Design of ICCMA 2023, 5th International Competition on Computational Models of Argumentation: A Preliminary Report

Matti Järvisalo, Tuomo Lehtonen and Andreas Niskanen

HIIT, Department of Computer Science, University of Helsinki, Finland

Abstract

ICCMA 2023 constitutes the 5th instantiation of International Competitions on Computational Models of Argumentation, the main series of international competitions for evaluating the state of the art in practical system implementations for argumentative reasoning. In this short preliminary report, we provide an overview of the design of ICCMA 2023.

1. Introduction

The series of International Competitions on Computational Models of Argumentation (ICCMA, http://argumentationcompetition.org) aims at nurturing research and development of implementations for computational models of argumentation. The year 2023 marks the 5th instantiation of the biennial ICCMA competitions. ICCMA 2023 (https://iccma2023.github.io) welcomed contributions from the community at large in the forms of new argumentation reasoning problem benchmarks, and implementations of argumentation reasoners (for abstract and assumption-based argumentation) to be evaluated within ICCMA 2023 on a heterogeneous collection of benchmarks. The community at large was invited to submit argumentation reasoning system implementations (solvers) for participation in the competition as well as interesting and/or challenging benchmark instances for evaluating solvers competing in any of the ICCMA 2023 competition tracks. We provide a short preliminary overview of the design of ICCMA 2023. The results of the competition will be presented in conjunction with the KR 2023 conference after the writing of this overview and made available through the competition webpages.

2. Competition Tracks

ICCMA 2023 consists of four tracks: the *main track* and the special *approximate*, *dynamic*, and *ABA* tracks. Each track is composed of multiple subtracks, defined by a combination of a reasoning problem and an argumentation semantics. We use the following shorthands for semantics and reasoning tasks: CO, ST, PR, SST, STG, ID for complete, stable, preferred,

Arg&App 2023: International Workshop on Argumentation and Applications, September 2023, Rhodes, Greece

matti.jarvisalo@helsinki.fi (M. Järvisalo); tuomo.lehtonen@helsinki.fi (T. Lehtonen); andreas.niskanen@helsinki.fi (A. Niskanen)

0000-0003-2572-063X (M. Järvisalo); 0000-0003-3197-2075 (A. Niskanen)

semi-stable, stage and ideal semantics, respectively, and DC, DS and SE for credulous and skeptical acceptance, and finding a single extension, respectively. Argumentation systems could be submitted for evaluation into any choice of subtracks, i.e., no requirement to support e.g. all semantics for a specific reasoning problem, or all reasoning problems for a specific semantics were enforced.

Main Track concerns solvers for reasoning in abstract argumentation [1]. The focus of the Main track is to evaluate sequential core argumentation reasoning engines available in open source. Systems combining different core reasoning engines e.g. via portfolio-style techniques, systems employing parallel computations via the use of multiple processor cores, as well as systems which will not be made available in open source were invited to the special No-Limits track which consists of the same subtracks as the Main track. The ranking is otherwise the same as for the Main track, but wall-clock time is used instead of CPU time. The following combinations of the semantics and reasoning modes constitute the Main and No-Limits subtracks: DC-{CO|ST|SST|STG}, DS-{PR|ST|SST|STG}, SE-{PR|ST|SST|STG|ID}.

Approximate Track concerns in-exact solvers developed for abstract argumentation, i.e., solvers which may not in all cases provide correct YES/NO answers to credulous/skeptical queries. Correctness requirements and ranking are different than other tracks: incorrect solutions are simply discarded and only the number of correct solutions is taken into account. The subtracks in the Approximate track are DC-{CO|ST|SST|STG|ID}, DS-{PR|ST|SST|STG}.

Dynamic Track invites solvers built especially for answering credulous/skeptical queries over sequences of related AFs. Dynamic changes to an initial AF and acceptance queries are issued by different applications via IPAFAIR, an API for incremental reasoning in abstract argumentation specified for the first time for ICCMA 2023. Similarly to the Main track, an instance of the Dynamic track is an AF and a query argument. An instance is given as input to a program which modifies the initial AF by iteratively adding and deleting arguments and attacks, and checks whether the query argument is accepted in the resulting modified AFs. Resource limits are applied to this program as a whole, and an instance is solved exactly when this program terminates correctly. The subtracks in the Dynamic track are DC-CO, DS-PR, DC-ST, and DS-ST.

ABA Track concerns solvers developed for reasoning in the structured argumentation formalism of Assumption-based Argumentation (ABA) [2], specifically focusing on so-called flat ABA frameworks in the commonly studied logic programming fragment of ABA. In this fragment, atoms are derived from assumptions using rules with a list of atoms in the body and a non-assumption atom as the head. Assumptions have contraries, the derivation of which produces an attack on this assumption. The subtracks for the ABA track are DC-{CO|ST}, DS-{PR|ST}, SE-{PR|ST}.

