Complexity of Verification and Existence Problems in Epistemic Argumentation Framework

Gianvincenzo Alfano, Sergio Greco, Domenico Mandaglio, Francesco Parisi and Irina Trubitsyna

Department of Informatics, Modeling, Electronics and System Engineering,
University of Calabria, Italy
{g.alfano,greco,d.mandaglio,fparisi,trubitsyna}@dimes.unical.it

Abstract. Dung's Argumentation Framework (AF) has been extended in several directions. An interesting extension, among others, is the *Epistemic AF* (EAF) which allows representing the agent's belief by means of epistemic constraints. In particular, an epistemic constraint is a propositional formula over labeled arguments (e.g. $\mathbf{in}(\mathtt{a})$, $\mathbf{out}(\mathtt{c})$) extended with the modal operators $\mathbf{K}$ and $\mathbf{M}$ that intuitively state that the agent believes that a given formula is certainly or possibly true, respectively. In this paper, focusing on EAF, we investigate the complexity of the *possible* and *necessary* variants of three canonical problems in abstract argumentation: *verification*, *existence*, and *non-empty existence*. Moreover, we explore the relationship between EAF and *incomplete AF* (iAF), an extension of AF where arguments and attacks may be uncertain. Our complexity analysis shows that the verification problem in iAF can be naturally reduced to the verification in EAF, while it turns out that a similar result cannot hold for the necessary (non-empty) existence problem.

1 Introduction

In the last decades, Formal Argumentation has become an important research field in AI [40]. Argumentation has potential applications in several contexts, including e.g. modeling dialogues, negotiation [7, 31], and persuasion [54]. Dung's Argumentation Framework (AF) is a simple yet powerful formalism for modeling disputes between two or more agents [33]. An AF consists of a set of *arguments* and a binary *attack* relation over the set of arguments that specifies the interactions between arguments: intuitively, if argument a attacks argument b, then b is acceptable only if a is not. Hence, arguments are abstract entities whose status is entirely determined by the attack relation. An AF can be seen as a directed graph, whose nodes represent arguments and edges represent attacks. Several argumentation semantics—e.g. *grounded* (gr), *complete* (co), *preferred* (pr), *stable* (st), and *semi-stable* (sst) [33, 24]—have been defined for AF, leading to the characterization of *σ-extensions*, that intuitively consist of the sets of arguments that can be collectively accepted under semantics $\sigma \in \{\mathtt{gr}, \mathtt{co}, \mathtt{st}, \mathtt{pr}, \mathtt{sst}\}$.

Example 1. Consider AF $\Lambda_1 = \langle \mathrm{A}_1 = \{\mathtt{a}, \mathtt{b}, \mathtt{c}, \mathtt{d}\}, \mathrm{R}_1 = \{(\mathtt{a}, \mathtt{b}), (\mathtt{b}, \mathtt{c}), (\mathtt{c}, \mathtt{d}), (\mathtt{d}, \mathtt{c})\}\rangle$ whose graph is shown in Figure 1 (left). Λ_1 describes the following scenario. A party planner invites Alice (a), Bob (b), Carl (c) and David (d) to join a party. Alice replies that she will join the party. However, due to their rivalry, (*i*) Bob replies that he will join the party if Alice does not; (*ii*) Carl replies that he will join the party if both Bob and David do not; (*iii*) David replies that he will join the party if Carl does not. This situation can be modeled by AF Λ_1, where an argument x states that "*(the person whose initial is)* x *joins the party*". Under the preferred (stable, and semi-stable) semantics, Λ_1 has extensions $E_1 = \{\mathtt{a}, \mathtt{c}\}$ and $E_2 = \{\mathtt{a}, \mathtt{d}\}$, meaning that either Alice and Carl, or Alice and David will attend the party. □

Figure 1: AF Λ_1 of Example 1 (left); AF Λ_3 of Example 3 (right).

Argumentation semantics can be also defined in terms of *labelling* [10]. Intuitively, a σ-labelling for an AF is a total function $\mathcal{L}$ assigning to each argument the label $\mathbf{in}$ if it is accepted, $\mathbf{out}$ if it is rejected, and $\mathbf{und}$ if it is undecided under σ. For instance, $\mathcal{L}_1 = \{\mathbf{in}(\mathtt{a}), \mathbf{out}(\mathtt{b}), \mathbf{in}(\mathtt{c}), \mathbf{out}(\mathtt{d})\}$ and $\mathcal{L}_2 = \{\mathbf{in}(\mathtt{a}), \mathbf{out}(\mathtt{b}), \mathbf{out}(\mathtt{c}), \mathbf{in}(\mathtt{d})\}$ are the σ-labellings for AF Λ_1 of Example 1 under semantics $\sigma \in \{\mathtt{st}, \mathtt{pr}, \mathtt{sst}\}$. Herein, $\mathcal{L}_1$ and $\mathcal{L}_2$ correspond to E_1 and E_2, respectively.

Despite the expressive power and generality of Dung's framework, in some cases it is difficult to accurately model domain knowledge by an AF in a natural and easy-to-understand way. For this reason, Dung's framework has been extended by introducing further constructs, such as preferences [5, 51], weights [18, 19, 17], supports [28, 25, 44], topics [21], and constraints [29, 9, 55, 2], to achieve more comprehensive, natural, and compact ways for representing useful relationships among arguments.

In the following we focus on an interesting extension of Dung's framework with *epistemic constraints* called *Epistemic Argumentation Framework* (EAF) [55]. Herein, an epistemic constraint represents the belief of an agent that must be satisfied. In particular, an epistemic constraint is a propositional formula over labeled arguments (e.g. $\mathbf{in}(\mathtt{a})$, $\mathbf{out}(\mathtt{c})$) extended with the modal operators $\mathbf{K}$ and $\mathbf{M}$. Intuitively, $\mathbf{K}\phi$ (resp. $\mathbf{M}\phi$) states that the considered agent believes that ϕ is always (resp. possibly) true. The semantics of an EAF is given by the set of so-called *σ-epistemic labelling* sets. Intuitively, a σ-epistemic labelling set is a collection of σ-labellings that reflects the belief of an agent. More in detail, every *σ-epistemic labelling* set consists of σ-labellings of the underlying AF and it is a maximal set of σ-labellings that satisfy the epistemic constraint.

Example 2. Consider the AF $\Lambda_1 = \langle \mathrm{A}_1, \mathrm{R}_1\rangle$ of Example 1, and assume that the party planner believes that Carl will certainly join the party. This can be modeled by EAF $\Delta_2 = \langle \mathrm{A}_1, \mathrm{R}_1, \varphi \rangle$, where the epistemic constraint $\varphi = \mathbf{Kin}(\mathtt{c})$ states that c must be accepted in every solution. For $\sigma \in \{\mathtt{st}, \mathtt{pr}, \mathtt{sst}\}$, Δ_2 has one σ-epistemic labelling set consisting of $\mathcal{L}_1$ only, meaning that the party planner concludes that Alice and Carl will attend the party. □

An EAF may have multiple σ-epistemic labelling sets.

Example 3. Consider the AF $\Lambda_3 = \langle \mathrm{A}_3, \mathrm{R}_3 \rangle$ shown in Figure 1 (right), where $\mathrm{A}_3 = \mathrm{A}_1$, $\mathrm{R}_3 = \mathrm{R}_1 \cup \{(\mathtt{b}, \mathtt{a})\}$, and A_1 and R_1 are as defined in Example 1. The set of its σ-labellings with $\sigma \in \{\mathtt{st}, \mathtt{pr}, \mathtt{sst}\}$ is $\{\mathcal{L}_1, \mathcal{L}_2, \mathcal{L}_3\}$, where $\mathcal{L}_1$ and $\mathcal{L}_2$ are the σ-labellings for AF Λ_1 of Example 1 and $\mathcal{L}_3 = \{\mathbf{out}(\mathtt{a}), \mathbf{in}(\mathtt{b}), \mathbf{out}(\mathtt{c}), \mathbf{in}(\mathtt{d})\}$. Then, EAF $\Delta_3 = \langle \mathrm{A}_3, \mathrm{R}_3, \mathbf{Kin}(\mathtt{a}) \vee \mathbf{Kin}(\mathtt{d}) \rangle$ has two σ-epistemic labelling sets, $\{\mathcal{L}_1, \mathcal{L}_2\}$ and $\{\mathcal{L}_2, \mathcal{L}_3\}$, representing the scenarios compliant with the belief of the party planner that Alice or David will certainly join the party. □

However, the existence of a σ-labelling is not guaranteed in EAF. That is, even for semantics prescribing at least one σ-labelling for the underlying AF (e.g. $\sigma \in \{\mathtt{gr}, \mathtt{co}, \mathtt{pr}, \mathtt{sst}\}$), we can have EAFs having no non-empty σ-epistemic labelling set.

