DOI: 10.1609/aaai.12011

COMPETITION REPORT

Summary report for the third international competition on computational models of argumentation

Stefano Bistarelli[1] | Lars Kotthof[2] | Francesco Santini[1] | Carlo Taticchi[1]

[1] Department of Mathematics and Computer Science, University of Perugia, Via Vanvitelli 1, Perugia, Italy

[2] Department of Computer Science, University of Wyoming, Dept 3315, 1000 E University Ave, Laramie, WY, USA

Correspondence

Francesco Santini, Department of Mathematics and Computer Science, University of Perugia, Via Vanvitelli 1, 06123 Perugia, Italy.
Email: francesco.santini@unipg.it

Abstract

The Third International Competition on Computational Models of Argumentation (ICCMA'19) focused on reasoning tasks in abstract argumentation frameworks. Submitted solvers were tested on a selected collection of benchmark instances, including artificially generated argumentation frameworks and some frameworks formalizing real-world problems. This competition introduced two main novelties over the two previous editions: the first one is the use of the Docker platform for packaging the participating solvers into virtual "light" containers; the second novelty consists of a new track for dynamic frameworks.

Argumentation Theory is a field of Artificial Intelligence that provides formalisms for reasoning with conflicting information. Many different areas, ranging from healthcare (Longo and Hederman 2013) to explainable AI (Zhong et al. 2019), make use of notions coming from the research in computational argumentation. The International Competition on Computational Models of Argumentation (ICCMA) aims to nurture research and development of implementations for computational models of argumentation. The objectives of the competition are to provide a forum for the empirical comparison of solvers, to highlight challenges to the community, to propose new directions for research, and to provide a core of common benchmark in-stances and a representation formalism to aid in the comparison and evaluation of solvers.

For the third edition of the competition (ICCMA 2019), we proposed two main novelties. First, we organized a new track for dynamic argumentation frameworks, which benchmarks efficient approaches for the re-computation of extensions. In this track, dedicated to dynamic solvers (Bistarelli et al. 2018), results from a previously-solved instance can be used to rapidly find a solution on an argumentation framework with small modifications instead of solving the whole problem from scratch. The second novelty is the use of Docker for the containerization of the solvers. This allows them to be submitted with all dependencies for easier evaluation and to allow for the recomputation of the results in different environments. More information about the competition, including complete results and benchmarks, can be found on the competition website.[1]

An Abstract Argumentation Framework (Dung 1995) (AF), is represented by a pair <A, R> consisting of a set of arguments and a binary relationship of attack defined on them. Given a framework, it is possible to determine which set(s) of arguments (called extensions) collectively surviving the conflicts defined by R and can be accepted. A very simple example of an AF is ⟨{a, b}, {(a, b), (b, a)}⟩, where two arguments a and b attack each other. In this case, each of the two positions represented by either {a} or {b} can be intuitively valid. The acceptability of the arguments depends on the criterion used for analyzing the AF; such criteria are called semantics and starting in 1995, have been proposed by various authors in the literature. Four are proposed by Dung in his seminal paper (Dung 1995), namely the complete (CO), preferred (PR), stable (ST), and grounded (GR) semantics. All these semantics rely on the notion of conflict-freeness (accepted arguments must not attack each other) and defense (a ∈ A is defended by E ⊆ A if and only if ∀b ∈ A such that (b,a) ∈ R, ∃c ∈ E such that (c,b) ∈ R). For the competition, we consider three additional semantics: semi-stable (SST) (Caminada, Carnielli, and Dunne 2012), stage (STG) (Verheij 1996), and ideal (ID) (Dung, Mancarella, and Toni 2007).

More in detail, a complete extension is a set of arguments that do not attack each other and that includes all the defended arguments; a preferred extension is a maximally complete (with respect to set inclusion) extension; the grounded extension is the minimally complete (with respect to set inclusion) extension; and a stable extension is a complete extension such that all the arguments outside the extension are attacked by arguments inside the extension. A semi-stable extension is a complete extension that maximizes the set of the union of the acceptable arguments and the arguments they attack; a stage extension is a conflict-free extension that maximizes the set of the union of the acceptable arguments and the arguments they attack; and an ideal extension is the maximal subset of each preferred extension. The competition consists of 7 main tracks, one for each semantics. A track is composed of 4 (respectively 2 for single-status semantics, i.e., grounded and ideal) tasks. Finally, a task is a reasoning problem under a particular semantics. Following the same approach as in ICCMA 2015 and ICCMA 2017, we consider four different problems for each semantics $\sigma \in$ {CO,PR,ST,SST,STG,GR, ID}: (1) given F = ⟨A,R⟩ and a ∈ A, decide whether a is contained in some extension of σ (DC-σ); (2) given F = ⟨A,R⟩ and a ∈ A, decide whether a is contained in all extensions of σ (DS-σ); (3) given F = ⟨A, R⟩, return some set E ⊆ A that is an extension of σ (SE-σ); (4) given F = ⟨A, R⟩, enumerate all sets E ⊆ A that are extensions of σ (EE-σ).