Ranking Scheme. For the Main, Dynamic, Approximate and ABA tracks, the score of a solver on a subtrack is the sum of PAR-2 scores (CPU time if instance solved within resource limits, 2× the per-instance time limit otherwise) of the solver over all instances of a subtrack. The No-limits ranking is otherwise the same but wall-clock time is used instead of CPU time. The winner of a subtrack is the solver with the lowest score. For the Approximate track, the solver with the largest number of correctly solved instances wins. If needed, cumulative CPU running time over solved instances is used as a tie-breaker.

Input-Output Interface. In short (see the website for details), a specific compact numerical input format for AFs was enforced for the Main, Dynamic and Approximate tracks. The format was also extended for use in the ABA track by beginning-of-line identifiers for distinguishing between assumptions, rules and contraries. In the Dynamic track, I/O is implemented using IPAFAIR, an incremental API for reasoning in AFs, with functionality for initializing a solver with an input AF and semantics, adding and deleting arguments and attacks, and performing credulous and skeptical acceptance queries. For details on IPAFAIR, see https://bitbucket.org/coreo-group/ipafair.

3. Rules and Execution

The rules are available in full on the ICCMA 2023 webpages. As a new development for 2023, the requirement of witnessing certificates was enforced in the main track as follows. For DS-σ, if the query argument is credulously accepted, solvers should output “YES” *along with a certificate*, i.e., a σ-extension containing the query. Analogously, for DS-σ, if the query argument is not skeptically accepted, solvers should output “NO” along with a certificate, i.e., a σ-extension not containing the query. The certificates were checked as follows (with the subtrack specification, an AF, a query argument, and an output produced by a solver participating in the Main track as input). First, we verified that a certificate is contained in the output in the required cases (SE apart from “NO” answers on SE-ST, “YES” answers for DC, and “NO” answers for DS), and for DC and DS, that it contains (DC) or does not contain (DS) the query. For subtracks involving CO and ST semantics, we constructed a standard SAT encoding [3] and verified that the certificate extends to a satisfiable assignment. For subtracks involving PR, SST, and STG semantics, we built the standard SAT encoding of CO (for PR and SST) or conflict-free (for STG) semantics, and in addition to verifying that the certificate yields a satisfiable assignment, verified the absence of a counterexample (a superset or a range-superset) via a SAT solver call. All solver calls were performed using the SAT solver Glucose [4] (v4.1) invoked via PySAT [5]. The UNSAT proofs produced by Glucose were recorded. For the SE-ID track, we instead verified that all solvers reported the same ideal extension.

The organizers used fuzz testing to check for potential buggy behavior exhibited by submitted solvers before the execution of the competition. When bugs were detected, the authors of the solvers concerned were contacted and bug fixes were allowed to the extent feasible in order to execute the competition on time. As for further solver requirements, solver descriptions were mandatory. Furthermore, for all tracks apart from No-Limits, solver source code originating from the authors (including modifications to third-party source code such as SAT solvers as part of a solver) must be submitted together with a corresponding solver binary. In practice, ICCMA 2023 was executed on a computing cluster of University of Helsinki, Finland, with

2.60-GHz Intel Xeon E5-2670 CPUs and 57GB RAM under AlmaLinux 8.4, including GCC 12.2.0, Clang 12.0.1, Boost 1.76.0, GLib 2.68.2, Rust 1.70.0, Java 17.0.4, and Python 3.9.5. A per-instance memory limit of 16 GB was enforced on all subtracks. A 1200-second per-instance time limit was enforced on the Main, Dynamic, and ABA tracks; for the Approximate track we set a 60-s per-instance time limit.

4. Benchmarks

Abstract Argumentation: Main, Approximate, and Dynamic Tracks. To sample benchmarks for the Main, Approximate, and Dynamic tracks, we collected all benchmark AFs submitted to ICCMA 2017 [6] (11 domains) and ICCMA 2019 [7] (2 domains). For the so-called GroundedGenerator, SccGenerator, and StableGenerator domains, new AFs (100 per domain) with similar parameters were generated by Matthias Thimm. In addition to these, a benchmark generator **crusti_g2io** (by Jean-Marie Lagniez, Emmanuel Lonca, Jean-Guy Mailly, Julien Rossit) submitted to ICCMA 2023 was used with suggested parameters to generate a new set of 450 AFs. This procedure resulted in 14 benchmark domains. From each of these, we sampled 25 AFs for the final benchmark set, with the exception of the crusti_g2io domain, from which 32 AFs were sampled. Finally, query arguments were sampled from the set of arguments with a non-zero number of attackers which are not self-attackers, to avoid trivial acceptance queries.

