Introducing the Fourth International Competition on Computational Models of Argumentation

Jean-Marie LAGNIEZ [a], Emmanuel LONCA [b], Jean-Guy MAILLY [c], Julien ROSSIT [c]
[a] *Huawei Technologies Ltd, Boulogne-Billancourt,Île-de-France, France*
jean.marie.lagniez@huawei.com
[b] *CRIL, Université d'Artois & CNRS, France*
lonca@cril.fr
[c] *LIPADE, Université de Paris, France*
{*jean-guy.mailly, julien.rossit*}*@u-paris.fr*

Abstract. Since 2015, the International Competition on Computational Models of Argumentation (ICCMA) has allowed to compare the different algorithms for solving some classical reasoning problems in the domain of (abstract) argumentation. In this paper, we describe the rules of the fourth ICCMA, that will be held in 2021. We introduce some minor modifications regarding the existing tracks (*i.e.* reasoning with static and dynamic abstract argumentation). Also, for the first time, we present a new track dedicated to structured argumentation, more precisely Assumption-based Argumentation.

Keywords. argumentation solvers, competition, ICCMA

Formal argumentation [1] is a major topic in symbolic artificial intelligence. This formalism allows to reason with conflicting information, and has applications *e.g.* in automated negotiation [2] or decision making [3]. The most classical reasoning tasks in this kind of formalism are intractable in the general case (see *e.g.* [4] for an overview of computational complexity in formal argumentation). This has conducted to the organization of the International Competition on Computational Models of Argumentation (ICCMA), that allows to compare the efficiency of the different algorithms that have been proposed for these reasoning problems. The previous competitions [5,6,7] have shown that, in spite of the theoretical hardness of argumentative reasoning, some powerful techniques allow to handle them efficiently.

In this paper, we describe how the next competition (ICCMA 2021) will be organized. In Section 1, we recall the definitions of the formal concepts related to the different tracks of ICCMA 2021. Then we define the different tracks of the competition. Sections 2 and 3 describe the minor changes that we make to the existing tracks (corresponding to reasoning with Dung's AFs, respectively in a static or dynamic way), while Section 4 introduces for the first time structured argumentation at ICCMA, namely Assumption-based Argumentation. We describe the scoring rules of the different tracks in Section 5. Finally, Section 6 concludes the paper.

1. Background

1.1. Abstract Argumentation

An abstract argumentation framework (AF) [8] is a directed graph $F = \langle A, R \rangle$, where A is the set of arguments, and $R \subseteq A \times A$ is the attack relation. For $a, b, c \in A$, we say that *a attacks b* if $(a, b) \in R$. If in turn b attacks c, then *a defends c* against b. Similarly, a set $S \subseteq A$ attacks (respectively defends) an argument b if there is some $a \in S$ that attacks (respectively defends) b. For $S \subseteq A$ a set of arguments, S^+ is the set of arguments that are attacked by S, formally $S^+ = \{b \in A \mid \exists a \in S \text{ s.t. } (a, b) \in R\}$. The range of S is $S^\oplus = S \cup S^+$.

Different semantics have been defined for evaluating the acceptability of (sets of) arguments.

Definition 1. *Given an AF $F = \langle A, R \rangle$, a set of arguments $S \subseteq A$ is conflict-free iff $\forall a, b \in S$, $(a, b) \notin R$. A conflict-free set S is admissible iff $\forall a \in S$, S defends a against all its attackers. Conflict-free and admissible sets are respectively denoted by $\mathbf{CF}(F)$ and $\mathbf{ADM}(F)$.*

Now, we formally introduce the extension-based semantics. For $S \subseteq A$,

- *$S \in \mathbf{CO}(F)$ iff $S \in \mathbf{ADM}(F)$ and $\forall a \in A$ that is defended by S, $a \in S$;*
- *$S \in \mathbf{PR}(F)$ iff S is a $\subseteq$-maximal admissible set;*
- *$S \in \mathbf{ST}(F)$ iff S is a conflict-free set that attacks each $a \in A \setminus S$;*
- *$S \in \mathbf{SST}(F)$ iff $S \in \mathbf{CO}(F)$ and there is no $S_2 \in \mathbf{CO}(F)$ s.t. $S^\oplus \subset S_2^\oplus$;*
- *$S \in \mathbf{STG}(F)$ iff $S \in \mathbf{CF}(F)$ and there is no $S_2 \in \mathbf{CF}(F)$ s.t. $S^\oplus \subset S_2^\oplus$;*
- *$S \in \mathbf{ID}(F)$ iff $S \in \mathbf{ADM}(F)$, $S \subseteq \cap\mathbf{PR}(F)$, and there is no $S_2 \subseteq \cap\mathbf{PR}(F)$ such that $S_2 \in \mathbf{ADM}(F)$ and $S \subset S_2$.*

CO, **PR**, **ST**, **SST**, **STG** and **ID** stand (respectively) for the complete, preferred, stable [8], semi-stable [9], stage [10] and ideal [11] semantics. We refer the interested reader to [12] for more details about these semantics.

