Computational Models of Argument
F. Toni et al. (Eds.)

doi:10.3233/FAIA220174

probo2: A Benchmark Framework for Argumentation Solvers

Jonas KLEIN and Matthias THIMM
Artificial Intelligence Group, University of Hagen, Germany

Abstract. We introduce `probo2`, an end-to-end benchmark framework for abstract argumentation solvers. It offers evaluation capabilities and analysis features for a wide range of computational problems and is easily customizable.

Keywords. abstract argumentation, solver, benchmark, evaluation

1. Introduction

Approaches to formal argumentation [1,7] are a significant part of active research in Artificial Intelligence. Especially, solving reasoning problems in Dung's [5] abstract argumentation framework is fundamental for many argumentation systems. Solving such computational complex reasoning problems requires efficient algorithms. In recent years, there has been an increased effort to develop such algorithms and solvers [2]. To assess the performance of a system, the evaluation on meaningful benchmarks is crucial and adequate tools supporting researchers and developers in this task are essential. An example of such a tool is the original `probo` [4] framework, which provides a general interface and allows the comparison of abstract argumentation solvers in terms of correctness and performance. However, this tool lacks functionality for extensive analysis and visualizations.

In this extended abstract, we present `probo2`, an end-to-end benchmark framework for abstract argumentation solvers. This framework aims at providing researchers and developers with an easy-to-use, robust, and flexible command-line tool to simplify and speed up typical tasks in benchmarking abstract argumentation solvers. Therefore, `probo2` offers functionalities to (1) generate benchmarks, (2) execute solvers and collect data, (3) verify the correctness of solvers, (4) compare the performance of solvers, (5) perform statistical analysis and—of particular interest in the research context—(6) generate "publication-ready" visualizations of results. `probo2` offers a standardized pipeline for evaluating argumentation solvers. Users can specify which performance metrics to use and how to visualize the results, allowing to compare different solvers easily. To reproduce results reliably, any experiment is completely described by a configuration file. In addition, `probo2` supports the parallelization of experiments. During development, we set a strong focus on customizability. The modular design of `probo2` allows users to modify and extend functionality to their needs. For now, the framework focuses on abstract argumentation. However, since `probo2` is still being further developed, extensions to structured argumentation frameworks are foreseen for the future.

2. Implementation Overview

probo2 is written in Python. The source code and a detailed documentation are publicly available on GitHub[1]. It is compatible with any solver implementing the ICCMA interface. All computational problems and semantics of past competitions are supported, including classical problems such as deciding acceptability and enumerating extensions as well as corresponding tasks for dynamic problems and counting tasks. probo2 accepts abstract argumentation frameworks in the ASPARTIX format [6] and the Trivial Graph Format[2]. The framework also incorporates various graph generators such as the *StableGenerator* [4] or *AFBenchGen* [3]. This enables the generation of diverse and challenging argumentation graphs. A configuration file fully describes experiments, allowing for reproducing results reliably. Correctness of solutions is verified by specifying a reference solver or providing pre-calculated solutions. In order to compare the performance of different solvers, various established performance measures such as the penalized average runtime, instance coverage, and the ICCMA scorings are available. For more extensive statistical analysis, probo2 offers parametric and non-parametric significance tests, including posthoc analysis. Users can choose between various result visualizations, including scatter plots, cactus plots, and pie charts. Tabular data can be directly exported to LatTeX, HTML, or just plain text formats. As stated before, a key feature of probo2 is its customizability. All visualizations can be customized according to user preferences. Custom functionalities can be integrated into the existing pipeline with a single command.

3. Summary

In this extended abstract, we gave a short overview on the probo2 benchmark framework. probo2 is continuously being improved and we welcome any feedback or suggestions for further features.

References

[1] Pietro Baroni, Dov Gabbay, Massimilino Giacomin, and Leendert van der Torre, editors. *Handbook of Formal Argumentation*. College Publications, 2018.

[2] Federico Cerutti, Sarah A. Gaggl, Matthias Thimm, and Johannes P. Wallner. Foundations of implementations for formal argumentation. In *Handbook of Formal Argumentation*, chapter 15. College Publications, 2018.

[3] Federico Cerutti, Massimiliano Giacomin, and Mauro Vallati. Generating challenging benchmark afs. *COMMA*, 14:457–458, 2014.

[4] Federico Cerutti, Nir Oren, Hannes Strass, Matthias Thimm, and Mauro Vallati. A benchmark framework for a computational argumentation competition. In *COMMA*, pages 459–460, 2014.

[5] 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.

[6] U. Egly, S. A. Gaggl, and S. Woltran. Answer-set programming encodings for argumentation frameworks. *Argument and Computation*, 1(2):147–177, 2010.

[7] Dov Gabbay, Massimiliano Giacomin, Guillermo R. Simari, and Matthias Thimm, editors. *Handbook of Formal Argumentation*, volume 2. College Publications, August 2021.

[1] https://github.com/aig-hagen/probo2

[2] http://en.wikipedia.org/wiki/Trivial_Graph_Format