ABA Track. The benchmarks for the ABA track were generated with a simple random instance generator. The varying parameters are the number of atoms (25, 100, 500, 2000 or 5000), the proportion of atoms that are assumptions (10% or 30%), the maximum number of rules deriving each sentence (5 or 10), and the maximum size of each rule body (5 or 10). Ten instances with each combination of these parameters were generated for a total of 400 instances. For acceptance problems, the query for each instance was selected at random from the atoms for which there is at least one derivation in the given instance.

5. Participants

Number of solvers submitted for each track: three for the Main track, one for the No-Limits track, five for the Approximate track, and five for the ABA track.

Crustabri (by Jean-Marie Lagniez, Emmanuel Lonca and Jean-Guy Mailly) is a SAT-based solver—a rewritten version of CoQuiAAS [8]—supporting all subtracks in the Main track and ABA track, as well as DC-CO, DC-ST, and DS-ST in the Dynamic track.

Fudge [9] (by Matthias Thimm, Federico Cerutti and Mauro Vallati) is a SAT-based solver with support for all subtracks in the Main track.

μ-**toksia** [10, 11] (by Andreas Niskanen and Matti Järvisalo) is a SAT-based solver with support for all subtracks of the Main and Dynamic tracks.

PORTSAT (by Sylvain Declercq, Quentin Januel Capellini, Christophe Yang, Jérôme Delobelle and Jean-Guy Mailly) is a solver based on a portfolio of SAT solvers with support for DC-CO, DC-ST, DS-PR, DS-ST, SE-PR, and SE-ST subtracks of the No-Limits track.

κ-solutions (by Christian Pasero and Johannes P. Wallner) is a SAT-based solver with support for DC-CO, DC-ST, and DS-ST in the Dynamic track.

AFGCN v2 [12] (by Lars Malmqvist) is based on employing graph convolutional neural networks, and supports all subtracks of the Approximate track.

ARIPOTER-Degrees (by Jérôme Delobelle, Jean-Guy Mailly and Julien Rossit) is based on computing the grounded extension and comparing the in-degree and out-degree of the query argument, and supports all subtracks of the Approximate track.

ARIPOTER-HCAT (by Jérôme Delobelle, Jean-Guy Mailly and Julien Rossit) is based on the grounded and h-Categorizer gradual semantics, and supports all subtracks of the Approximate track.

fargo-limited (by Matthias Thimm) is based on an exact DPLL-style search algorithm for admissible sets, and supports all subtracks of the Approximate track.

harper++ (by Matthias Thimm) is based on approximating all acceptance tasks by using grounded semantics, and supports all subtracks of the Approximate track.

AcbAr [13] (by Tuomo Lehtonen, Anna Rapberger, Markus Ulbricht and Johannes P. Wallner) is based on translating ABA frameworks to AFs with support for all reasoning tasks in the ABA track.

ASPforABA [14, 15] (by Tuomo Lehtonen, Matti Järvisalo and Johannes P. Wallner) is an answer set programming approach for ABA with support for all reasoning tasks in the ABA track.

ASTRA (Andrei Popescu and Johannes P. Wallner) employs dynamic programming and supports DC-CO, DC-ST, DS-ST and SE-ST in the ABA track.

flexABle [16, 17] (Martin Diller, Sarah Alice Gaggl, Piotr Gorczynca) implements specialized ABA algorithms, flexible dispute derivations and supports DC-CO and DC-ST in the ABA track.

As agreed with the ICCMA steering committee, for transparency all solver submissions involving any of the organizers of ICCMA 2023 were made known to the ICCMA steering committee before the submission deadline. In addition, benchmark selection was done using a random seed—811543731122527—concatenated from numbers sent separately to the organizers by each ICCMA steering committee member. The seed and benchmark selection scripts are available on the ICCMA 2023 website.

Acknowledgments

Work financially supported by Academy of Finland (grants 322869, 347588, 356046) and University of Helsinki Doctoral Programme in Computer Science DoCS. The organizers wish to thank the Finnish Computing Competence Infrastructure (FCCI) for supporting this project with computational and data storage resources.

References

[1] P. M. Dung, On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games, Artificial Intelligence 77 (1995) 321–358.

[2] A. Bondarenko, P. M. Dung, R. A. Kowalski, F. Toni, An abstract, argumentation-theoretic approach to default reasoning, Artificial Intelligence 93 (1997) 63–101.

[3] P. Besnard, S. Doutre, Checking the acceptability of a set of arguments, in: J. P. Delgrande, T. Schaub (Eds.), 10th International Workshop on Non-Monotonic Reasoning (NMR 2004), Whistler, Canada, June 6-8, 2004, Proceedings, 2004, pp. 59–64.