For $\sigma \in \{\mathbf{CO}, \mathbf{PR}, \mathbf{ST}, \mathbf{SST}, \mathbf{STG}, \mathbf{ID}\}$ a semantics, an argument $a \in A$ is credulously (respectively skeptically) accepted in $F = \langle A, R \rangle$ with respect to σ iff $a \in S$ for some (respectively each) $S \in \sigma(F)$.

1.2. Assumption-based Argumentation

Now, let us introduce a particular framework for structured argumentation, namely Assumption-based Argumentation (ABA) [13]. ABA is one of the most popular structured argumentation frameworks, with applications in various domains, like *e.g.* information seeking and inquiry dialogues [14], decision making in a medical context [15], or explanation of automated decisions [16]. An ABA framework is a tuple $F = \langle L, R, A, \overline{} \rangle$, where:

- L is a set of symbols called the language;
- R is a set of rules of the form $x_0 \leftarrow x_1, \ldots, x_n$, with $x_i \in L$ for $i \in \{0, \ldots, n\}$ and $n \geq 0$;
- $A \subseteq L$ is a non-empty set of particular symbols called assumptions;
- $\overline{} : A \to L$ is a total mapping that expresses a notion of contrariness.

In a rule $x_0 \leftarrow x_1, \ldots, x_n$, the left-hand part ($x_0$) is called the head of the rule, while the right-hand part ($x_1, \ldots, x_n$) is called the body. A rule $x_0 \leftarrow$ with no body can be interpreted as $x_0 \leftarrow \top$.

A deduction for $x \in L$ supported by $X_L \subseteq L$ and $X_R \subseteq R$ is a (finite) tree rooted in x, with nodes labeled by symbols in L or $\top$, such that each leaf is either a symbol in X_L or $\top$, and for each internal node with label x', its children are the elements $x_1, \ldots, x_n$ of the body of some rule $x' \leftarrow x_1, \ldots, x_n \in X_R$. An argument for the claim $x \in L$ supported by $X_A \subseteq A$ (denoted by $X_A \vdash x$) is a deduction for x supported by X_A (and some $X_R \subseteq R$). An argument $A_1 \vdash x_1$ attacks an argument $A_2 \vdash x_2$ iff x_1 is the contrary of some assumption in A_2.

Now, there are two equivalent ways of reasoning with an ABA framework: either the status of arguments is evaluated, or the status of assumptions. Here, we choose to consider the latter.

Definition 2. *Given $F = \langle L, R, A, \overline{} \rangle$ an ABA framework, a set of assumptions $A_1 \subseteq A$ attacks a set of assumptions $A_2 \subseteq A$ iff an argument supported by a subset of A_1 attacks an argument supported by a subset of A_2. A set of assumptions defends an assumption a if it attacks each set of assumptions that attacks a. Then, given a set of assumptions $X_A \subseteq A$,*

- *$X_A \in \mathbf{CF}(F)$ iff it does not attack itself;*
- *$X_A \in \mathbf{ADM}(F)$ iff $X_A \in \mathbf{CF}(F)$ and X_A defends all its elements;*
- *$X_A \in \mathbf{CO}(F)$ iff $X_A \in \mathbf{ADM}(F)$ and $\forall a \in A$ that is defended by X_A, $a \in X_A$;*
- *$X_A \in \mathbf{PR}(F)$ iff X_A is a $\subseteq$-maximal admissible set;*
- *$S \in \mathbf{ST}(F)$ iff $X_A \in \mathbf{CF}(F)$ and X_A attacks each $a \in A \setminus X_A$.*

2. Abstract Argumentation Track

The first track of the competition is concerned by static abstract argumentation. We introduce some minor changes in this track, compared to the previous editions.

The set of semantics remains the same, except for the grounded semantics, *i.e.* we consider the ones introduced in Definition 1: $\sigma \in \{\mathbf{CO}, \mathbf{PR}, \mathbf{ST}, \mathbf{SST}, \mathbf{STG}, \mathbf{ID}\}$. The grounded semantics was removed since its low complexity makes it easily computable by most solvers for any problem. For each of them, we define one sub-track, that will be divided into several reasoning problems.

