A Logical Encoding for k-m-Realization of Extensions in Abstract Argumentation*

Jean-Guy Mailly[0000-0001-9102-7329]

Université Paris Cité, LIPADE, F-75006 Paris, France
jean-guy.mailly@u-paris.fr

Abstract. We study the notion of realization of extensions in abstract argumentation. It consists in reversing the usual reasoning process: instead of computing the extensions of an argumentation framework, we want to determine whether a given set of extensions corresponds to some (set of) argumentation framework(s) (AFs); and more importantly we want to identify such an AF (or set of AFs) that realizes the set of extensions. While deep theoretical studies have been concerned with realizability of extensions sets, there are few computational approaches for solving this problem. In this paper, we generalize the concept of realizability by introducing two parameters: the number k of auxiliary arguments (*i.e.* those that do not appear in any extension), and the number m of AFs in the result. We define a translation of k-m-realizability into Quantified Boolean Formulas (QBFs) solving. We also show that our method allows to guarantee that the result of the realization is as close as possible to some input AF. Our method can be applied in the context of AF revision operators, where revised extensions must be mapped to a set of AFs while ensuring some notion of proximity with the initial AF.

Keywords: Abstract Argumentation · Semantics Realizability · Logic-based Encoding.

Abstract argumentation frameworks (AFs) [11] are one of the most prominent models in the domain of computational argumentation. Reasoning with such AFs usually relies on the notion of extensions, *i.e.* sets of jointly acceptable arguments. There are now many efficient approaches for computing the extensions and determine the acceptability of arguments (see *e.g.* [19, 6, 24, 31, 14, 20]). However, the opposite question (how to find an AF that corresponds to a given set of extensions) has mainly received an attention on the theoretical side. This is the notion of realizability [12, 3], *i.e.* given a set of extensions and a semantics, is there an AF such that its extensions w.r.t. the given semantics correspond to the expected ones? But there are almost no study of computational approaches for building this AF in the case where it exists. This question has a practical interest in the context of AF revision [7, 10] and merging [9], where the notion of generation operators is closely related to realizability. Intuitively, these works

* This work benefited from the support of the project AGGREEY ANR-22-CE23-0005 of the French National Research Agency (ANR).

revise or merge AFs at the level of their extensions (in a way, using extensions as models in propositional belief revision [16] or merging [18]). Then, the generation step consists in mapping the revised (or merged) extensions with an AF or a set of AFs that realizes them. While some generation operators have been defined from a theoretical point of view, there has been no proposal of an algorithmic approach that would compute them.

In this paper, we study the *realization problem*, *i.e.* instead of answering the question "Can this set of extensions be realized?", we produce an AF (or a set of AFs) which realizes the given set of extensions. We propose an approach for solving the realization problem based on Quantified Boolean Formulas. More precisely, we generalize the concept of realizability by introducing two parameters: k the number of auxiliary arguments (*i.e.* those which do not belong to any extension) that may appear in the result AF(s), and m the number of AFs in the result. The question of k-m-realizability is then "Using k auxiliary arguments, can we find m AFs $\mathcal{F}_1, \dots, \mathcal{F}_m$ such that the union of their extensions is exactly equal to a given set of extensions $\mathbb{S}$?". We provide Quantified Boolean Formulas (QBFs) encodings that allow to solve this problem for some prominent semantics, and to obtain the resulting AF (or set of AFs). Then we study the question of minimal change which is of utmost importance in the application context of belief revision [7, 10] (or merging [9]). More precisely, we provide a QMaxSAT [15] variant of our encoding which guarantees that the resulting AF(s) will be as close as possible to the initial AF.

The paper is organized as follows. Section 1 describes background notions on argumentation and propositional logic (in particular, Quantified Boolean Formulas and Quantified MaxSAT). In Section 2, we introduce the generalization of realizability with two parameters: the number of auxiliary arguments and the number of AFs in the result. Section 3 describes the encoding of our new form of reasoning into QBFs, and Section 4 shows how QMaxSAT can be used to solve the optimization version. Finally, Section 5 discusses related work, and Section 6 concludes the paper.

1 Background Notions

1.1 Abstract Argumentation

Let us introduce the basic notions of abstract argumentation.

Definition 1. *An argumentation framework (AF) [11] is a directed graph $\mathcal{F} = \langle \mathcal{A}, \mathcal{R} \rangle$ where $\mathcal{A}$ is the set of arguments, and $\mathcal{R} \subseteq \mathcal{A} \times \mathcal{A}$ is the attack relation.*

Given an AF, for $a, b \in \mathcal{A}$, we say that a *attacks* b if $(a, b) \in \mathcal{R}$. Moreover, a set $S \subseteq \mathcal{A}$ *defends* an argument $c \in \mathcal{A}$ if, $\forall b \in \mathcal{A}$ s.t. $(b, c) \in \mathcal{R}$, $\exists a \in S$ s.t. $(a, b) \in \mathcal{R}$. Different notions of collective acceptance of arguments are defined by Dung, based on the notion of *extension*. An extension semantics is a function σ that maps an AF $\mathcal{F} = \langle \mathcal{A}, \mathcal{R} \rangle$ to its set of extensions $\sigma(\mathcal{F}) \in 2^{\mathcal{A}}$. Most semantics are based on two simple notions: a set $S \subseteq \mathcal{A}$ is

– *conflict-free* iff $\forall a, b \in S$, $(a, b) \notin \mathcal{R}$;
– *admissible* iff S is conflict-free and S defends all its elements.

We only introduce the extension semantics that are used in this work:

Definition 2. *Given $\mathcal{F} = \langle \mathcal{A}, \mathcal{R} \rangle$ an AF, the set of arguments $S \subseteq \mathcal{A}$ is*

– *a* stable extension *iff S is conflict-free and $\forall b \in \mathcal{A} \setminus S$, $\exists a \in S$ s.t. $(a, b) \in \mathcal{R}$;*
– *a* complete extension *iff S is admissible and $\forall a \in \mathcal{A}$ that is defended by S, $a \in S$.*

Given an AF $\mathcal{F}$, we use $\mathsf{cf}(\mathcal{F})$, $\mathsf{ad}(\mathcal{F})$, $\mathsf{co}(\mathcal{F})$ and $\mathsf{st}(\mathcal{F})$ to denote (respectively) the conflict-free sets, the admissible sets, the complete extensions and the stable extensions of $\mathcal{F}$. Although we focus on these semantics, let us also mention the preferred extensions $\mathsf{pr}(\mathcal{F})$ which are the $\subseteq$-maximal complete extensions, and the (unique) grounded extension $\mathsf{gr}(\mathcal{F})$ which is the $\subseteq$-minimal complete extension. We refer the interested reader to [2] for a more detailed overview of extension semantics.

Example 1. The AF $\mathcal{F}$ depicted at Figure 1 admits a single complete (and stable) extension: $\mathsf{co}(\mathcal{F}) = \mathsf{st}(\mathcal{F}) = \{\{a, c, e\}\}$.

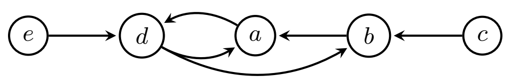


Fig. 1: $\mathcal{F}$ from Example 1

The notion of realizability of a set of extensions is defined as follows.