For single-status semantics (GR and ID), only the problems SE and DC are considered (indeed EE is equivalent to SE, and DS is equivalent to DC). Also note that DC-CO and DC-PR are equivalent as well; however, in order to allow the participation in the PR track without implementing tasks on the CO semantics (or vice-versa), we kept the task. The combination of problems and semantics results in a total of 24 tasks.

The 2019 edition of ICCMA also features additional tracks to evaluate solvers on Dung's dynamic frameworks. The aim was to benchmark the solvers dedicated to efficiently computing a solution based on a previously-computed solution for a framework with minor modifications. In this case, an instance consists of an initial framework (as for the classical tracks) and a sequence of additions/deletions of attacks for the initial framework (at least 15 changes). This sequence is pro-vided in a simple text format, e.g., +att(a; b) (attack addition) or -att(d; e) (attack deletion). The output of a solver needs to report the solution for the initial framework and a solution for each modified framework, i.e., one solution per change.

The four new dynamic tracks concern the following semantics and problems, for a total of 14 different tasks: complete semantics (SE, EE, DC, DS), preferred semantics (SE, EE, DC, DS), stable semantics (SE, EE, DC, DS) and grounded semantics (only SE and DC). Each solver participating in the competition can choose to compete in an arbitrary set of tasks. If a solver chooses to participate in all tasks in a track, it also automatically participates in the corresponding track.

The competition received 9 submissions from research groups in Austria, Fin-land, France, Germany, Italy, Romania, and the UK. Among them, 3 were submitted to all tracks (including dynamic tracks). The authors of the solvers used different techniques to implement their applications. In particular, 4 were based on the transformation of argumentation problems to SAT, 1 used an ASP model, 1 relied on machine learning (or better, Deep Reinforcement Learning and Monte Carlo Tree Search), and 3 were built on custom-made algorithms.

We required participants to package their solvers in a Docker container for submission (Bistarelli et al. 2018). Docker[2] is an open-source implementation of operating-system-level virtualization, also known as containerization. Docker allows independent "containers" to run within a single Linux (or other operating system's) instance. The main motivation behind the use of Docker is to allow each solver to be de-livered with its complete run time environment to make setup and deployment easier; moreover, a dockerized application can be launched on different platforms (e.g., Windows, Linux, macOS, and in the cloud), making it possible to recompute the experiments anywhere.

We allowed 4 GB of RAM and 10 min of CPU time for each run to compute the results for an instance in both the classical and dynamic tracks. A total of 326 argumentation framework instances were selected from those used in previous competitions and 2 new benchmarks submitted for ICCMA 2019.[3]

A solver was awarded 1 point for a run if it delivered the correct result, with fractional points awarded for incomplete correct results (based on the fraction of the total results found); 5 points if it delivered an incorrect result; 0 points if the result was empty (e.g., the timeout was reached without answer) or if it was not parseable (e.g., some unexpected error message). For SE, DC, and DS, the assigned score was 1 if the solver returned the correct answer (respectively "yes", "no", or just an extension). For EE, a solver received a (0; 1] fraction of points depending on the fraction of found enumerated extensions (1 if it returned all of them). The ranking of solvers for a track was based on the sum of scores over all tasks of the track. Ties were broken by the total time it took the solver to return correct results. All tasks for one semantics had the same number of instances to avoid skewing the result. All computed results were compared to the reference solution computed by ConArg (Bistarelli, Rossi, and Santini 2017).

The timeout to compute an answer for the dynamic track was 5 min for each change (or derived framework); half of the time in the classical track for a single instance. For the solvers participating in the dynamic tracks, a result was considered correct and complete if, for n changes, n + 1 correct and complete results were given. The score for a correct and complete result was 1 as usual. A partial (incomplete) result was considered correct if it gave fewer than n + 1 answers, but each of the given answers was correct and complete (with respect to the corresponding static tasks), for all problems (SE, DC, DS, EE) in the dynamic track. A correct but incomplete result scored a value in (0; 1], depending on the fraction of correct sub-solutions given. For dynamic tasks that involved enumeration (i.e., EE), if the last solution a solver provided was correct but partial, a score in (0; 1] was assigned depending on the fraction of returned enumerated extensions in that solution. If any of the sub-solutions were incorrect, then the overall output was considered incorrect (5 points). Again, if no answer was given, 0 points were assigned (usually this happened due to a timeout). In the final ranking, ties were broken by the total time it took the solver to return correct results for all considered frameworks (starting plus changes).