Example 4. The EAF $\Delta_4 = \langle \mathrm{A}_3, \mathrm{R}_3, \mathbf{K}(\mathbf{in}(\mathtt{a}) \wedge \mathbf{in}(\mathtt{b})) \rangle$ (that differs from the EAF Δ_3 of Example 3 in the epistemic constraint only) has no grounded, complete, preferred, stable, and semi-stable labelling, meaning that any σ-epistemic labelling set is empty for any $\sigma \in \{\mathtt{gr}, \mathtt{co}, \mathtt{st}, \mathtt{pr}, \mathtt{sst}\}$. □

Besides the problem of deciding the existence of a σ-labelling, two additional fundamental problems investigated in AF are *i*) *non-empty existence*, that is deciding if there is a σ-labelling prescribing a non-empty set of accepted arguments, and *ii*) *verification*, that is deciding whether a given assignment of labels in $\{\mathbf{in}, \mathbf{out}, \mathbf{und}\}$ to each argument is a σ-labelling. In general, the (non-empty) existence and verification problems are important to understand the behavior of argumentation semantics. For this reason, the complexity of these problems have been explored in detail for AF [36] as well as for several frameworks extending AF, such as *incomplete AF* (iAF) [14, 13, 57], where arguments and attacks may be uncertain.

In this paper, we investigate the complexity of the *verification*, *existence*, and *non-empty existence* problems in EAF and explore the relationship between EAF and iAF.

Contributions. Our main contributions are as follows.

- We first introduce the *possible* and *necessary* variants of the verification, existence, and non-empty existence problems in EAF by taking into account the fact that an EAF may have multiple σ-epistemic labelling sets. Intuitively, given an EAF, the possible (resp. necessary) verification problem consist in deciding whether a candidate labelling—a given assignment of the labels in $\{\mathbf{in}, \mathbf{out}, \mathbf{und}\}$ to each argument—belongs to *any* (resp. *every*) σ-epistemic labelling set, with $\sigma \in \{\mathtt{gr}, \mathtt{co}, \mathtt{st}, \mathtt{pr}, \mathtt{sst}\}$. The *possible* (resp. *necessary*) existence problem consists in deciding whether there is σ-labelling in *any* (resp. *every*) σ-epistemic labelling set. Moreover, the *possible* (resp. *necessary*) non-empty existence problem consists in deciding whether there is a σ-labelling prescribing a non-empty set of accepted arguments in *any* (resp. *every*) σ-epistemic labelling set.
- We explore the complexity of the possible and necessary variants of the verification, existence, and non-empty existence problems for EAF, showing that in most cases these problems are harder than those for AF. In particular, the complexity of the verification problems for EAF increases of one level in the polynomial hierarchy w.r.t. that for AF (cf. Table 1). Moreover, the complexity of the possible and necessary existence problems for EAF increase of at least one level (in the polynomial hierarchy) w.r.t. that for AF, except for the stable semantics for which it remains the same. The complexity of the possible and necessary non-empty existence problems for EAF increase of one level w.r.t. that for AF for both the preferred and semi-stable semantics, while it remains the same for the complete and stable semantics. A complete picture of the complexity of the (non-empty) existence problems is given in Table 2, where the results for iAF are reported as well.
- Finally, we analyze the relationship between EAF and iAF, showing that (possible and necessary) verification in iAF can be reduced to (possible and necessary) verification in EAF under complete, preferred, stable and semi-stable semantics.

2 Preliminaries

In this section, after recalling some complexity classes, we review the AF-based frameworks considered in the paper.

2.1 Complexity Classes

We recall here the main complexity classes used in the paper. NP is the class of decision problems solvable by a non-deterministic Turing machine in polynomial time, while coNP is the class of decision problems whose complement is in NP. Moreover, the classes Σ_k^p and Π_k^p with $k \geq 0$ are defined as (see e.g. [53]):

- $\Sigma_0^p = \Pi_0^p = P$;
- $\Sigma_1^p = NP$ and $\Pi_1^p = coNP$;
- $\Sigma_k^p = NP^{\Sigma_{k-1}^p}$ and $\Pi_k^p = co\Sigma_k^p, \forall k > 0$.

For a complexity class C, NP^C denotes the class of problems that can be solved in polynomial time using an oracle in C by a non-deterministic Turing machine. Under the standard complexity-theoretic assumptions, we have that $\Sigma_k^p \subset \Sigma_{k+1}^p \subseteq PSPACE$ and $\Pi_k^p \subset \Pi_{k+1}^p \subseteq PSPACE\ \forall k \geq 0$. A problem is said to be C-hard if there is a polynomial-time many-to-one reduction to it from any problem in C. Additionally, if the considered problem belongs to C, then it is said to be C-complete. Also, a problem is C-hard if there is a polynomial-time reduction to it from a C-complete problem.

2.2 Argumentation Framework

An abstract *Argumentation Framework* (AF) is a pair $\langle \mathrm{A}, \mathrm{R} \rangle$, where A is a set of *arguments* and $\mathrm{R} \subseteq \mathrm{A} \times \mathrm{A}$ is a set of *attacks*. If $(a, b) \in \mathrm{R}$ then we say that a attacks b.

Given an AF $\Lambda = \langle \mathrm{A}, \mathrm{R} \rangle$ and a set $S \subseteq \mathrm{A}$ of arguments, an argument $a \in \mathrm{A}$ is said to be *i*) *defeated* w.r.t. S iff $\exists b \in S$ such that $(b, a) \in \mathrm{R}$, and *ii*) *acceptable* w.r.t. S iff for every argument $b \in \mathrm{A}$ with $(b, a) \in \mathrm{R}$, there is $c \in S$ such that $(c, b) \in \mathrm{R}$. The sets of defeated and acceptable arguments w.r.t. S are as follows (where Λ is understood):

- $\mathit{Def}(S) = \{a \in \mathrm{A} \mid \exists (b, a) \in \mathrm{R} \,.\, b \in S\}$;
- $\mathit{Acc}(S) = \{a \in \mathrm{A} \mid \forall (b, a) \in \mathrm{R} \,.\, b \in \mathit{Def}(S)\}$.

Given an AF $\langle \mathrm{A}, \mathrm{R} \rangle$, a set $S \subseteq \mathrm{A}$ of arguments is said to be:

- *conflict-free* iff $S \cap Def(S) = \emptyset$;
- *admissible* iff it is conflict-free and $S \subseteq Acc(S)$.

Different argumentation semantics have been proposed to characterize collectively acceptable sets of arguments, called *extensions* [33, 24]. Every extension is an admissible set satisfying additional conditions. Specifically, the complete, preferred, stable, semi-stable, and grounded extensions of an AF are defined as follows.

Given an AF $\langle \mathrm{A}, \mathrm{R} \rangle$, a set $S \subseteq \mathrm{A}$ is an *extension* called:

- *complete* ($\mathtt{co}$) iff it is an admissible set and $S = Acc(S)$;
- *preferred* ($\mathtt{pr}$) iff it is a $\subseteq$-maximal complete extension;
- *stable* ($\mathtt{st}$) iff it is a total preferred extension, i.e. a preferred extension such that $S \cup Def(S) = \mathrm{A}$;
- *semi-stable* ($\mathtt{sst}$) iff it is a preferred extension such that $S \cup Def(S)$ is maximal (w.r.t. $\subseteq$);
- *grounded* ($\mathtt{gr}$) iff it is a $\subseteq$-minimal complete extension.