Definition 3. *Given $\mathcal{A}$ a set of arguments and σ a semantics, the set $\mathbb{S} \subseteq 2^{\mathcal{A}}$ is σ-realizable [12] (or just realizable, if σ is clear from the context) iff there is an AF $\mathcal{F} = \langle \mathcal{A}, \mathcal{R} \rangle$ such that $\sigma(\mathcal{F}) = \mathbb{S}$.*

Moreover, we say that $\mathbb{S}$ is compactly σ-realizable [3] iff there is a compact AF $\mathcal{F} = \langle \mathcal{A}, \mathcal{R} \rangle$ such that $\sigma(\mathcal{F}) = \mathbb{S}$, i.e. $(\bigcup_{E \in \sigma(\mathcal{F})} E) = \mathcal{A}$ *(or, with words, each argument in F appears in at least one extension).*

We can easily give examples of (non-)realizable sets.

Example 2. Let $\mathbb{S}_1 = \{\{a, b\}, \{a, c\}\}$ be a set of extensions. The AF $\mathcal{F}_1$ given at Figure 2 realizes $\mathbb{S}_1$ with respect to the stable semantics (*i.e.* $\mathsf{st}(\mathcal{F}_1) = \mathbb{S}_1$).

It is also easy to exhibit a set of extensions that is not realizable w.r.t. the stable semantics. Let $\mathbb{S}_2 = \{\{a, b\}, \{a, c\}, \{b, c\}\}$ be a set of extensions. We suppose that $\mathbb{S}_2$ is st-realizable. In that case, let $\mathcal{F}_2 = \langle \mathcal{A}_2, \mathcal{R}_2 \rangle$ be an AF with $\mathsf{st}(\mathcal{F}_2) = \mathbb{S}_2$ and $\{a, b, c\} \subseteq \mathcal{A}_2$. By definition of the stable semantics, if $\{a, b\}$ is a stable extension then each argument in $\mathcal{A}_2 \setminus \{a, b\}$ is attacked by a or b, including c. If $(a, c) \in \mathcal{R}_2$, we have a contradiction with the fact that $\{a, c\}$ is a stable extension. Similarly if $(b, c) \in \mathcal{R}_2$, then $\{b, c\}$ cannot be a stable extension. So we conclude that $\mathcal{F}_2$ does not exist, and $\mathbb{S}_2$ is not st-realizable.

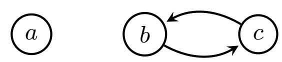


Fig. 2: $\mathcal{F}_2$ from Example 2

1.2 Propositional Logic, Quantified Boolean Formulas and Quantified MaxSAT

Now we recall some basic notions of propositional logic. A propositional formula is built on a set of Boolean variables V, *i.e.* variables that can be assigned a (truth) value in $\mathbb{B} = \{0, 1\}$, where 0 is interpreted as *false*, and 1 as *true*. A well-formed propositional formula is either an atomic formula (*i.e.* simply a Boolean variable), or built with connectives following the recursive definition:

- negation: if ϕ is a formula, then $\neg\phi$ is a formula;
- conjunction: if ϕ, ψ are formulas, then $\phi \wedge \psi$ is a formula;
- disjunction: if ϕ, ψ are formulas, then $\phi \vee \psi$ is a formula;
- implication: if ϕ, ψ are formulas, then $\phi \to \psi$ is a formula;
- equivalence: if ϕ, ψ are formulas, then $\phi \leftrightarrow \psi$ is a formula.

The semantics of propositional formulas is defined with interpretations, *i.e.* mappings $\omega : V \to \mathbb{B}$, that can be extended to arbitrary formulas:

- $\omega(\neg\phi) = 1 - \omega(\phi)$;
- $\omega(\phi \wedge \psi) = \min(\omega(\phi), \omega(\psi))$;
- $\omega(\phi \vee \psi) = \max(\omega(\phi), \omega(\psi))$;
- $\omega(\phi \to \psi) = \omega(\neg\phi \vee \psi)$;
- $\omega(\phi \leftrightarrow \psi) = \omega((\phi \to \psi) \wedge (\psi \to \phi))$.

Some normal forms are defined based on these notions: a *literal* is either an atomic formula or the negation of an atomic formula, a *clause* is a disjunction of literals, and a *cube* is a conjunction of literals. A propositional formula is a CNF (Conjunctive Normal Form) if it is a conjunction of clauses. CNF formulas can also be represented as sets of clauses. A DNF (Disjunctive Normal Form) is a disjunction of cubes.

Now, let us introduce Quantified Boolean Formulas (QBFs). QBFs are a natural extension of propositional formulas with two quantifiers: $\forall$ (universal quantifier) and $\exists$ (existential quantifier). Any propositional formula is a particular QBF. Then, if Φ is a QBF, then for $x \in V$, $\exists x, \Phi$ and $\forall x, \Phi$ are well formed QBFs as well. $\exists x, \Phi$ is true if it is possible to assign a truth value to x such that Φ is true, and $\forall x, \Phi$ is true if Φ is true for both possible truth values of x. If $\mathbf{Q}x, \mathbf{Q}y, \Phi$ is a QBF where $\mathbf{Q}$ is either $\exists$ or $\forall$, then we simply write $\mathbf{Q}\{x, y\}, \Phi$.

In the rest of the paper, we focus on *prenex QBFs*, which are QBFs that are written $\mathbf{Q}_1 V_1, \mathbf{Q}_2 V_2 \ldots, \mathbf{Q}_n V_n, \phi$ where:

- $V_1, \ldots, V_n$ are disjoint sets of Boolean variables such that $V_1 \cup \cdots \cup V_n = V$;
- $\forall i \in \{1, \ldots, n\}$, $\mathbf{Q}_i \in \{\forall, \exists\}$ is a quantifier;
- $\forall i \in \{1, \ldots, n-1\}$, $\mathbf{Q}_i \neq \mathbf{Q}_{i+1}$ (*i.e.* quantifiers are alternated);

– ϕ is a propositional formula called the *matrix* of the QBF.

We write $\overrightarrow{\mathbf{Q}}\phi$ for any prenex QBF, *i.e.* $\overrightarrow{\mathbf{Q}}$ is a shorthand for $\mathbf{Q}_1 V_1, \mathbf{Q}_2 V_2 \ldots, \mathbf{Q}_n V_n$.

Finally, we introduce an optimization problem related to QBFs, which is a generalization of MaxSAT [21] to quantified formulas. Consider a formula $\overrightarrow{\mathbf{Q}}\phi_H \wedge \phi_S$ where the matrix is the conjunction of two types of constraints: the *hard constraints* ϕ_H and the *soft constraints* ϕ_S represented as a CNF formula (*i.e.* a set of clauses). Then QMaxSAT (Quantified MaxSAT) [15] is the problem consisting in finding the largest possible subset of clauses $\phi_S^* \subseteq \phi_S$ such that $\overrightarrow{\mathbf{Q}}\phi_H \wedge \phi_S^*$ is true.