The ranking of solvers for all the tracks can be found on the dedicated page of the competition website.[4] In the global ranking for the static tracks, the solvers μ-toksia, CoQuiAAS, and ASPARTIX-V19 ranked first, second, and third, respectively. μ-toksia achieved the highest score for every single track as well. It is interesting to observe that these three solvers use different approaches to solve argumentation problems, namely μ-toksia and CoQuiAAS use a SAT-based implementation, while ASPARTIX-V19 relies on an ASP solver.

In the dynamic tracks, the first and second place were also held by μ-toksia, and CoQuiAAS, respectively. In this case, however, CoQuiAAS performed better than μ-toksia in the complete and grounded tracks.

NOTES

[1] ICCMA 2019 website: https://iccma2019.dmi.unipg.it.

[2] Docker Inc.: https://www.docker.com.

[3] ICCMA'19 benchmarks: https://iccma2019.dmi.unipg.it/submissions.html.

[4] ICCMA'19 results: https://iccma2019.dmi.unipg.it/results.html.

REFERENCES

Bistarelli, Stefano, Lars Kottho, Francesco Santini, and Carlo Taticchi. 2018. "Containerisation and Dynamic Frameworks in Iccma'19." In Proceedings of the Second International Workshop on Systems and Algorithms for Formal Argumentation (SAFA 2018), volume 2171 of CEUR Workshop Proceedings, 4–9. CEUR-WS.org.

Bistarelli, Stefano, Fabio Rossi, and Francesco Santini. 2017. "A Conarg-based Library for Abstract Argumentation." In 29th IEEE International Conference on Tools with Artificial Intelligence, ICTAI, 374–81. IEEE Computer Society.

Caminada, Martin W. A., Walter Alexandre Carnielli, and Paul E. Dunne. 2012. "Semi-Stable Semantics." *Journal of Logic and Computation* 22(5): 1207–54.

Dung, Phan Minh. 1995. "On the Acceptability of Arguments and its Fundamental Role in Nonmonotonic Reasoning, Logic Programming and N-Person Games." *Artificial Intelligence* 77(2): 321–58.

Dung, Phan Minh, Paolo Mancarella, and Francesca Toni. 2007. "Computing Ideal Sceptical Argumentation." *Artificial Intelligence* 171(10-15): 642–74.

Longo, Luca, and Lucy Hederman. 2013. "Argumentation Theory for Decision Support in Health-Care: A Comparison with Machine Learning." In Brain and Health Informatics - International Conference, BHI, volume 8211 of Lecture Notes in Computer Science, 168–80. Springer.

Verheij, Bart. 1996. "Two Approaches to Dialectical Argumentation: Admissible Sets and Argumentation Stages." In Proceedings of the biannual International Conference on Formal and Applied Practical Reasoning (FAPR) workshop, 357–68. Universiteit.

Zhong, Qiaoting, Xiuyi Fan, Xudong Luo, and Francesca Toni. 2019. "An Explain-Able Multi-Attribute Decision Model Based on Argumentation." *Expert Systems With Applications* 117: 42–61.0pc

AUTHOR BIOGRAPHIES

Stefano Bistarelli is Full Professor of Computer Science at the University of Perugia. His research interests range from Artificial Intelligence to Programming Languages and his currently active research topics are Computational Argumentation and Distributed Ledgers.

Lars Kotthof is Assistant Professor at University of Wyoming, where he directs the Artificially Intelligent Manufacturing center (AIM). His research combines artificial intelligence and machine learning to build robust systems with state-of-the-art performance.

Francesco Santini is Associate Professor at the University of Perugia, after years spent at (extra) European scientific institutes. His main research interest concern Knowledge Representation and Reasoning, Multi-agent systems, and Concurrent Models and Languages.

Carlo Taticchi is PostDoc at the University of Perugia and former PhD student at Gran Sasso Science Institute (GSSI). His thesis concerns Computational Argumentation and Concurrent Programming, with a focus on labeled and weighted semantics.

How to cite this article: Bistarelli, Stefano, Lars Kotthof, Francesco Santini, and Carlo Taticchi. 2021. "Summary report for the third international competition on computational models of argumentation." *AI Magazine*. 42: 70–73. https://doi.org/10.1609/aaai.12011