[4] G. Audemard, L. Simon, On the glucose SAT solver, Int. J. Artif. Intell. Tools 27 (2018) 1840001:1–1840001:25.

[5] A. Ignatiev, A. Morgado, J. Marques-Silva, Pysat: A python toolkit for prototyping with SAT oracles, in: O. Beyersdorff, C. M. Wintersteiger (Eds.), Theory and Applications of Satisfiability Testing - SAT 2018 - 21st International Conference, SAT 2018, Held as Part of the Federated Logic Conference, FloC 2018, Oxford, UK, July 9-12, 2018, Proceedings, volume 10929 of *Lecture Notes in Computer Science*, Springer, 2018, pp. 428–437.

[6] S. A. Gaggl, T. Linsbichler, M. Maratea, S. Woltran, Design and results of the Second International Competition on Computational Models of Argumentation, Artif. Intell. 279 (2020).

[7] S. Bistarelli, L. Kotthoff, F. Santini, C. Taticchi, Summary report for the Third International Competition on Computational Models of Argumentation, AI Mag. 42 (2021) 70–73.

[8] J. Lagniez, E. Lonca, J. Mailly, CoQuiAAS: A constraint-based quick abstract argumentation solver, in: 27th IEEE International Conference on Tools with Artificial Intelligence, ICTAI 2015, Vietri sul Mare, Italy, November 9-11, 2015, IEEE Computer Society, 2015, pp. 928–935.

[9] M. Thimm, F. Cerutti, M. Vallati, Skeptical reasoning with preferred semantics in abstract argumentation without computing preferred extensions, in: Z. Zhou (Ed.), Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI 2021, Virtual Event / Montreal, Canada, 19-27 August 2021, ijcai.org, 2021, pp. 2069–2075.

[10] A. Niskanen, M. Järvisalo, Algorithms for dynamic argumentation frameworks: An incremental SAT-based approach, in: G. D. Giacomo, A. Catalá, B. Dilkina, M. Milano, S. Barro, A. Bugarín, J. Lang (Eds.), ECAI 2020 - 24th European Conference on Artificial Intelligence, 29 August-8 September 2020, Santiago de Compostela, Spain, August 29 - September 8, 2020 - Including 10th Conference on Prestigious Applications of Artificial Intelligence (PAIS 2020), volume 325 of *Frontiers in Artificial Intelligence and Applications*, IOS Press, 2020, pp. 849–856.

[11] A. Niskanen, M. Järvisalo, μ-toksia: An efficient abstract argumentation reasoner, in: D. Calvanese, E. Erdem, M. Thielscher (Eds.), Proceedings of the 17th International Conference on Principles of Knowledge Representation and Reasoning, KR 2020, Rhodes, Greece, September 12-18, 2020, 2020, pp. 800–804.

[12] L. Malmqvist, Approximate Solutions to Abstract Argumentation Problems Using Graph Neural Networks., Ph.D. thesis, University of York, 2022.

[13] T. Lehtonen, A. Rapberger, M. Ulbricht, J. P. Wallner, Argumentation Frameworks Induced by Assumption-based Argumentation: Relating Size and Complexity, in: Proceedings

of the 20th International Conference on Principles of Knowledge Representation and Reasoning, 2023, pp. 440–450.

[14] T. Lehtonen, J. P. Wallner, M. Järvisalo, Declarative algorithms and complexity results for assumption-based argumentation, Journal of Artificial Intelligence Research 71 (2021) 265–318.

[15] T. Lehtonen, J. P. Wallner, M. Järvisalo, Harnessing incremental answer set solving for reasoning in assumption-based argumentation, Theory Pract. Log. Program. 21 (2021) 717–734.

[16] M. Diller, S. A. Gaggl, P. Gorczyca, Flexible dispute derivations with forward and backward arguments for assumption-based argumentation, in: P. Baroni, C. Benzmüller, Y. N. Wáng (Eds.), Logic and Argumentation - 4th International Conference, CLAR 2021, Hangzhou, China, October 20-22, 2021, Proceedings, volume 13040 of *Lecture Notes in Computer Science*, Springer, 2021, pp. 147–168.

[17] M. Diller, S. A. Gaggl, P. Gorczyca, Strategies in flexible dispute derivations for assumption-based argumentation, in: S. A. Gaggl, J. Mailly, M. Thimm, J. P. Wallner (Eds.), Proceedings of the Fourth International Workshop on Systems and Algorithms for Formal Argumentation co-located with the 9th International Conference on Computational Models of Argument (COMMA 2022), volume 3236 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2022, pp. 59–72.