Example 3. The QBF formula $\exists\{x, y\} x \vee y$ is true: there is at least one truth value for x and y such that $x \vee y$ is true (for instance, the interpretation $\omega(x) = 1$ and $\omega(y) = 0$). Now, if one needs to obtain an interpretation of the variables which makes the formula true such that its cardinality is maximal, one can transform the QBF formula into an instance of QMaxSAT: $\exists\{x, y\}(x \vee y) \wedge \phi_S$, where the soft constraints are given by $\phi_S = x \wedge y$.[1] The interpretation of the variables $\{x, y\}$ which satisfies $x \vee y$ and maximizes the number of satisfied (soft) unit clauses is ω' such that $\omega'(x) = \omega'(y) = 1$, which satisfies both soft clauses.

2 Generalizing Realizability

Now we define our new types of realizability, where two natural numbers are given as parameters, k and m representing respectively the number of auxiliary arguments (*i.e.* those which do not appear in any extension) and the number of AFs in the result. We have described previously the interest of this new approach for representing the result of AF revision or merging operators [7, 9]. Both parameters k and m are necessary in this case, since it is possible that some arguments from the initial AF do not appear in any extension of the result (then, k is the number of these arguments) and not any set of extensions can be represented by a single AF (which explains the need for the parameter m). We can also think to a variant of the *rationalisation* problem [1] where agents provide (sets of) extensions instead of AFs, *i.e.* we assume a scenario where the set of arguments involved in a debate is known (but not the full processing of a debate, *i.e.* the attack relation), and several agents provide their opinion about the acceptability of arguments. Realization is a means of constructing possible representations of the debate.

We start this section with the special case where $m = 1$, thus defining k-realizablity in Section 2.1. The more general k-m-realizability is introduced in Section 2.2

[1] Observe that ϕ_S is the conjunction of two unit clauses, x on the one hand, and y on the other hand.

2.1 k-Realizability

In the literature, two types of realizability have been defined. Either the set of extensions $\mathbb{S}$ is realizable with a compact AF (*i.e.* each argument that appears in the AF belongs to at least one extension), or it is "simply" realizable (*i.e.* some arguments may not appear in any extension). We define a variant of realizability that takes into account the exact number of arguments that appear in the AF but do not appear in any extension.

Definition 4 (k-Realizability). *Given $\mathcal{A}$ a set of arguments, $k \in \mathbb{N}$, a semantics σ, and $\mathbb{S} \subseteq 2^{\mathcal{A}}$ a set of extensions s.t. $\bigcup_{E \in \mathbb{S}} E = \mathcal{A}$, we say that $\mathbb{S}$ is σ-k-realizable if there is an AF $\mathcal{F} = \langle \mathbf{A}, \mathcal{R} \rangle$ s.t. $\sigma(\mathcal{F}) = \mathbb{S}$, with $\mathbf{A} = \mathcal{A} \cup \mathcal{A}'$, where $\mathcal{A} \cap \mathcal{A}' = \emptyset$ and $|\mathcal{A}'| = k$.*

We drop σ from the notation when it is clear from the context. Obviously, compact realizability corresponds to 0-realizability, while the question "Is $\mathbb{S}$ realizable?" is equivalent to "Is there some $k \in \mathbb{N}$ such that $\mathbb{S}$ is k-realizable?".

Example 4. To show the importance of fixing a good value for k, we borrow an example from [3]. Consider the sets of arguments $A = \{a_1, a_2, a_3\}$, $B = \{b_1, b_2\}$ and $C = \{c_1, c_2, c_3\}$. We focus on the set of extensions $\mathbb{S} = \{\{a_i, b_j, c_k\} \mid i \in \{1,2,3\}, j \in \{1,2\}, k \in \{1,2,3\}\} \setminus \{a_1, b_1, c_2\}$, *i.e.* each extension contains one of the a_i arguments, one of the b_j, and one of the c_k, but the combination $\{a_1, b_1, c_2\}$ is forbidden. We have $\mathcal{A} = A \cup B \cup C$. [3] proves that the set of extensions $\mathbb{S}$ cannot be compactly realized under the stable semantics, *i.e.* it is not st-0-realizable. On the contrary, it is st-1-realizable, choosing for instance $\mathcal{A}' = \{z\}$. See Figure 3 for an example of $\mathcal{F} = \langle \mathcal{A} \cup \mathcal{A}', \mathcal{R} \rangle$ such that $\mathsf{st}(\mathcal{F}) = \mathbb{S}$.

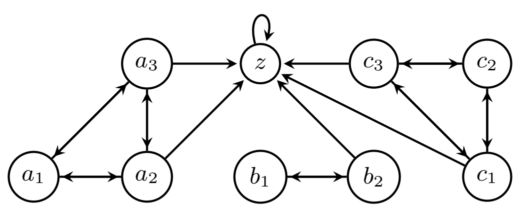


Fig. 3: An AF that st-1-realizes $\mathbb{S}$

Fixing the set of auxiliary arguments seems reasonable in situations where one already knows all the possible arguments at hand (the set $\mathbf{A}$), and the result of the argumentative process (*i.e.* the extensions, and then $\mathcal{A}$ which is the union of the extensions), but one needs to find the relations between arguments (*i.e.* the structure of the graph) that would explain the arguments acceptability. In this case, it is not possible to add any number of auxiliary arguments, but only $k = |\mathbf{A} \setminus \mathcal{A}|$. Moreover, from a technical point of view, fixing k allows to determine the number of Boolean variables required to encode the problem in logic, as described in Section 3.

2.2 k-m-Realizability

Now, we focus on how to realize a set of extensions with a set of AFs. Indeed, if the set $\mathbb{S}$ cannot be realized by a single AF, whatever the number k of arguments, we need several AFs to do it, as it is the case when AFs are revised [7] or merged [9]. So we introduce a second parameter, m, that represents the number of AFs that are used to realize the set of extensions. This means that the goal is now to obtain a set of m AFs such that the union of their extensions corresponds to the given set $\mathbb{S}$.

Definition 5 (k-m-Realizability). *Given $\mathcal{A}$ a set of arguments, $k, m \in \mathbb{N}$ with $m > 0$, a semantics σ, and $\mathbb{S} \subseteq 2^{\mathcal{A}}$ a set of extensions s.t. $\bigcup_{E \in \mathbb{S}} E = \mathcal{A}$, we say that $\mathbb{S}$ is σ-k-m-realizable if there exists a set of AFs $\mathbb{F} = \{\mathcal{F}_1 = \langle \mathbf{A}, \mathcal{R}_1 \rangle, \dots, \mathcal{F}_m = \langle \mathbf{A}, \mathcal{R}_m \rangle\}$ such that $(\bigcup_{\mathcal{F} \in \mathbb{F}} \sigma(\mathcal{F})) = \mathbb{S}$, with $\mathbf{A} = \mathcal{A} \cup \mathcal{A}'$, where $\mathcal{A} \cap \mathcal{A}' = \emptyset$ and $|\mathcal{A}'| = k$.*

Again, we drop σ from the notation when it is clear from the context. The generation operators from [7, 9] are a special case of 0-m-realizability.