The set of complete (resp. preferred, stable, semi-stable, grounded) extensions of an AF Λ will be denoted by $\mathtt{co}(\Lambda)$ (resp. $\mathtt{pr}(\Lambda)$, $\mathtt{st}(\Lambda)$, $\mathtt{sst}(\Lambda)$, $\mathtt{gr}(\Lambda)$). It is well-known that the set of complete extensions forms a complete semilattice w.r.t. $\subseteq$, where $\mathtt{gr}(\Lambda)$ is the meet element, whereas the greatest elements are the preferred extensions. All the above-mentioned semantics except the stable admit at least one extension. The grounded semantics, that admits exactly one extension, is said to be a *unique status* semantics, while the others are said to be *multiple status* semantics. With a little abuse of notation, in the following we also use $\mathtt{gr}(\Lambda)$ to denote the grounded extension. For any AF Λ the following inclusion relations hold: *i)* $\mathtt{st}(\Lambda) \subseteq \mathtt{sst}(\Lambda) \subseteq \mathtt{pr}(\Lambda) \subseteq \mathtt{co}(\Lambda)$, *ii)* $\mathtt{gr}(\Lambda) \in \mathtt{co}(\Lambda)$, and *iii)* $\mathtt{st}(\Lambda) \neq \emptyset$ implies that $\mathtt{st}(\Lambda) = \mathtt{sst}(\Lambda)$. Arguments occurring in an extension are said to be accepted, whereas arguments attacked by accepted arguments are said to be rejected; the remaining arguments are said to be undecided (w.r.t. the considered extension).

2.2.1 Labelling

The argumentation semantics can be also defined in terms of *labelling* [10]. A labelling for an AF $\langle \mathrm{A}, \mathrm{R} \rangle$ is a total function $\mathcal{L} : \mathrm{A} \rightarrow \{\mathbf{in}, \mathbf{out}, \mathbf{und}\}$ assigning to each argument a label: $\mathcal{L}(a) = \mathbf{in}$ means that a is accepted, $\mathcal{L}(a) = \mathbf{out}$ means that a is rejected, and $\mathcal{L}(a) = \mathbf{und}$ means that a is undecided.

Let $\mathbf{in}(\mathcal{L}) = \{a \mid a \in \mathrm{A} \wedge \mathcal{L}(a) = \mathbf{in}\}$, $\mathbf{out}(\mathcal{L}) = \{a \mid a \in \mathrm{A} \wedge \mathcal{L}(a) = \mathbf{out}\}$, and $\mathbf{und}(\mathcal{L}) = \{a \mid a \in \mathrm{A} \wedge \mathcal{L}(a) = \mathbf{und}\}$, a labelling $\mathcal{L}$ can be represented by means of a triple $\langle \mathbf{in}(\mathcal{L}), \mathbf{out}(\mathcal{L}), \mathbf{und}(\mathcal{L}) \rangle$. We also use the notation $\mathbf{in}(a)$ (resp. $\mathbf{out}(a)$, $\mathbf{und}(a)$) to denote that $a \in \mathbf{in}(\mathcal{L})$ (resp. $a \in \mathbf{out}(\mathcal{L})$, $a \in \mathbf{und}(\mathcal{L})$).

Given an AF $\Lambda = \langle \mathrm{A}, \mathrm{R} \rangle$, a labelling $\mathcal{L}$ for A is said to be *admissible (or legal)* if $\forall a \in \mathbf{in}(\mathcal{L}) \cup \mathbf{out}(\mathcal{L})$ it holds that:

(i) $\mathcal{L}(a) = \mathbf{out}$ iff $\exists\,(b, a) \in \mathrm{R}$ such that $\mathcal{L}(b) = \mathbf{in}$; and
(ii) $\mathcal{L}(a) = \mathbf{in}$ iff $\forall (b, a) \in \mathrm{R}$, $\mathcal{L}(b) = \mathbf{out}$ holds.

Moreover, $\mathcal{L}$ is a *complete* labelling iff conditions (*i*) and (*ii*) hold for all arguments $a \in \mathrm{A}$.

Between complete extensions and complete labellings there is a bijective mapping defined as follows: for each extension E there is a unique labelling $\mathcal{L}(E) = \langle E, Def(E), \mathrm{A} \setminus (E \cup Def(E)) \rangle$ and for each labelling $\mathcal{L}$ there is a unique extension, that is $\mathbf{in}(\mathcal{L})$. We say that $\mathcal{L}(E)$ is the labelling *corresponding* to E. Moreover, we say that $\mathcal{L}(E)$ is a σ-labelling for a given AF Λ and semantics $\sigma \in \{\mathtt{co}, \mathtt{pr}, \mathtt{st}, \mathtt{sst}, \mathtt{gr}\}$ iff E is a σ-extension of Λ.

In the following, we say that the *status of an argument* a w.r.t. a labelling $\mathcal{L}$ (or its corresponding extension $\mathbf{in}(\mathcal{L})$) is $\mathbf{in}$ (resp. $\mathbf{out}$, $\mathbf{und}$) iff $\mathcal{L}(a) = \mathbf{in}$ (resp. $\mathcal{L}(a) = \mathbf{out}$, $\mathcal{L}(a) = \mathbf{und}$). We will avoid to mention explicitly the labelling (or the extension) whenever it is understood.

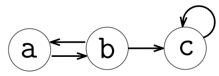

Figure 2: AF Λ_5 of Example 5.

Example 5. Let $\Lambda_5 = \langle \mathrm{A}_5, \mathrm{R}_5 \rangle$ be an AF where $\mathrm{A}_5 = \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$ and $\mathrm{R}_5 = \{(\mathtt{a}, \mathtt{b}), (\mathtt{b}, \mathtt{a}), (\mathtt{b}, \mathtt{c}), (\mathtt{c}, \mathtt{c})\}$ whose graph is shown in Figure 2. AF Λ_5 has three complete extensions: $E_1 = \emptyset, E_2 = \{\mathtt{a}\}, E_3 = \{\mathtt{b}\}$, whose corresponding complete labellings are $\mathcal{L}_1 = \langle \emptyset, \emptyset, \{\mathtt{a}, \mathtt{b}, \mathtt{c}\} \rangle$, $\mathcal{L}_2 = \langle \{\mathtt{a}\}, \{\mathtt{b}\}, \{\mathtt{c}\} \rangle$, and $\mathcal{L}_3 = \langle \{\mathtt{b}\}, \{\mathtt{a}, \mathtt{c}\}, \emptyset \rangle$. Also, the set of preferred extensions is $\{E_2, E_3\}$, whereas the set of stable (and semi-stable) extensions is $\{E_3\}$, and the grounded extension is E_1. Correspondingly, the $\mathtt{pr}$-labelling set is $\{\mathcal{L}_2, \mathcal{L}_3\}$, the $\mathtt{st}$- and $\mathtt{sst}$-labelling set is $\{\mathcal{L}_3\}$, while the $\mathtt{gr}$-labelling set is $\{\mathcal{L}_1\}$. □

Three fundamental problems in AF are *verification*, *existence* and *non-empty existence*. Given an AF $\Lambda = \langle \mathrm{A}, \mathrm{R} \rangle$, a set $S \subseteq A$ of arguments, and a semantics $\sigma \in \{\mathtt{gr}, \mathtt{co}, \mathtt{st}, \mathtt{pr}, \mathtt{sst}\}$:

- the *verification* problem (denoted as V_σ) is the problem of deciding whether $S \in \sigma(\Lambda)$, that is, deciding whether S is a σ-extension of Λ;
- the *existence* problem (denoted as EX_σ) is the problem of deciding whether $\sigma(\Lambda) \neq \emptyset$, that is, deciding whether there exists at least one σ-extension of Λ;
- the *non-empty existence* problem (denoted as $\mathsf{EX}_\sigma^{\neg\emptyset}$) is the problem of deciding whether there exists $S \in \sigma(\Lambda)$ s.t. $S \neq \emptyset$, i.e. deciding whether there exists a non-empty σ-extension of Λ.

Clearly, for the grounded, complete, preferred and semi-stable semantics, which always admit at least one extension, the existence problem is trivial—this is not the case of deciding the non-empty existence problem. The complexity of (non-empty) existence and verification problems for AF has been thoroughly investigated (see [36] for a survey). A summary of the results is given in Tables 1 and 2.

2.3 Incomplete ArgumentationFramework

We now recall the incomplete AF [14, 39].