- **CE**-σ: Given an AF $F = \langle A, R \rangle$, give the number of σ-extensions of F.
- **SE**-σ: Given an AF $F = \langle A, R \rangle$, give one σ-extension of F.
- **DC**-σ: Given an AF $F = \langle A, R \rangle$ and $a \in A$ an argument, is a credulously accepted in F?
- **DS**-σ: Given an AF $F = \langle A, R \rangle$ and $a \in A$ an argument, is a skeptically accepted in F?

The main difference for this track is the introduction of **CE**-σ instead of **EE**-σ: this is the counting of extensions, while the previous competitions considered the enumeration of the whole set of extensions. We made this choice because of the space complexity of the extension enumeration, which does not ensure solutions can be represented in practice, even for easy problems. **CE**-σ can be simply implemented from a **EE**-σ algorithm and is hard-enough to provide challenging tracks – counting the extensions is a $\#P$-complete problem: given such an oracle, one can solve any decision problem which complexity is in the polynomial hierarchy using a polynomial time [17].

Let us also mention that only two problems are included in the **ID** track: since there is only one ideal extension for any AF, **CE-ID** is trivial, contrary to **SE-ID**, and **DC-ID** coincides with **DS-ID**.

The input file format (`.apx` or `.tgf`), the command line interface and the solver output will be similar to the ones from the previous edition of ICCMA.

3. Dynamic Argumentation Track

Argumentation dynamics has been considered for the first time at ICCMA 2019. Recall that it consists in solving a sequence of (similar) tasks in a dynamic environment. For instance, a solver can be asked to determine whether the argument a is credulously accepted with respect to the stable semantics in a given AF F. Then, the same question is asked after the addition or removal of some attack or argument in the AF. Such an addition or removal is called an *update* of F.

We consider the three classical Dung's semantics [8] mentioned at Definition 1: $\sigma \in \{\mathbf{CO}, \mathbf{PR}, \mathbf{ST}\}$, and the same problems as for the abstract argumentation track (**CE**-σ, **SE**-σ, **DC**-σ, **DS**-σ). However, we slightly modify the input of the solvers. At the previous competition, the dynamic solvers were reading in a (`.apx` or `.tgf`) file the initial AF, and then in another file the whole set of updates. This method allows the solver to be tuned for the set of arguments that will be added. But it does not seem possible to know the whole set of arguments in advance, for some applications of argumentation (for instance, in automated negotiation, an agent may know some of the arguments of her opponents, but generally not all of them [2]).

In order to be suited to more realistic situations, for ICCMA 2021, the dynamic solvers will have a fully "online" behavior: after reading the initial AF in a file (similarly to the abstract argumentation track), the solver will wait for updates on the standard input. Each update will be provided to the solver only after it has printed the correct answer to the previous request (the same sequences of updates will be provided to all solvers). An empty line will be given to the solver to indicate that the computation is over. We plan to provide a software helping developers to deal with this new behavior before the submission deadline.

4. Structured Argumentation Track: ABA

For this first introduction of structured argumentation to ICCMA, we will consider the same reasoning tasks as for the first track (**CE**-σ, **SE**-σ, **DC**-σ and **DS**-

σ), or more precisely their assumption counterpart, for $\sigma \in \{\mathbf{CO}, \mathbf{PR}, \mathbf{ST}\}$ (see Definition 2).

The solvers input will be described in an adaptation of the ASPARTIX format that is used for the other tracks. For instance, we consider the ABA framework $F = \langle L, R, A, \overline{} \rangle$ with $L = \{a, b, c, p, q, r, s, t\}$, $R = \{(p \leftarrow q, a), (q \leftarrow), (r \leftarrow b, c)\}$, $A = \{a, b, c\}$ and $\overline{a} = r$, $\overline{b} = s$, $\overline{c} = t$. This example, borrowed from [13], can be represented as the following `.apx` file:

```
rule(p,q,a).
rule(q).
rule(r,b,c).
assum(a).
assum(b).
assum(c).
cont(a,r).
cont(b,s).
cont(c,t).
```

For each line `rule(...).` in the file, the first parameter corresponds to the head of the rule, and the other (optional) parameters correspond to the body. For instance, `rule(p,q,a).` represents the rule $p \leftarrow q, a$, and `rule(q).` corresponds to $q \leftarrow$. Assumptions and the contrariness mapping are represented, respectively, by `assum(...).` and `cont(...).` lines. Finally, the language does not need to be explicitly given, since it is simply the set of all the symbols that appear in the rules and assumptions.