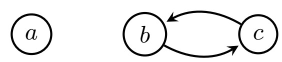
Notice that any set of extensions $\mathbb{S}$ is 0-m-realizable under most semantics when $m = |\mathbb{S}|$. If $\emptyset \notin \mathbb{S}$, we can define $\mathbb{F} = \{\mathcal{F}_1, \dots, \mathcal{F}_m\}$ such that, in $\mathcal{F}_i$, the arguments in the extension $E_i \in \mathbb{S}$ are unattacked, and the arguments that do not appear in E_i are attacked by some argument in E_i. Then, $\sigma(\mathcal{F}_i) = E_i$ and thus $\mathbb{F}$ 0-m-realizes $\mathbb{S}$. If $\emptyset \in \mathbb{S}$, this particular extension can be realized under most semantics by an AF where all the arguments are self-attacking. Among the main semantics studied in the literature, only the stable semantics cannot realize $\emptyset$, thus $\mathbb{S}$ is 0-m-realizable under the stable semantics when $\emptyset \notin \mathbb{S}$.[2] While this is a proof that $\mathbb{S}$ is 0-m-realizable, it does not mean that the set $\mathbb{F}$ is an adequate solution in any situation. For instance, in a context of AF revision [7], it is unlikely that the $\mathcal{F}_i$ AFs will be related to the initial AF, and then this result $\mathbb{F}$ would not comply with the minimal change principle. Also, we show here that $m = |\mathbb{S}|$ is only an upper bound, but $\mathbb{S}$ may be realizable with only m' AFs (where $m' < |\mathbb{S}|$).

This discussion can be summarized by Proposition 1.

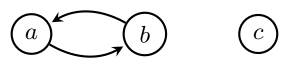
Proposition 1. *Let $\mathbb{S}$ be a set of extensions. $\mathbb{S}$ is 0-$|\mathbb{S}|$-realizable under $\sigma \in \{\mathsf{co}, \mathsf{gr}, \mathsf{pr}\}$. Moreover, if $\emptyset \notin \mathbb{S}$, then $\mathbb{S}$ is 0-$|\mathbb{S}|$-realizable under $\sigma = \mathsf{st}$.*

Example 5. Consider again $\mathbb{S}_2 = \{\{a, b\}, \{a, c\}, \{b, c\}\}$ from Example 2. As stated previously, it cannot be realized by a single AF under the stable semantics, *i.e.* it is not st-0-1-realizable (neither st-k-1-realizable with any $k \in \mathbb{N}$). However, it is st-0-2-realizable, as can be seen with $\mathbb{F} = \{\mathcal{F}_1, \mathcal{F}_2\}$ from Figure 4. We have $\mathsf{st}(\mathcal{F}_1) = \{\{a, b\}, \{a, c\}\}$ and $\mathsf{st}(\mathcal{F}_2) = \{\{a, c\}, \{b, c\}\}$, hence $\bigcup_{\mathcal{F} \in \mathbb{F}} \mathsf{st}(\mathcal{F}) = \mathbb{S}_2$.

[2] Formally, if we allow empty AFs, then $\mathcal{F}_\emptyset = \langle \emptyset, \emptyset \rangle$ realizes the empty set under the stable semantics. However authorizing such an empty AF means that $\mathbf{A} = \emptyset$, and then no other extension can be realized.



(a) $\mathcal{F}_1$



(b) $\mathcal{F}_2$

Fig. 4: $\mathbb{F} = \{\mathcal{F}_1, \mathcal{F}_2\}$ st-0-2-realizes $\mathbb{S}_2$

3 k-m-Realizability as QBF Solving

Now we propose a QBF-based approach for solving k-m-realizability, *i.e.*

- determining whether a set of extensions $\mathbb{S}$ is k-m-realizable,
- and providing a set of AFs which realizes $\mathbb{S}$, when it exists.

We start with the simpler case of k-realizability in Section 3.1, and then we explain how we generalize the encoding to represent the set of m AFs in Section 3.2.

3.1 Encoding k-realizability with QBFs

We suppose that we know $\mathcal{A}$ the set of all the arguments that appear in the extensions, and $\mathcal{A}'$ (with $|\mathcal{A}'| = k$) the set of arguments that do not appear in any extension but appear in the argumentation framework(s). Then, let $\mathbf{A} = \mathcal{A} \cup \mathcal{A}'$ be the set of all the arguments. We will define propositional formulas such that each model represents a set of arguments $X \subseteq \mathbf{A}$ and attacks in $\mathbf{A} \times \mathbf{A}$. The approach is inspired by [4]. Our goal is to write these formulas directly as sets of clauses, in order to be able to feed a QBF solver with them without a (possibly) expensive translation from an arbitrary formula into a CNF formula. We define the following Boolean variables:

- for $a \in \mathbf{A}$, in_a is true iff a is in the set of arguments of interest;
- for $a, b \in \mathbf{A}$, $att_{a,b}$ is true iff a attacks b.

Conflict-freeness We define ϕ_{cf} that represents the conflict-free sets in a classical way:

$$\phi_{\mathsf{cf}} = \bigwedge_{a \in \mathbf{A}, b \in \mathbf{A}} (att_{a,b} \rightarrow \neg in_a \vee \neg in_b)$$

Stable Semantics Now, we focus on the stable semantics. Let us recall that a stable extension is a conflict-free set $X \subseteq \mathbf{A}$ that attacks every argument in $\mathbf{A} \setminus X$, *i.e.* any argument is either a member of X, or attacked by a member of X. Thus, the stable semantics can be encoded by the formula $\widehat{\phi}_{\mathsf{st}}$:

$$\widehat{\phi}_{\mathsf{st}} = \phi_{\mathsf{cf}} \wedge (\bigwedge_{a \in \mathbf{A}} (in_a \vee \bigvee_{b \in \mathbf{A}} (in_b \wedge att_{b,a})))$$

The second part of the formula expresses that each argument a is either in the set (in_a), or attacked by an argument b in the set ($in_b \wedge att_{b,a}$). This enforces the set $\{a \in \mathbf{A} \mid in_a = 1\}$ as a stable extension of the AF $\mathcal{F} = \langle \mathbf{A}, \mathcal{R} \rangle$, with $\mathcal{R} = \{(a, b) \in \mathbf{A} \times \mathbf{A} \mid att_{a,b} = 1\}$. We need to transform the second part into a set of clauses. For facilitating this transformation, we introduce new variables. We remark that for each $a \in \mathbf{A}$,

$$in_a \vee \bigvee_{b \in \mathbf{A}} (in_b \wedge att_{b,a}) \equiv in_a \vee \bigvee_{b \in \mathbf{A}} det_{b,a}$$

where $det_{b,a}$ is a newly introduced variable that means that b *defeats* a (*i.e.* b is accepted and b attacks a). We formally encode the meaning of these variables by:

$$\phi_{det} = \bigwedge_{a \in \mathbf{A}, b \in \mathbf{A}} det_{a,b} \leftrightarrow (in_a \wedge att_{a,b})$$