Definition 1 (iAF). An *incomplete AF* (iAF) is a tuple $\langle \mathrm{A}, \mathrm{B}, \mathrm{R}, \mathrm{T} \rangle$, where A and B are disjoint sets of arguments, and R and T are disjoint sets of attacks between arguments in $\mathrm{A} \cup \mathrm{B}$. Arguments in A and attacks in R are said to be *certain*, while arguments in B and attacks in T are said to be *uncertain*.

Certain arguments in A are definitely known to exist, while uncertain arguments in B are not known for sure: they may occur or may not. Analogously, certain attacks in R are definitely known to exist if their incident arguments exist, while for uncertain attacks in T it is not known for sure if they hold, even if the incident arguments exist.

An iAF $\langle \mathrm{A}, \mathrm{B}, \mathrm{R}, \mathrm{T} \rangle$ is said to be an *arg-iAF* iff $\mathrm{T} = \emptyset$, i.e. it does not contain uncertain attacks. We may omit the empty set T and use a triple $\langle \mathrm{A}, \mathrm{B}, \mathrm{R} \rangle$ to denote an arg-iAF.

An iAF compactly represents alternative AF scenarios, called *completions*. A *completion* for an iAF $\Delta = \langle \mathrm{A}, \mathrm{B}, \mathrm{R}, \mathrm{T} \rangle$ is an AF $\Lambda = \langle \mathrm{A}', \mathrm{R}' \rangle$ such that $\mathrm{A} \subseteq \mathrm{A}' \subseteq \mathrm{A} \cup \mathrm{B}$ and $\mathrm{R} \cap (\mathrm{A}' \times \mathrm{A}') \subseteq \mathrm{R}' \subseteq (\mathrm{R} \cup \mathrm{T}) \cap (\mathrm{A}' \times \mathrm{A}')$.

Existence problems in iAF have been investigated in [57]. Given an iAF Δ and a semantics $\sigma \in \{\mathtt{gr}, \mathtt{co}, \mathtt{pr}, \mathtt{st}, \mathtt{sst}\}$,

1. the *possible existence* problem under σ, denoted as PEX_σ, consists in deciding whether there exists a completion Λ of Δ that has at least one σ-extension;

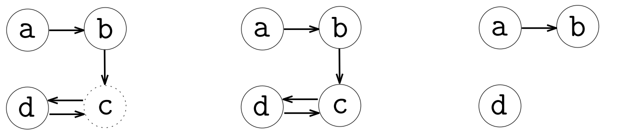

Figure 3: Arg-iAF Δ_7 of Example 7 (left), and its completions Λ_7' (center) and Λ_7'' (right).

2. the *possible non-empty existence* problem under σ, denoted as $\mathsf{PEX}_\sigma^{\neg\emptyset}$, consists in deciding whether there exists a completion Λ of Δ that has a non-empty σ-extension;
3. the *necessary existence* problem under σ, denoted as NEX_σ, consists in deciding whether all completions Λ of Δ have a σ-extension;
4. the *necessary non-empty existence* problem under σ, denoted as $\mathsf{NEX}_\sigma^{\neg\emptyset}$, consists in deciding whether all completions Λ of Δ have a non-empty σ-extension.

As the completions of (any) iAF prescribe at least one σ-extension for $\sigma \in \{\mathtt{gr}, \mathtt{co}, \mathtt{pr}, \mathtt{sst}\}$, PEX_σ and NEX_σ are trivial under σ [57].

Fact 1. *For any iAF and semantics $\sigma \in \{\mathtt{gr}, \mathtt{co}, \mathtt{pr}, \mathtt{sst}\}$, PEX_σ and NEX_σ are trivial.*

However, under stable semantics, the existence of at least one extension for any completion is not guaranteed.

Example 6. Let $\Delta_6 = \langle \mathrm{A}_5 \setminus \{\mathtt{b}\}, \{\mathtt{b}\}, \mathrm{R}_5, \emptyset\rangle$ be an iAF, where $\langle \mathrm{A}_5, \mathrm{R}_5\rangle$ is the AF of Example 5. The completion $\langle \mathrm{A}_5, \mathrm{R}_5\rangle$ has the stable extension $\{\mathtt{b}\}$ (as observed in Example 5), while the completion $\langle\{\mathtt{a}, \mathtt{c}\}, \{(\mathtt{c}, \mathtt{c})\}\rangle$ has no stable extension. □

It is worth noting that, for any iAF, $\mathsf{PEX}_{\mathtt{st}}$ is NP-complete and $\mathsf{NEX}_{\mathtt{st}}$ is Π_2^p-complete. These results follow from Proposition 18 and Theorem 21 in [57] by observing that $\mathsf{PEX}_{\mathtt{st}}(\Delta)$ (resp. $\mathsf{NEX}_{\mathtt{st}}(\Delta)$) is true iff $\mathsf{PEX}_{\mathtt{st}}^{\neg\emptyset}(\Delta)$ (resp. $\mathsf{NEX}_{\mathtt{st}}^{\neg\emptyset}(\Delta)$) is true. The complexity of the existence problems for iAF is reported in Table 2.

The following verification problems for iAF have been investigated in [14, 39]. Given an iAF $\Delta = \langle \mathrm{A}, \mathrm{B}, \mathrm{R}, \mathrm{T}\rangle$, a set of arguments $S \subseteq (\mathrm{A} \cup \mathrm{B})$, and a semantics $\sigma \in \{\mathtt{gr}, \mathtt{co}, \mathtt{pr}, \mathtt{st}, \mathtt{sst}\}$,

1. the *possible verification* problem under σ (denoted as PV_σ) consists in deciding whether there exists a completion Λ of Δ such that S is a σ-extension of Λ;
2. the *necessary verification* problem under σ (denoted as NV_σ) consists in deciding whether for all completions Λ of Δ it holds that S is a σ-extension of Λ.

Example 7. Consider the AF of Example 1 and assume that the participation of Carl is uncertain. This can be modeled by the (arg-)iAF Δ_7=$\langle\{\mathtt{a}, \mathtt{b}, \mathtt{d}\}, \{\mathtt{c}\}, \{(\mathtt{a}, \mathtt{b}), (\mathtt{b}, \mathtt{c}), (\mathtt{c}, \mathtt{d}), (\mathtt{d}, \mathtt{c})\}, \emptyset\rangle$ whose graph is shown in Figure 3 (left), where the uncertain argument is represented by a dotted circle. Δ_7 has 2 completions: Λ_7'= $\langle\{\mathtt{a}, \mathtt{b}, \mathtt{c}, \mathtt{d}\}, \{(\mathtt{a}, \mathtt{b}), (\mathtt{b}, \mathtt{c}), (\mathtt{c}, \mathtt{d}), (\mathtt{d}, \mathtt{c})\}\rangle$ and Λ_7''= $\langle\{\mathtt{a}, \mathtt{b}, \mathtt{d}\}, \{(\mathtt{a}, \mathtt{b})\}\rangle$, also shown in Figure 3. Under semantics $\sigma \in \{\mathtt{st}, \mathtt{pr}, \mathtt{sst}\}$, AF Λ_7' has two extensions, E_1=$\{\mathtt{a}, \mathtt{d}\}$ and E_2=$\{\mathtt{a}, \mathtt{c}\}$, while AF Λ_7'' has only one extension, that is E_1. Thus, given iAF Δ_7 and either E_1 or E_2 we have that PV_σ is true, while NV_σ is true for E_1 only. That is, E_1 and E_2 are possible extensions, but only E_1 is a necessary one. □

The complexity of PV_σ and NV_σ for iAF has been investigated in [14] for semantics $\sigma \in \{\mathtt{gr}, \mathtt{co}, \mathtt{st}, \mathtt{pr}\}$ and in [3] for the semi-stable semantics. The complexity results are summarized in Table 1.

2.4 Epistemic Argumentation Framework

In this section, we review the *Epistemic Argumentation Framework* [55], which extends Dungs' framework with epistemic constraints.