5. Scoring rules

Now let us describe the scoring rules. For each sub-track, the solvers' ranking will be defined as follows:

- in case of any wrong result, the solver is excluded of the sub-track; [1]
- for every correct answer in the runtime limit, the solver gets a score of 1; in case of timeout or non-parsable output, the solver gets a score of 0;
- in case of ties, the cumulated runtime over the instances that were correctly solved will be used.

With this method, we will define a ranking for each sub-track (recall that a sub-track is made of two or four reasoning tasks, all associated with the same semantics). This means that there will be twelve rankings in the competition:

- six rankings for the Abstract Argumentation Track;
- three rankings for the Dynamic Argumentation Track;
- three rankings for the Structured Argumentation Track.

In order to be ranked, a solver needs to participate to the full sub-track (*i.e.* the two or four reasoning tasks), however there is no requirement that a solver should participate to all the (sub-)tracks.

[1] A test phase will be conducted before the actual competition.

6. Conclusion

This paper describes the preliminary design of ICCMA 2021. In the next months, we will advertise the competition through a call for participation, as well as a call for benchmarks. The detailed technical requirements for the solvers will be communicated at this time. To help the participants to prepare their solver, we will also provide some material before the deadline for the submission (including preliminary benchmarks and the software dedicated to dynamic argumentation), and we will perform some tests on the submitted solver before the actual competition phase. This will allow to identify and correct minor bugs that might appear in the solvers. We refer the reader to the competition website for up-to-date information: `http://argumentationcompetition.org/2021/`

References

[1] Baron P, Gabbay DM, Giacomin M. Handbook of Formal Argumentation. College Publications; 2018.

[2] Dimopoulos Y, Mailly JG, Moraitis P. Argumentation-based Negotiation with Incomplete Opponent Profiles. In: Proc. of AAMAS'19; 2019. p. 1252–1260.

[3] Amgoud L, Vesic S. On the Use of Argumentation for Multiple Criteria Decision Making. In: Proc. of IPMU'12; 2012. p. 480–489.

[4] Dvorák W, Dunne PE. Computational Problems in Formal Argumentation and their Complexity. In: Baroni P, Gabbay D, Giacomin M, van der Torre L, editors. Handbook of Formal Argumentation. College Publications; 2018. p. 631–688.

[5] Thimm M, Villata S. The first international competition on computational models of argumentation: Results and analysis. Artif Intell. 2017;252:267–294.

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

[7] Bistarelli S, Kotthoff L, Santini F, Taticchi C. Containerisation and Dynamic Frameworks in ICCMA'19. In: Proc. of SAFA'18; 2018. p. 4–9.

[8] Dung PM. On the Acceptability of Arguments and its Fundamental Role in Nonmonotonic Reasoning, Logic Programming and n-Person Games. Artif Intell. 1995;77(2):321–358.

[9] Caminada MWA, Carnielli WA, Dunne PE. Semi-stable semantics. J Log Comput. 2012;22(5):1207–1254.

[10] Verheij B. Two approaches to dialectical argumentation: admissible sets and argumentation stages. In: Proc.of NAIC'96; 1996. p. 357–368.

[11] Dung PM, Mancarella P, Toni F. Computing ideal sceptical argumentation. Artif Intell. 2007;171(10-15):642–674.

[12] Baroni P, Caminada M, Giacomin M. Abstract Argumentation Frameworks and Their Semantics. In: Baroni P, Gabbay D, Giacomin M, van der Torre L, editors. Handbook of Formal Argumentation. College Publications; 2018. p. 159–236.

[13] Toni F. A tutorial on assumption-based argumentation. Argument & Computation. 2014;5(1):89–117.

[14] Fan X, Toni F. Agent Strategies for ABA-based Information-seeking and Inquiry Dialogues. In: Proc. of ECAI'12; 2012. p. 324–329.

[15] Fan X, Craven R, Singer R, Toni F, Williams M. Assumption-Based Argumentation for Decision-Making with Preferences: A Medical Case Study. In: Proc. of CLIMA XIV; 2013. p. 374–390.

[16] Zhong Q, Fan X, Toni F, Luo X. Explaining Best Decisions via Argumentation. In: Proc. of ECSI'14; 2014. p. 224–237.

[17] Toda S. On the Computational Power of PP and +P. In: 30th Annual Symposium on Foundations of Computer Science, Research Triangle Park, North Carolina, USA, 30 October - 1 November 1989. IEEE Computer Society; 1989. p. 514–519.