So now, we define ϕ_{st}:

$$\phi_{\mathsf{st}} = \phi_{\mathsf{cf}} \wedge (\bigwedge_{a \in \mathbf{A}} (in_a \vee \bigvee_{b \in \mathbf{A}} det_{b,a})) \wedge \phi_{det}$$

This formula is equi-satisfiable with $\widehat{\phi_{\mathsf{st}}}$. Moreover, the models of ϕ_{st} can be bijectively associated with the models of $\widehat{\phi_{\mathsf{st}}}$, since the values of the det-variables are completely determined by the values of the in and att-variables. Now, to express ϕ_{st} as a CNF formula, we need to rewrite ϕ_{det} as an equivalent set of clauses. This is done as follows:

$$\begin{aligned} det_{a,b} \leftrightarrow (in_a \wedge att_{a,b}) \equiv\ & (\neg det_{a,b} \vee in_a) \\ & \wedge (\neg det_{a,b} \vee att_{a,b}) \\ & \wedge (\neg in_a \vee \neg att_{a,b} \vee det_{a,b}) \end{aligned}$$

Admissibility For encoding admissibility, we need to express that an argument is defended by the set of arguments which is characterized. Thus we introduce a new kind of variable, for each $a \in \mathbf{A}$, def_a means that a is defended. Formally, this is encoded by

$$\phi_{def} = \bigwedge_{a \in \mathbf{A}} (def_a \leftrightarrow \bigwedge_{b \in \mathbf{A}} (att_{b,a} \rightarrow \bigvee_{c \in \mathbf{A}} det_{c,b}))$$

This means that admissible sets can be encoded by

$$\phi_{\mathsf{ad}} = \phi_{\mathsf{cf}} \wedge \phi_{det} \wedge \phi_{def} \wedge \bigwedge_{a \in \mathbf{A}} in_a \rightarrow def_a$$

Complete Semantics Finally, since a complete extension is an admissible set which contains exactly what it defends, we can encode the complete semantics by

$$\phi_{\mathsf{co}} = \phi_{\mathsf{ad}} \wedge \bigwedge_{a \in \mathbf{A}} def_a \rightarrow in_a$$

Again, classical transformations allow to obtain ϕ_{ad} and ϕ_{co} as CNF formulas.

Encoding k-Realizability For $\mathcal{A}$ a set of arguments, we suppose that $\mathbb{S} = \{E_1, \dots, E_n\}$, such that $E_i \subseteq \mathcal{A}$ for each $i \in \{1, \dots, n\}$, is the set of extensions to be realized. The approach is generic: for any semantics σ, we suppose that ϕ_σ is the propositional formula that encodes the relationship between an AF and its extensions, for the semantics σ. It works for any semantics such that reasoning is at most at the first level of the polynomial hierarchy, which can thus be polynomially encoded into propositional logic. So in the rest of the section, $\sigma \in \{\mathsf{cf}, \mathsf{ad}, \mathsf{st}, \mathsf{co}\}$. We need to encode $\mathbb{S}$ into a propositional formula as well:

$$\phi_{\mathbb{S}} = \bigvee_{E_i \in \mathbb{S}} \phi_{E_i} \text{ with } \phi_{E_i} = \bigwedge_{a \in E_i} in_a \wedge \bigwedge_{a \in \mathbf{A} \setminus E_i} \neg in_a$$

The fact that the extensions of the AF must correspond to the extensions in $\mathbb{S}$ can be encoded as an equivalence:

$$\widehat{\phi_\sigma^{\mathbb{S}}} = \phi_\sigma \leftrightarrow \phi_{\mathbb{S}}$$

In order to facilitate the transformation of this formula into a CNF, we introduce two new variables, x_σ and $x_{\mathbb{S}}$, such that $x_\sigma \leftrightarrow \phi_\sigma$ and $x_{\mathbb{S}} \leftrightarrow \phi_{\mathbb{S}}$. So we obtain

$$\phi_\sigma^{\mathbb{S}} = (x_\sigma \leftrightarrow x_{\mathbb{S}}) \wedge (x_\sigma \leftrightarrow \phi_\sigma) \wedge (x_{\mathbb{S}} \leftrightarrow \phi_{\mathbb{S}})$$

The different parts of this formula can be easily written as sets of clauses, either by standard manipulations of the formula, or (in the case where a DNF appears because of the standard manipulations) by a simple application of the Tseytin transformation method [32].

Finally, the set $\mathbb{S}$ is σ-k-realizable if there is a valuation of the att-variables such that each possible valuation of the in-variables satisfies $\phi_\sigma^{\mathbb{S}}$, or said otherwise if the QBF

$$\exists ATT, \forall IN, \phi_\sigma^{\mathbb{S}}$$

is valid, where $ATT = \{att_{a,b} \mid a, b \in \mathbf{A}\}$ and $IN = \{in_a \mid a \in \mathbf{A}\}$. In order to obtain a fully defined prenex QBF that can be given as input to a QBF solver, let us remark that the det-variables, the def-variables, x_σ, $x_{\mathbb{S}}$ and the variables introduced by the Tseytin transformation must be existentially quantified at the third level of the QBF. The truth values assigned to the ATT variables can be obtained from a QBF solver (*e.g.* CAQE [29]), providing $\omega : ATT \rightarrow \{0, 1\}$. The AF $\mathcal{F}$ that realizes $\mathbb{S}$ is then defined by $\mathcal{F} = \langle \mathbf{A}, \mathcal{R} \rangle$, with $\mathcal{R} = \{(a, b) \mid \omega(att_{a,b}) = 1\}$.

3.2 Encoding *k*-*m*-realizability with QBFs

Now, to encode k-m-realizability instead of k-realizability, we need to introduce variables that will represent the structure of the m AFs that realize $\mathbb{S}$. For $a, b \in \mathbf{A}$ and $i \in \{1, \dots, m\}$, $att^i_{a,b}$ is true iff a attacks b in $\mathcal{F}_i$ and $det^i_{a,b}$ is true iff a defeats b in $\mathcal{F}_i$ (*i.e.* $det^i_{a,b} \leftrightarrow in_a \wedge att^i_{a,b}$). Then, we call ϕ^i_σ the formula ϕ_σ where each att or det variable is replaced by the corresponding att^i or det^i.