Given an AF $\Lambda = \langle \mathrm{A}, \mathrm{R}\rangle$, an epistemic atom over Λ is of the form $\mathbf{K}\,\varphi$ or $\mathbf{M}\,\varphi$, where $\mathbf{K}$ and $\mathbf{M}$ are called modal operators, and φ is a propositional formula built from $\lambda_A = \{\mathbf{in}(a), \mathbf{out}(a), \mathbf{und}(a) \mid a \in \mathrm{A}\}$ by using the connectives $\neg$, $\vee$, and $\wedge$. Moreover, an epistemic literal is an epistemic atom or its negation. An *epistemic formula* (over λ_A) is a propositional formula constructed over epistemic literals and connectives $\wedge$ and $\vee$. Epistemic formulae introduce subjective knowledge of agents, whereas the AF encodes the objective knowledge. Intuitively, $\mathbf{K}\,\varphi$ (resp. $\mathbf{M}\,\varphi$) means that the considered agent believes that φ is certainly true (resp. φ is possibly true).[1]

The satisfaction of a propositional formula φ over λ_A w.r.t. a labelling $\mathcal{L}$ (denoted as $\mathcal{L}(S) \models \varphi$) holds if the formula obtained from φ by replacing every atom occurring in $\mathcal{L}(S)$ with $\mathtt{t}$ ($\mathtt{true}$), and every atom not occurring in $\mathcal{L}(S)$ with $\mathtt{f}$ ($\mathtt{false}$), evaluates to true.

A set $\mathcal{L}^S$ of labellings satisfies an epistemic formula φ, denoted as $\mathcal{L}^S \models \varphi$, if one of the following conditions holds:

- $\varphi = \mathtt{t}$,
- $\varphi = \mathbf{K}\psi$ and $\mathcal{L} \models \psi$ for every $\mathcal{L} \in \mathcal{L}^S$,
- $\varphi = \mathbf{M}\psi$ and $\mathcal{L} \models \psi$ for some $\mathcal{L} \in \mathcal{L}^S$,
- $\varphi = \neg\psi$ and $\mathcal{L}^S \not\models \psi$,
- $\varphi = \varphi_1 \wedge \varphi_2$ and ($\mathcal{L}^S \models \varphi_1$ and $\mathcal{L}^S \models \varphi_2$),
- $\varphi = \varphi_1 \vee \varphi_2$ and ($\mathcal{L}^S \models \varphi_1$ or $\mathcal{L}^S \models \varphi_2$).

An epistemic formula φ is consistent if there exists a (non-empty) set $\mathcal{L}^S$ of labellings such that $\mathcal{L}^S \models \varphi$; otherwise, φ is inconsistent. The following basic properties hold:

- $\mathcal{L}^S \models \neg\mathbf{M}\varphi$ iff $\mathcal{L}^S \models \mathbf{K}\neg\varphi$,
- $\mathcal{L}^S \models \neg\mathbf{K}\varphi$ iff $\mathcal{L}^S \models \mathbf{M}\neg\varphi$,
- $\mathcal{L}^S \models \mathbf{M}(\varphi_1 \vee \varphi_2)$ iff $\mathcal{L}^S \models \mathbf{M}\varphi_1$ or $\mathcal{L}^S \models \mathbf{M}\varphi_2$,
- $\mathcal{L}^S \models \mathbf{K}(\varphi_1 \wedge \varphi_2)$ iff $\mathcal{L}^S \models \mathbf{K}\varphi_1$ and $\mathcal{L}^S \models \mathbf{K}\varphi_2$.

Definition 2 (EAF). *An* Epistemic AF (EAF) *is a triple $\langle \mathrm{A}, \mathrm{R}, \varphi\rangle$ where $\langle \mathrm{A}, \mathrm{R}\rangle$ is an AF and φ is an epistemic formula to be satisfied, also called epistemic constraint.*

Let $\Delta = \langle \mathrm{A}, \mathrm{R}, \varphi\rangle$ be an EAF and $\sigma \in \{\mathtt{gr}, \mathtt{co}, \mathtt{pr}, \mathtt{st}, \mathtt{sst}\}$ be a semantics. A set $\mathcal{L}^S$ of labellings is a *σ-epistemic labelling set* of Δ if (*i*) each $\mathcal{L} \in \mathcal{L}^S$ is a σ-labelling of $\langle \mathrm{A}, \mathrm{R}\rangle$, and (*ii*) $\mathcal{L}^S$ is a $\subseteq$-maximal set of σ-labellings of $\langle \mathrm{A}, \mathrm{R}\rangle$ that satisfies φ.

As discussed in Section 1, an EAF may have multiple σ-epistemic labelling sets. In fact, a σ-epistemic labelling set is a collection of σ-labellings that represent the belief of an agent. In particular, EAF $\Delta = \langle \mathrm{A}, \mathrm{R}, \mathtt{t}\rangle$ has a unique σ-epistemic labelling set that coincides with the set of σ-labellings of the underlying AF $\langle \mathrm{A}, \mathrm{R}\rangle$. By definition, an EAF always has a σ-epistemic labelling set (possibly an empty set). For instance, the EAF $\langle \mathrm{A}, \mathrm{R}, \mathtt{f}\rangle$ has the σ-epistemic labelling set $\emptyset$.

Example 8. Consider the EAF $\Delta_3 = \langle \mathrm{A}_3, \mathrm{R}_3, \mathbf{Kin}(\mathtt{a}) \vee \mathbf{Kin}(\mathtt{d})\rangle$, whose preferred (stable and semi-stable)-epistemic labelling sets are given in Example 3. We have that the only grounded epistemic labelling set for Δ_3 is $\emptyset$, as the grounded labelling $\mathcal{L} = \{\mathbf{und}(\mathtt{a}), \mathbf{und}(\mathtt{b}), \mathbf{und}(\mathtt{c}), \mathbf{und}(\mathtt{d})\}$ of the underlying AF $\Lambda_3 = \langle \mathrm{A}_1, \mathrm{R}_1 \cup \{(\mathtt{b}, \mathtt{a})\}\rangle$ does not satisfy the epistemic constraint, that is, $\mathcal{L} \not\models (\mathbf{Kin}(\mathtt{a}) \vee \mathbf{Kin}(\mathtt{d}))$. □

[1] We use $\mathbf{K}$ (‘Known’) and $\mathbf{M}$ (‘May hold’) to follow the original EAF notation which is based on that of epistemic logic programs, that in turn is borrowed from modal logic.

In the following, we assume that epistemic constraints are of form $\varphi = \varphi_1 \vee \cdots \vee \varphi_n$, where $\varphi_i = \mathbf{K}\varphi_{i,0} \wedge \cdots \wedge \mathbf{K}\varphi_{i,k_i} \wedge \mathbf{M}\varphi_{i,k_{i+1}} \wedge \cdots \wedge \mathbf{M}\varphi_{i,m_i}$ and each $\varphi_{i,j}$ (with $i \in [1..n]$, $j \in [0..m_i]$) is a general propositional formula.

3 Verification Problems in EAF

We first introduce the possible and necessary verification problems for EAF, and then investigate their complexity.

Definition 3 (Possible/Necessary Verification)**.** Given an EAF Δ, a semantics $\sigma \in \{\mathtt{gr}, \mathtt{co}, \mathtt{st}, \mathtt{pr}, \mathtt{sst}\}$, and a labelling $\mathcal{L}$,

1. the *possible verification* problem under σ (denoted as PV_σ) consists in deciding whether there is a σ-epistemic labelling set $\mathcal{L}^S$ of Δ such that $\mathcal{L} \in \mathcal{L}^S$;
2. the *necessary verification* problems under σ (denoted as NV_σ) consists in deciding whether for all σ-epistemic labelling sets $\mathcal{L}^S$ of Δ it holds that $\mathcal{L} \in \mathcal{L}^S$.

Given a pair $(\Delta, \mathcal{L})$, we use $\mathsf{PV}_\sigma(\Delta, \mathcal{L})$ (resp. $\mathsf{NV}_\sigma(\Delta, \mathcal{L})$) to denote the output of problem PV_σ (resp. NV_σ) over such instance.

