source code of ZJU-ARG as well as the instructions on how to use it are available at <http://mypage.zju.edu.cn/en/beishui/685664.html>.

## 2 Design choices and lessons learned

The first version of the solver was originally presented in [4–6], without considering the efficiency problem for competition. In this version, we add some additional components to meet the rules of the competition, but without considering the improvement of the efficiency of the solver. Some considerations to improve the solver are as follows.

First, since the efficiency of this solver highly depends on the topologies of argumentation frameworks, we will improve the solver in the following ways:

- Use the decomposition-based approach recursively.
- Propose an approach to decompose the AFs that are not sparse, by using some existing theories (for instance argumentation multipoles [9]).

Second, the solver could be made more efficient by replacing Modgil and Caminadas algorithms[7] for computing the labellings of each sub-framework with some more efficient algorithms.

Third, the current version of the solver is basically oriented to computing the preferred labellings of an AF. In our future work, we will extend it to be applicable under some other important semantics, such as stable and ideal, etc.

## Acknowledgment

The research reported in this paper was partially supported by the National Science Foundation of China (No.61175058, No.61203324) and Zhejiang Provincial Natural Science Foundation of China (No. LY14F030014).

## References

1. Liao, B., Jin, L., Koons, R.: Dynamics of argumentation systems: A basic theory. In: the 17th International Conference on Logic for Programming, Artificial Intelligence and Reasoning, Yogyakarta, Indonesia (2010)
2. Liao, B., Jin, L., Koons, R.C.: Dynamics of argumentation systems: A division-based method. *Artificial Intelligence* **175**(11) (2011) 1790–1814
3. Liao, B., Huang, H.: Partial semantics of argumentation: Basic properties and empirical results. *Journal of Logic and Computation* **23**(3) (2013) 541–562

4. Liao, B.: Toward incremental computation of argumentation semantics: A decomposition-based approach. *Annals of Mathematics and Artificial Intelligence* **67** (2013) 319–358
5. Liao, B., Lei, L., Dai, J.: Computing preferred labellings by exploiting sccs and most sceptically rejected arguments. In: *Theory and Applications of Formal Argumentation*. Volume 8306 of *Lecture Notes in Computer Science.*, Springer (2014) 194–208
6. Liao, B.: *Efficient Computation of Argumentation Semantics*. Elsevier (2014)
7. Modgil, S., Caminada, M.: Proof theories and algorithms for abstract argumentation frameworks. In: Rahwan, I., and Simari G.R., eds., *Argumentation in Artificial Intelligence*, Springer (2009) 105–129
8. Baroni, P., Giacomin, M., Guida, G.: Scc-recursiveness: A general schema for argumentation semantics. *Artificial Intelligence* **168**(1-2) (2005) 165–210
9. Baroni, P., Boella, G., Cerutti, F., Giacomin, M., van der Torre, L., Villata, S.: On the input/output behavior of argumentation frameworks. *Artificial Intelligence* **217** (2014) 144–197