To represent the fact that the union of the extensions of $\mathbb{F} = \{\mathcal{F}_1, \dots, \mathcal{F}_m\}$ corresponds to $\mathbb{S}$, we write:

$$\widehat{\phi_\sigma^{m,\mathbb{S}}} = (\bigvee_{i \in \{1,\dots,m\}} \phi_\sigma^i) \leftrightarrow \phi_\mathbb{S}$$

In order to write this formula as a CNF, we apply the same technique as for k-realizability. For each $i \in \{1, \dots, m\}$, we introduce a variable x_σ^i, and we consider the formula $x_\sigma^i \leftrightarrow \phi_\sigma^i$, that can be easily transformed into a set of clauses. Thus, we can replace $\widehat{\phi_\sigma^{m,\mathbb{S}}}$ by the CNF

$$\phi_\sigma^{m,\mathbb{S}} = ((\bigvee_{i \in \{1,\dots,m\}} x_\sigma^i) \leftrightarrow x_\mathbb{S}) \wedge (\bigwedge_{i \in \{1,\dots,m\}} (x_\sigma^i \leftrightarrow \phi_\sigma^i)) \wedge (x_\mathbb{S} \leftrightarrow \phi_\mathbb{S})$$

where the first part of the formula is equivalent to the set of clauses $(\bigwedge_{i \in \{1,\dots,m\}} (\neg x_\sigma^i \vee x_\mathbb{S})) \wedge (\neg x_\mathbb{S} \vee x_\sigma^1 \vee \dots \vee x_\sigma^m)$. Finally, we encode the k-m-realizability with a QBF

$$\exists ATT, \forall IN, \phi_\sigma^{m,\mathbb{S}}$$

where $ATT = \{att_{a,b}^i \mid a, b \in \mathbf{A}, i \in \{1, \dots, m\}\}$ and $IN = \{in_a^i \mid a \in \mathbf{A}, i \in \{1, \dots, m\}\}$. Then each $\mathcal{F}_i$ can be obtained, as previously, from the values of the att^i-variables provided by the QBF solver.

4 Optimal k-m-Realization

Now we suppose that the realization process is guided by a minimal change principle, *i.e.* there is an input AF $\mathcal{F}^* = \langle \mathcal{A}^*, \mathcal{R}^* \rangle$, and the AFs produced must be as close as possible to $\mathcal{F}^*$. This is, for instance, an important feature of belief revision operators [7].

Before introducing optimal realization, we introduce the tools required to quantify the closeness between AFs.

Definition 6. *The Hamming distance between two AFs $\mathcal{F}_1 = \langle \mathcal{A}, \mathcal{R}_1 \rangle$ and $\mathcal{F}_2 = \langle \mathcal{A}, \mathcal{R}_2 \rangle$ is $d_H(\mathcal{F}_1, \mathcal{F}_2) = |(\mathcal{R}_1 \setminus \mathcal{R}_2) \cup (\mathcal{R}_2 \setminus \mathcal{R}_1)|$.*

The distance d_H simply counts the number of attacks which differ between two AFs. To quantify the closeness between an AF and a set of AFs (the result of the realization), we sum these distances:

Definition 7. *Given $\mathcal{F} = \langle \mathcal{A}, \mathcal{R} \rangle$ and $\mathbb{F} = \{\mathcal{F}_1 = \langle \mathcal{A}, \mathcal{R}_1 \rangle, \dots, \mathcal{F}_m = \langle \mathcal{A}, \mathcal{R}_m \rangle\}$, we define $d_H^\sum$ by $d_H^\sum(\mathcal{F}, \mathbb{F}) = \sum_{\mathcal{F}_i \in \mathbb{F}} d_H(\mathcal{F}, \mathcal{F}_i)$.*

Definition 8 (Optimal k-m-Realization). *Given $\mathcal{F}^* = \langle \mathcal{A}^*, \mathcal{R}^* \rangle$ an AF, $\mathcal{A}$ a set of arguments, $k, m \in \mathbb{N}$ with $m > 0$, a semantics σ, and $\mathbb{S} \subseteq 2^\mathcal{A}$ a set of extensions s.t. $\bigcup_{E \in \mathbb{S}} E = \mathcal{A}$, we say that $\mathbb{S}$ is* optimally σ-k-m-realized *by $\mathbb{F} = \{\mathcal{F}_1 = \langle \mathcal{A}^*, \mathcal{R}_1 \rangle, \dots, \mathcal{F}_m = \langle \mathcal{A}^*, \mathcal{R}_m \rangle\}$ with $\mathcal{A}^* = \mathcal{A} \cup \mathcal{A}'$, where $\mathcal{A} \cap \mathcal{A}' = \emptyset$ and $|\mathcal{A}'| = k$, if*

- $(\bigcup_{\mathcal{F} \in \mathbb{F}} \sigma(\mathcal{F})) = \mathbb{S}$, *and*
- *for any* $\mathbb{F}'$ *satisfying the conditions above,* $d_H^{\sum}(\mathcal{F}^*, \mathbb{F}) \leq d_H^{\sum}(\mathcal{F}^*, \mathbb{F}')$.

Now we show how to adapt the QBF-based approach for k-m-realizability into a QMaxSAT approach for optimal k-m-realization. Assume that $\overrightarrow{\mathbf{Q}}\phi_\sigma^{k,m,\mathbb{S}}$ is the formula allowing to determine the k-m-realizability of $\mathbb{S}$ under the semantics σ (as described in the previous section). We introduce new variables that describe the attack relation of the initial AF $\mathcal{F}^* = \langle \mathcal{A}^*, \mathcal{R}^* \rangle$: for each pair of arguments $(a,b) \in \mathcal{A}^* \times \mathcal{A}^*$, $att^*_{a,b}$ means that $(a,b) \in \mathcal{R}^*$.

Then, for every $i \in \{1, \dots, m\}$, and every pair of arguments $(a,b) \in \mathcal{A}^* \times \mathcal{A}^*$, we introduce the variable $nd^i_{a,b}$ which means that there is no difference between the existence of the attack (a,b) in $\mathcal{F}^*$ and $\mathcal{F}_i$. This is formally characterized by the formula

$$\psi^i_{a,b} = nd^i_{a,b} \leftrightarrow (att^*_{a,b} \leftrightarrow att^i_{a,b})$$

(which can be easily transformed into a set of four clauses made of three literals each).

Finally, we need a way to represent the attack relation in the initial AF $\mathcal{F}^*$. To do that, we define

$$\theta(\mathcal{F}^*) = \bigwedge_{(a,b) \in \mathcal{R}^*} att^*_{a,b} \wedge \bigwedge_{(a,b) \in (\mathcal{A}^* \times \mathcal{A}^*) \setminus \mathcal{R}^*} \neg att^*_{a,b}.$$

Now, we can define the QMaxSAT instance $\overrightarrow{\mathbf{Q}}\phi_H \wedge \phi_S$ where the hard constraints are

$$\phi_H = \phi_\sigma^{k,m,\mathbb{S}} \wedge (\bigwedge_{i=1}^{m} \bigwedge_{(a,b) \in \mathcal{A}^* \times \mathcal{A}^*} \psi^i_{a,b}) \wedge \theta(\mathcal{F}^*)$$

and the soft constraints are

$$\phi_S = \bigwedge_{i=1}^{m} \bigwedge_{(a,b) \in \mathcal{A}^* \times \mathcal{A}^*} nd^i_{a,b}.$$

An optimal solution of this QMaxSAT instance can be decoded into a set of AFs $\mathbb{F} = \{\mathcal{F}_1, \dots, \mathcal{F}_m\}$ which optimally realizes the set of extensions $\mathbb{S}$. This optimal solution can be obtained thanks to dedicated algorithms like the ones from [15, Section 4.1].

Observe that the generation operator $\mathcal{AF}_\sigma^{card,AF}$ from [7] (which searches a set of AFs that minimizes the cardinality of the result, and then the distances between graphs as a tie-breaker) can be computed by iteratively solving the QMaxSAT encoding for optimal k-m-realization, with m varying from 1 to $|\mathbb{S}|$. In the case of the operator $\mathcal{AF}_\sigma^{dg,AF}$ (which minimizes the distances between graphs, and then the cardinality to break ties) we can solve k-m-realizability for every $m \in \{1, \dots, |\mathbb{S}|\}$, select the sets of AFs which minimize the distance, and then (in case of ties) choose the one such that m is minimal.