Example 9. Let $\Delta_3 = \langle \mathrm{A}_3, \mathrm{R}_3, \mathbf{Kin}(\mathtt{a}) \vee \mathbf{Kin}(\mathtt{d}) \rangle$ be the EAF of Example 3. Recall that Δ_3 has two σ-epistemic labelling sets (under semantics $\sigma \in \{\mathtt{pr}, \mathtt{st}, \mathtt{sst}\}$): $\{\mathcal{L}_1, \mathcal{L}_2\}$ and $\{\mathcal{L}_2, \mathcal{L}_3\}$. Then $\mathcal{L}_1$, $\mathcal{L}_2$, and $\mathcal{L}_3$ are possible σ-labellings of Δ_3, while only $\mathcal{L}_2$ is a necessary σ-labelling of Δ_3. That is, PV_σ is true for $\mathcal{L}_1$, $\mathcal{L}_2$, and $\mathcal{L}_3$, while NV_σ is true for $\mathcal{L}_2$ only. □

The following theorems characterize the complexity of the possible and necessary verification problems for EAF.

Theorem 1. *For any EAF,* PV_σ *is:*

- *in P for* $\sigma = \mathtt{gr}$*;*
- NP*-complete for* $\sigma \in \{\mathtt{co}, \mathtt{st}\}$*;*
- Σ_2^p*-complete for* $\sigma \in \{\mathtt{pr}, \mathtt{sst}\}$.

Proof hint. The hardness results for complete and stable semantics can be shown by providing a reduction from the non-empty existence problem for AF. In fact, there exists a non-empty complete (resp. stable) extension in an AF $\langle \mathrm{A}, \mathrm{R} \rangle$ iff $\mathcal{L} = \{\mathbf{in}(\alpha), \mathbf{out}(\beta)\} \cup \{\mathbf{out}(x) \mid x \in \mathrm{A}\}$ belongs to at least one σ-epistemic labelling set for the EAF $\langle \mathrm{A}' = \mathrm{A} \cup \{\alpha, \beta\}, \mathrm{R}' = \mathrm{R} \cup \{(a, \alpha), (\alpha, a), (\alpha, \beta) \mid a \in \mathrm{A}\}, \varphi = \mathbf{M}(\mathbf{in}(\beta)) \rangle$. For the preferred and semi-stable semantics, we can show that any AF $\Lambda = \langle \mathrm{A}, \mathrm{R} \rangle$ is *not* coherent [34] iff $\{\mathbf{in}(\alpha), \mathbf{out}(\beta), \mathbf{und}(\gamma),\} \cup \{\mathbf{out}(a) \mid a \in \mathrm{A}\}$ belongs to at least one σ-epistemic labelling set for the EAF $\langle \mathrm{A}' = \mathrm{A} \cup \{\alpha, \beta, \gamma\}, \mathrm{R}' = \mathrm{R} \cup \{(a, \alpha), (\alpha, a) \mid a \in \mathrm{A}\} \cup \{(\alpha, \beta), (\beta, \gamma), (\gamma, \gamma)\}, \mathbf{M}(\bigvee_{a \in A} \mathbf{und}(a)) \rangle$. $\mathsf{PV}_{\mathtt{gr}}$ is in P since it suffices to check that the input labelling $\mathcal{L}$ is the grounded labelling of the underlying AF and that $\{\mathcal{L}\} \models \varphi$ (in P, cf. Proposition 2). □

Theorem 2. *For any EAF,* NV_σ *is:*

- *in P for* $\sigma = \mathtt{gr}$*;*
- *co*NP*-complete for* $\sigma \in \{\mathtt{co}, \mathtt{st}\}$*;*
- Π_2^p*-complete for* $\sigma \in \{\mathtt{pr}, \mathtt{sst}\}$.

As shown in Table 1, the complexity of the verification problems for EAF increases of one level in the polynomial hierarchy w.r.t. that for AF. Moreover, while the complexity of the possible verification problem is the same as that for the corresponding problem for iAF, the complexity of the necessary verification problem for EAF increases of one level in the polynomial hierarchy w.r.t. that for iAF (that coincides with that for AF).

	AF	iAF		EAF	
σ	V_σ	PV_σ	NV_σ	PV_σ	NV_σ
$\mathtt{co}$	P	NP-c	P	NP-c	coNP-c
$\mathtt{st}$	P	NP-c	P	NP-c	coNP-c
$\mathtt{pr}$	coNP-c	Σ_2^p-c	coNP-c	Σ_2^p-c	Π_2^p-c
$\mathtt{sst}$	coNP-c	Σ_2^p-c	coNP-c	Σ_2^p-c	Π_2^p-c

Table 1: Complexity of the verification problems for AF, iAF and EAF under semantics $\sigma \in \{\mathtt{co}, \mathtt{st}, \mathtt{pr}, \mathtt{sst}\}$. For any complexity class C, C-c means C-complete. New results are highlighted in grey.

4 Existence Problems in EAF

The existence of solutions in EAF corresponds to determine the existence of σ-epistemic labelling sets. As there could be several σ-epistemic labelling sets, we consider two problems, namely *possible existence* and *necessary existence*, respectively checking whether *i*) there exists a not empty σ-epistemic labelling set, and *ii*) all σ-epistemic labelling sets are not empty. Moreover, for each problem, we also consider the non-empty labelling variant, namely *possible non-empty existence* and *necessary non-empty existence*, respectively checking whether *i*) there exists a not empty σ-epistemic labelling set containing a non-empty σ-labelling, and *ii*) all σ-epistemic labelling sets contain a non-empty σ-labelling. Herein, for an *empty labelling* we mean a labelling where all arguments are labelled as undecided. Observe that if a labelling prescribes that an argument a is not labelled as undecided, then it must also prescribe that there is an argument b (not necessarily distinct from a) which is labelled $\mathbf{in}$. Therefore, we say that a labelling $\mathcal{L}$ is empty iff $\mathbf{und}(\mathcal{L})$ is the whole set of arguments, which in turn means that $\mathbf{in}(\mathcal{L})$ is empty.

Definition 4 (Possible/Necessary Existence)**.** Let $\Delta = \langle \mathrm{A}, \mathrm{R}, \varphi \rangle$ be an EAF and $\sigma \in \{\mathtt{gr}, \mathtt{co}, \mathtt{st}, \mathtt{pr}, \mathtt{sst}\}$ a semantics,

1. the *possible existence* problem, denoted as PEX_σ, consists in deciding whether there exists a σ-epistemic labelling set $\mathcal{L}^S$ for Δ such that $\mathcal{L}^S \neq \emptyset$;
2. the *possible non-empty existence* problem, denoted as $\mathsf{PEX}_\sigma^{\neg\emptyset}$, consists in deciding whether there exists a σ-epistemic labelling set for Δ having at least one σ-labelling $\mathcal{L}$ such that $\mathbf{in}(\mathcal{L}) \neq \emptyset$;
3. the *necessary existence* problem, denoted as NEX_σ, consists in deciding whether for all σ-epistemic labelling set $\mathcal{L}^S$ for Δ it holds that $\mathcal{L}^S \neq \emptyset$;
4. the *necessary non-empty existence* problem, denoted as $\mathsf{NEX}_\sigma^{\neg\emptyset}$, consists in deciding whether all σ-epistemic labelling sets for Δ have at least one σ-labelling $\mathcal{L}$ such that $\mathbf{in}(\mathcal{L}) \neq \emptyset$.

Given an EAF Δ, we use $\mathsf{PEX}_\sigma(\Delta)$ (resp. $\mathsf{PEX}^{\neg\emptyset}(\Delta)$, $\mathsf{NEX}_\sigma(\Delta)$, $\mathsf{NEX}^{\neg\emptyset}(\Delta)$) to denote the output of problem PEX_σ (resp. $\mathsf{PEX}^{\neg\emptyset}$, NEX_σ, $\mathsf{NEX}^{\neg\emptyset}$) over such instance.

In the rest of this section, we investigate the complexity of the existence problems in EAF. We start by presenting some results that will be useful to characterize the complexity of the considered problems.

We first observe that possible and necessary existence in EAF coincide. Intuitively, this is due to the fact, by definition, σ-epistemic labelling sets enjoy a $\subseteq$-maximal property entailing that none of them can be contained in a non-empty σ-epistemic labelling set, if it exists.

Proposition 1. *For any EAF* Δ *and semantics* $\sigma \in \{\mathtt{gr}, \mathtt{co}, \mathtt{st}, \mathtt{pr}, \mathtt{sst}\}$, $\mathsf{PEX}_\sigma(\Delta) = \mathsf{NEX}_\sigma(\Delta)$.