The approach for AF revision defined in [10] guarantees that the revised set of extensions is (classically) realizable. This means that the resulting AF can be obtained by solving (optimal) k-1-realization.

The generation operators for AF merging [9] cannot be computed with our approach, since they require to compare a set of AFs with another set of AFs (while here, we only compare one AF with a set of AFs, see Definition 7). Adapting our approach to AF merging generation operators is left for future work.

5 Discussion

The initial work on extension realizability [12] defines the concept of canonical AF, *i.e.* a specific AF that realizes a set of extensions $\mathbb{S}$ if this set is realizable. This canonical AF is useful for proving the existence of some AF (*i.e.* answering the question "Is $\mathbb{S}$ realizable?"), but this construction may not be sensible for concrete applications. Especially, when extension realization is used in a context of AF revision or merging [7, 10, 9], the canonical AF that realizes the revised/merged extensions may not be a desirable outcome in general, since it can be completely unrelated with the initial AF(s), contrary to the result of optimal k-m-realization as described in Section 4.

The UNREAL system[3] [22] allows deciding realizability for Abstract Dialectical Frameworks [5] and various subclasses thereof, including standard AFs. There are various differences between this approach and our work. First of all, it only considers "classical" realizability in the sense that the result of the operation is a single AF, *i.e.* it does not solve k-m-realizability with $m > 1$. This means that this approach will simply return "unsatisfiable" when the set of extensions $\mathbb{S}$ is not realizable by a single AF. Then, the system can provide one AF realizing the given set of extensions, and iterate over the (potentially exponentially large) set of AFs that solve the problem, but it cannot provide an optimal one like the QMaxSAT-based approach from Section 4 (or do to so, one would need to enumerate all the potential solutions, compute their cost and keep only the ones with the minimal cost, which is unlikely to be feasible in practice).

A problem similar to realization is studied under the name *inverse problem* [17]. However, their hypothesis is that the information about arguments acceptability is noisy, hence the use of a probabilistic approach to obtain the AF. It is not the case in the context of AF revision or merging which motivates our study.

The synthesis of AFs [26] shares a similar intuition with realization: given a set of extensions P and a set of extensions N (called respectively positive examples and negative examples, each of them being associated with a weight), the goal is to obtain an AF of minimal cost, where the cost is the sum of the weights of positive examples which are not an extension of the AF, and the weights of the negative examples which are extensions of the AF. Realization can be captured by stating $P = \mathbb{S}$, and $N = 2^{\mathcal{A}} \setminus \mathbb{S}$, which is not efficient from the point of view of space, and by assuming that all examples have a infinite weight (*i.e.* no example should be violated).

[3] https://www.dbai.tuwien.ac.at/proj/adf/unreal

The case of the grounded semantics is particular. Since there is exactly one grounded extension for any AF, a set of extensions $\mathbb{S}$ requires exactly $m = |\mathbb{S}|$ to be realized. A possible way to do it is strict extension enforcement [25] which modifies an AF in order to obtain a new one with the expected grounded extension. Performing this operation for each $E_i \in \mathbb{S}$ can provide the set $\mathbb{F} = \{\mathcal{F}_1, \dots, \mathcal{F}_m\}$ such that E_i is the grounded extension of $\mathcal{F}_i$, for each $i \in \{1, \dots, m\}$.

Recent work has shown that any set of extensions $\mathbb{S}$ can be represented by a single *Constrained Incomplete Argumentation Framework* (CIAF) [23]. Such a CIAF is based on the Incomplete AF model, where arguments and attacks can be labeled as uncertain, and reasoning is made through completions, *i.e.* a set of classical AFs. CIAFs add a constraint on the set of completions, which allow to finely select the completions that will be used for reasoning. [23] shows that any set of extensions is 0-1-realizable if such a CIAF is expected as the result, instead of (classical) AFs. Adapting our technique to generate a CIAF instead of a (set of) AF(s) is an interesting future work.

Finally, realizability has been studied in the context of ranking-based [30] or gradual semantics [28]. In the former case, the goal is to obtain an AF $\mathcal{F}$ such that applying a given ranking-based semantics on $\mathcal{F}$ produces a given ranking; it is shown that any ranking is realizable for various semantics. In the latter case, given the graph structure of a weighted AF, and an acceptability degree for each argument, one wants to obtain arguments weights such that applying a given gradual semantics to the weighted AF produces the expected acceptability degrees. In the same vein, [27] focuses also on gradual semantics of weighted AFs, but this time the arguments weights are known, and the goal is to obtain the graph structure. All the works are intuitively connected with the question of realizability, but strongly differ from our work because the notion of acceptability semantics is not based on extensions.

6 Conclusion

In this paper, we have proposed a generalization of the notion of extension realizability, with two parameters representing respectively the number of auxiliary arguments and the number of AFs in the result. We have defined a logic-based computational approach for this problem, paving the way to practical implementations based on QBF solvers. Our work also induces a computational approach for generating the result of AF revision [7]. This means that we do not only focus on *realizability* (*i.e.* answer to the question "is there a solution?"), but more generally on the issue of *realization* (*i.e.* "if there is a solution, then provide it").

This preliminary study opens several interesting research tracks. First, a natural extension of our work is to consider other semantics. In particular, the semantics that cannot be (polynomially) encoded into a propositional formula (*e.g.* the preferred semantics) may need some particular attention. At least two options can be considered: directly encoding ϕ_σ as a QBF [13], or using an iterated resolution approach (in the spirit of the CEGAR-based approaches used for extension enforcement [33] or AF synthesis [26]). We also plan to implement our approach in order to empirically evaluate its efficiency, and the influence of the various parameters (the semantics σ, the number of auxiliary arguments k, the number of AFs in the result m) on the possibility to realize the given set of extensions. While the encoding described here are constructed step by step from the logical translation of argumentation basic principles (*e.g.* conflict-freeness, defense) and semantics (*e.g.* stable, complete), on the practical side our approach can benefit from some insights provided by existing logic-based argumentation tools (*e.g* [19, 24]) in order to improve the implementation of the QBF encoding. Then, another interesting question is how to define (and encode in QBF) optimal k-m-realization when the optimality is not based on the distance between the result and one input AF, but on the distance between the result and a set of AFs, like in the case of generation operators for AF merging [9]. Finally, recent work has shown how deep learning can be used to improve the efficiency of enforcement tools [8]. Studying whether such techniques can be used in a context of realization is an appealing question for future work.