Proof. Clearly, if $\mathsf{NEX}_\sigma(\Delta)$ is true then $\mathsf{PEX}_\sigma(\Delta)$ is true. If $\mathsf{PEX}_\sigma(\Delta)$ is true, and thus there is a σ-epistemic labelling set $\mathcal{L}^S$ for Δ such that $\mathcal{L}^S \neq \emptyset$, then every σ-epistemic labelling set for Δ is

	AF		iAF				EAF		
σ	EX_σ	$\mathsf{EX}_\sigma^{\neg\emptyset}$	PEX_σ	NEX_σ	$\mathsf{PEX}_\sigma^{\neg\emptyset}$	$\mathsf{NEX}_\sigma^{\neg\emptyset}$	PEX_σ/NEX_σ	$\mathsf{PEX}_\sigma^{\neg\emptyset}$	$\mathsf{NEX}_\sigma^{\neg\emptyset}$
co	T	NP-c	T	T	NP-c	Π_2^p-c	NP-c	NP-c	NP-c
st	NP-c	NP-c	NP-c	Π_2^p-c	NP-c	Π_2^p-c	NP-c	NP-c	NP-c
pr	T	NP-c	T	T	NP-c	Π_2^p-c	Σ_2^p-c	Σ_2^p-c	Σ_2^p-c
sst	T	NP-c	T	T	NP-c	Π_2^p-c	Σ_2^p-c	Σ_2^p-c	Σ_2^p-c

Table 2: Complexity of the verification and existence problems for AF, iAF and EAF under complete (co), stable (st), preferred (pr), and semi-stable (sst) semantics. For any complexity class C, C-c means C-complete. T stands for *trivial*. New results are highlighted in grey.

not empty, otherwise it would not be a $\subseteq$-maximal set of σ-labellings of the underlying AF that satisfies the epistemic constraint. $\square$

Proposition 2. *Checking whether a given labelling set satisfies a given epistemic formula is decidable in polynomial time.*

Proof. Let A be a set of arguments, $\mathcal{L}^S$ a set of labellings over A, and $\varphi = \varphi_1 \vee \cdots \vee \varphi_n$ an epistemic formula, where $\varphi_i = \mathbf{K}\varphi_{i,0} \wedge \cdots \wedge \mathbf{K}\varphi_{i,k_i} \wedge \mathbf{M}\varphi_{i,k_{i+1}} \wedge \cdots \wedge \mathbf{M}\varphi_{i,m_i}$. We need to check whether there is $i \in [1, n]$, s.t. $\mathcal{L}^S \models \varphi_i$. This can be done in P as it is sufficient to check that *a*) for each $j \in [k_{i+1}, m_i]$ there is $\mathcal{L} \in \mathcal{L}^S$ that satisfies $\varphi_{i,j}$ and *b*) all $\mathcal{L} \in \mathcal{L}^S$ satisfy the formulae $\varphi_{i,l}$ with $l \in [0, k_i]$. $\square$

As stated next, the grounded-epistemic labelling set is unique.

Fact 2. *For any EAF $\langle \mathrm{A}, \mathrm{R}, \varphi \rangle$, the* gr*-epistemic labelling set is $\{\mathcal{L}(\mathtt{gr}(\langle \mathrm{A}, \mathrm{R} \rangle))\}$ if $\mathcal{L}(\mathtt{gr}(\langle \mathrm{A}, \mathrm{R} \rangle)) \models \varphi$; $\emptyset$ otherwise.*

Although the presence of constraints in EAF breaks the meet-semilattice of complete extensions, reasoning under the grounded semantics remains tractable.

Proposition 3. *Checking whether an EAF admits a non-empty grounded-epistemic labelling set can be done in polynomial time.*

Proof. Let $\Delta = \langle \mathrm{A}, \mathrm{R}, \varphi \rangle$ be an EAF. A labelling set $\mathcal{L}^S = \{\mathcal{L}\}$ of Δ is the grounded-epistemic of Δ iff $\mathbf{in}(\mathcal{L})$ is the grounded extension of $\langle \mathrm{A}, \mathrm{R} \rangle$ and $\mathcal{L}^S \models \varphi$. Computing the grounded labelling $\mathcal{L}$ is in P [33]. Checking whether $\mathcal{L}^S$ satisfies a given epistemic constraint φ is also in P, from which the statement follows. $\square$

Therefore, since if a grounded-epistemic labelling set for an EAF exists then it is unique, deciding the possible existence problem in EAF under the grounded semantics is still polynomial. This is stated in the next theorem which also states the complexity of possible existence under the multiple status semantics $\sigma \in \{\mathtt{co}, \mathtt{pr}, \mathtt{st}, \mathtt{sst}\}$. Clearly, the following results also hold for NEX_σ (cf. Proposition 1).

Theorem 3. *For any EAF, PEX_σ is:*

- *in P for $\sigma = \mathtt{gr}$;*
- *NP-complete for $\sigma \in \{\mathtt{co}, \mathtt{st}\}$;*
- *Σ_2^p-complete for $\sigma \in \{\mathtt{pr}, \mathtt{sst}\}$.*

The following theorems characterize the complexity of the possible and necessary non-empty existence problems for EAF.

Theorem 4. *For any EAF, $\mathsf{PEX}_\sigma^{\neg\emptyset}$ is:*

- *in P for $\sigma = \mathtt{gr}$;*
- *NP-complete for $\sigma \in \{\mathtt{co}, \mathtt{st}\}$;*
- *Σ_2^p-complete for $\sigma \in \{\mathtt{pr}, \mathtt{sst}\}$.*

Theorem 5. *For any EAF, $\mathsf{NEX}_\sigma^{\neg\emptyset}$ is:*

- *in P for $\sigma = \mathtt{gr}$;*
- *NP-complete for $\sigma \in \{\mathtt{co}, \mathtt{st}\}$;*
- *Σ_2^p-complete for $\sigma \in \{\mathtt{pr}, \mathtt{sst}\}$.*

The results of this section show that deciding the possible or necessary existence problem in EAF is harder than in iAF (and AF), except for stable semantics for which deciding existence in EAF behaves as in AF. Moreover, it turns out that non-emptiness is not a source of complexity in EAF (i.e. deciding non-empty existence is not harder than deciding existence), while for iAF deciding non-empty existence is generally harder than deciding existence (except for stable semantics where the existence of a solution is not guaranteed in AF).

5 Relationship between EAF and iAF

In this section, we analyze the relationship between EAF and iAF. We focus on multiple status semantics only, avoiding considering the grounded semantics that behaves differently in the two frameworks. Indeed, differently from EAF where the grounded semantics remains unique status as in AF, for iAF the grounded semantics prescribes multiple extensions (one for each completion). Thus comparing EAF and iAF under grounded semantics would mean comparing a deterministic semantics with a non-deterministic one, that, in our opinion, does not fit well with our current setting where semantics prescribing multiple solutions are meant to represent uncertain information.

The following proposition states that EAF can be used to decide possible and necessary verification over iAF. Although the result is given for a special class of iAF (i.e. arg-iAF), we recall that arg-iAF is as expressive as (general) iAF [3, 14, 13].

Let $\Delta = \langle \mathrm{A}, \mathrm{B}, \mathrm{R} \rangle$ be an arg-iAF. We use $eaf(\Delta) = \langle \mathrm{A}^*, \mathrm{R}^*, \varphi \rangle$ to denote the EAF obtained from Δ as follows:

- $\mathrm{A}^* = \mathrm{A} \cup \mathrm{B} \cup \{x_b, \overline{x_b} \mid b \in \mathrm{B}\}$;
- $\mathrm{R}^* = \mathrm{R} \cup \{(x_b, \overline{x_b}), (\overline{x_b}, x_b), (x_b, b) \mid b \in \mathrm{B}\}$; and
- $\varphi = \mathbf{K}\Big(\bigwedge_{b \in \mathrm{B}} (\neg \mathbf{und}(x_b)) \Big)$.