References

1. Airiau, S., Bonzon, E., Endriss, U., Maudet, N., Rossit, J.: Rationalisation of profiles of abstract argumentation frameworks: Characterisation and complexity. J. Artif. Intell. Res. **60**, 149–177 (2017). https://doi.org/10.1613/jair.5436, https://doi.org/10.1613/jair.5436
2. Baroni, P., Caminada, M., Giacomin, M.: Abstract argumentation frameworks and their semantics. In: Baroni, P., Gabbay, D., Giacomin, M., van der Torre, L. (eds.) Handbook of Formal Argumentation, pp. 159–236. College Publications (2018)
3. Baumann, R., Dvorák, W., Linsbichler, T., Strass, H., Woltran, S.: Compact argumentation frameworks. In: Proc. of ECAI'14. pp. 69–74 (2014)
4. Besnard, P., Doutre, S.: Checking the acceptability of a set of arguments. In: Proc. of NMR'04. pp. 59–64 (2004)
5. Brewka, G., Strass, H., Ellmauthaler, S., Wallner, J.P., Woltran, S.: Abstract dialectical frameworks revisited. In: Proc. of IJCAI'13. pp. 803–809 (2013), http://www.aaai.org/ocs/index.php/IJCAI/IJCAI13/paper/view/6551
6. Cerutti, F., Giacomin, M., Vallati, M.: How we designed winning algorithms for abstract argumentation and which insight we attained. Artif. Intell. **276**, 1–40 (2019)
7. Coste-Marquis, S., Konieczny, S., Mailly, J.G., Marquis, P.: On the revision of argumentation systems: Minimal change of arguments statuses. In: Proc. of KR'14 (2014)
8. Craandijk, D., Bex, F.: Enforcement heuristics for argumentation with deep reinforcement learning. In: Proc. of AAAI'22. pp. 5573–5581. AAAI Press (2022), https://ojs.aaai.org/index.php/AAAI/article/view/20497
9. Delobelle, J., Haret, A., Konieczny, S., Mailly, J.G., Rossit, J., Woltran, S.: Merging of abstract argumentation frameworks. In: Proc. of KR'16. pp. 33–42 (2016)
10. Diller, M., Haret, A., Linsbichler, T., Rümmele, S., Woltran, S.: An extension-based approach to belief revision in abstract argumentation. Int. J. Approx. Reason. **93**, 395–423 (2018)

11. Dung, P.M.: On the acceptability of arguments and its fundamental role in non-monotonic reasoning, logic programming and n-person games. Artif. Intell. **77**(2), 321–358 (1995)
12. Dunne, P.E., Dvorák, W., Linsbichler, T., Woltran, S.: Characteristics of multiple viewpoints in abstract argumentation. Artif. Intell. **228**, 153–178 (2015)
13. Egly, U., Woltran, S.: Reasoning in argumentation frameworks using quantified boolean formulas. In: Proc. of COMMA'06. pp. 133–144 (2006)
14. Gaggl, S.A., Linsbichler, T., Maratea, M., Woltran, S.: Design and results of the second international competition on computational models of argumentation. Artif. Intell. **279** (2020)
15. Ignatiev, A., Janota, M., Marques-Silva, J.: Quantified maximum satisfiability. Constraints An Int. J. **21**(2), 277–302 (2016)
16. Katsuno, H., Mendelzon, A.O.: Propositional knowledge base revision and minimal change. Artif. Intell. **52**(3), 263–294 (1992)
17. Kido, H., Liao, B.: A bayesian approach to direct and inverse abstract argumentation problems. CoRR **abs/1909.04319** (2019), http://arxiv.org/abs/1909.04319
18. Konieczny, S., Pino Pérez, R.: Merging information under constraints: A logical framework. J. Log. Comput. **12**(5), 773–808 (2002)
19. Lagniez, J.M., Lonca, E., Mailly, J.G.: CoQuiAAS: A constraint-based quick abstract argumentation solver. In: Proc. of ICTAI'15. pp. 928–935 (2015)
20. Lagniez, J.M., Lonca, E., Mailly, J.G., Rossit, J.: Introducing the fourth international competition on computational models of argumentation. In: Proc. of SAFA'20. vol. 2672, pp. 80–85. CEUR-WS.org (2020), http://ceur-ws.org/Vol-2672/paper_9.pdf
21. Li, C.M., Manyà, F.: Maxsat, hard and soft constraints. In: Biere, A., Heule, M., van Maaren, H., Walsh, T. (eds.) Handbook of Satisfiability, Frontiers in Artificial Intelligence and Applications, vol. 185, pp. 613–631. IOS Press (2009). https://doi.org/10.3233/978-1-58603-929-5-613, https://doi.org/10.3233/978-1-58603-929-5-613
22. Linsbichler, T., Pührer, J., Strass, H.: A uniform account of realizability in abstract argumentation. In: Proc. of ECAI'16. pp. 252–260 (2016). https://doi.org/10.3233/978-1-61499-672-9-252, https://doi.org/10.3233/978-1-61499-672-9-252
23. Mailly, J.G.: Constrained incomplete argumentation frameworks. In: Proc. of ECSQARU'21. pp. 103–116. Springer (2021). https://doi.org/10.1007/978-3-030-86772-0_8, https://doi.org/10.1007/978-3-030-86772-0_8
24. Niskanen, A., Järvisalo, M.: μ-toksia: An efficient abstract argumentation reasoner. In: Proc. of KR'20. pp. 800–804 (2020). https://doi.org/10.24963/kr.2020/82, https://doi.org/10.24963/kr.2020/82
25. Niskanen, A., Wallner, J.P., Järvisalo, M.: Extension enforcement under grounded semantics in abstract argumentation. In: Proc. of KR'18. pp. 178–183. AAAI Press (2018)
26. Niskanen, A., Wallner, J.P., Järvisalo, M.: Synthesizing argumentation frameworks from examples. J. Artif. Intell. Res. **66**, 503–554 (2019). https://doi.org/10.1613/jair.1.11758, https://doi.org/10.1613/jair.1.11758
27. Oren, N., Yun, B.: Inferring attack relations for gradual semantics. CoRR **abs/2211.16118** (2022). https://doi.org/10.48550/arXiv.2211.16118, https://doi.org/10.48550/arXiv.2211.16118
28. Oren, N., Yun, B., Vesic, S., Baptista, M.S.: Inverse problems for gradual semantics. In: Proc. of IJCAI'22. pp. 2719–2725 (2022). https://doi.org/10.24963/ijcai.2022/377, https://doi.org/10.24963/ijcai.2022/377

29. Rabe, M.N., Tentrup, L.: CAQE: A certifying QBF solver. In: Proc. of FMCAD'15. pp. 136–143 (2015)
30. Skiba, K., Thimm, M., Rienstra, T., Heyninck, J., Kern-Isberner, G.: Realisability of rankings-based semantics. In: Proc. of SAFA'22. pp. 73–85. CEUR-WS.org (2022)
31. Thimm, M., Villata, S.: The first international competition on computational models of argumentation: Results and analysis. Artif. Intell. **252**, 267–294 (2017)
32. Tseytin, G.S.: On the complexity of derivation in propositional calculus. In: Studies in Constructive Mathematics and Mathematical Logic, Part II, Seminars in Mathematics. pp. 115–125 (1970), translated from Russian
33. Wallner, J.P., Niskanen, A., Järvisalo, M.: Complexity results and algorithms for extension enforcement in abstract argumentation. J. Artif. Intell. Res. **60**, 1–40 (2017)