Theorem 6. *Let $\Delta = \langle \mathrm{A}, \mathrm{B}, \mathrm{R} \rangle$ be an iAF, $S \subseteq A \cup B$ a set of arguments, $\sigma \in \{\mathtt{co}, \mathtt{pr}, \mathtt{st}, \mathtt{sst}\}$ a semantics. For any completion $\Lambda = \langle A_\Lambda, R_\Lambda \rangle$ of Δ, it holds that $S \in \sigma(\Lambda)$ iff $\mathcal{L}(S) \cup \{\mathbf{out}(x_b), \mathbf{in}(\overline{x_b}) \mid b \in B \cap A_\Lambda\} \cup \{\mathbf{in}(x_b), \mathbf{out}(\overline{x_b}), \mathbf{out}(b) \mid b \in B \setminus A_\Lambda\}$ is a σ-labelling for $eaf(\Delta)$.*

Proof. ($\Rightarrow$) If $S \in \sigma(\Lambda)$ then $\mathcal{L}(S) \cup \{\mathbf{out}(x_b), \mathbf{in}(\overline{x_b}) \mid b \in \mathrm{B} \cap \mathrm{A}_\Lambda\} \cup \{\mathbf{in}(x_b), \mathbf{out}(\overline{x_b}), \mathbf{out}(b) \mid b \in \mathrm{B} \setminus \mathrm{A}_\Lambda\}$ is a σ-labelling for $\Delta^* = eaf(\Delta)$ as $\mathcal{L}(S) \models \varphi$. ($\Leftarrow$) There is a σ-labelling $\mathcal{L}(S')$ for Δ^* containing the label $\mathbf{in}(a)$ s.t. $a \in S$. Then, let $\Lambda = \langle \mathrm{A}_\Lambda = (A \cup \{b \in B \mid \mathbf{in}(\overline{x_b}) \in \mathcal{L}(S')\}), \mathrm{R}_\Lambda = (\mathrm{R} \cap (\mathrm{A}_\Lambda \times \mathrm{A}_\Lambda)) \rangle$ be a completion of Δ, it holds that $S \in \sigma(\Lambda)$. $\square$

Example 10. Consider the iAF Δ_7 of Example 7 and the corresponding EAF $\Delta_{10} = eaf(\Delta_7) = \langle \{\mathtt{a}, \mathtt{b}, \mathtt{c}, \mathtt{d}, \mathtt{x_c}, \overline{\mathtt{x}}_\mathtt{c}\}, \{(\mathtt{a}, \mathtt{b}), (\mathtt{b}, \mathtt{c}), (\mathtt{c}, \mathtt{d}), (\mathtt{d}, \mathtt{c}), (\mathtt{x_c}, \overline{\mathtt{x}}_\mathtt{c}), (\overline{\mathtt{x}}_\mathtt{c}, \mathtt{x_c}), (\mathtt{x_c}, \mathtt{c})\}, \varphi = \mathbf{K}\neg\mathbf{und}(\mathtt{x_c}) \rangle$, whose underlying AF is shown in Figure 4. For $\sigma \in \{\mathtt{st}, \mathtt{pr}, \mathtt{sst}\}$, Δ_{10} has the σ-epistemic labelling set $\{\mathcal{L}_1' = \{\mathbf{in}(\mathtt{a}), \mathbf{out}(\mathtt{b}), \mathbf{out}(\mathtt{c}), \mathbf{in}(\mathtt{d}), \mathbf{out}(\mathtt{x_c}), \mathbf{in}(\overline{\mathtt{x}}_\mathtt{c})\}$, $\mathcal{L}_2' = \{\mathbf{in}(\mathtt{a}), \mathbf{out}(\mathtt{b}), \mathbf{in}(\mathtt{c}), \mathbf{out}(\mathtt{d}), \mathbf{out}(\mathtt{x_c}), \mathbf{in}(\overline{\mathtt{x}}_\mathtt{c})\}$, $\mathcal{L}_1'' = \{\mathbf{in}(\mathtt{a}), \mathbf{out}(\mathtt{b}), \mathbf{out}(\mathtt{c}), \mathbf{in}(\mathtt{d}), \mathbf{in}(\mathtt{x_c}), \mathbf{out}(\overline{\mathtt{x}}_\mathtt{c})\}$, whose labellings correspond (modulo arguments $\mathtt{x_c}$ and $\overline{\mathtt{x}}_\mathtt{c}$) to σ-labellings of Δ_7. $\square$

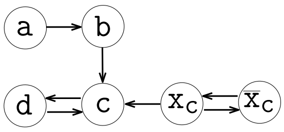

Figure 4: AF underlying EAF Δ_{10}, corresponding to iAF of Example 7.

From the result of Theorem 6 we have that EAF can encode iAF possible/necessary verification. In fact, according to the results of Section 3, verification in EAF is more expressive than in iAF for each considered semantics (cf. Table 2).

6 Related Work

Work on epistemic logic dates back to the early 1860s. Since then epistemic logic has played an important role also in computer science. This field is very active and important results are reported in a recent book surveying state-of-the-art research [58]. Epistemic Logic extends propositional logic by allowing to also express knowledge of agents, also called subjective knowledge. The idea of extending logic with epistemic constructs has been investigated also in the field of Answer Set Programming (ASP) [42, 38]. Epistemic logic programs, firstly proposed in [42], extend disjunctive logic programs under the stable model semantics with modal constructs called subjective literals [43, 22, 23, 38]. In such a context, several problems are still open and they regard the support required by stable models, as well as splitting properties that are satisfied by classical ASP semantics, but not satisfied by epistemic ASP-based semantics [56, 23, 46].

Constraints have been also used in the context of dynamic AFs to refer to the enforcement of some change [32]. In this context, extension enforcement aims at modifying an AF to ensure that a given set of arguments becomes (part of) an extension for the chosen semantics [12, 11, 60, 52]. This is different from the EAF approach [55] where epistemic constraints are used to discard unfeasible solutions (i.e. labellings/extensions), without enforcing that a new set of arguments becomes an extension.

As also discussed in [55], a difference between Constrained AF [29] and EAF concerns the meaning of constraints. Indeed, constraints in CAF are imposed on the admissibility of sets of arguments (i.e. over admissible sets) that are at the basis of σ-extensions, with $\sigma \in \{\mathtt{gr}, \mathtt{co}, \mathtt{pr}, \mathtt{st}, \mathtt{sst}\}$. As a consequence, a drawback of this approach is that σ-extensions of CAF are no longer guaranteed to be σ-extensions of the underlying AF, that is, we may have $E \in \sigma(\langle \mathrm{A}, \mathrm{R}, \mathrm{C}\rangle) \setminus \sigma(\langle \mathrm{A}, \mathrm{R}\rangle)$. Differently, EAF prescribe σ-labellings that are σ-labellings of underlying AF.

AF with epistemic attacks (EAAF) has been recently introduced in [4], where new types of epistemic AF attacks are considered. While in EAF the labelings of the underlying AF satisfying constraints are grouped into (multiple) epistemic labeling sets, EAAF extends AF by considering three kinds of attacks (classical, weak epistemic, and strong epistemic) and extends the concepts of defeated and acceptable argument. The two frameworks are significantly different, as confirmed by the different complexity results obtained.

The relationship between epistemic constraints and preferences has been explored in [55], where it is shown that EAF enables us to specify a kind of preferences over not only arguments but also justification states of arguments. Dung's framework has been extended in several ways for allowing preferences over arguments [6, 50]. In particular, preferences relying to so-called critical attacks, i.e. attacks from a less preferred argument to a more preferred one, can be handled by removing or invalidating such attacks or by "inverting" them [8]. Such kind of preferences can be encoded into EAF, possible through reductions relying on additional arguments and attacks [47].

Preferences can be also expressed in value-based AFs [15], where each argument is associated with a numeric value, and a set of possible orders (preferences) among the values is defined. In [35] weights are associated with attacks, and new semantics extending the classical ones on the basis of a given numerical threshold are proposed. [30] extends [35] by considering other aggregation functions over weights apart from sum. Except for weighted solutions under grounded semantics (that prescribes more than one weighted solution), the complexity of the main reasoning tasks in the above-considered AF-based frameworks is lower than that of EAFs, which suggests that EAFs are more expressive and can be used to model those frameworks (we plan to formally investigate these connections in future work).

7 Conclusions and Future Work

We have investigated the complexity of the existence and verification problems in EAF, where epistemic constraints are expressed by using modal operators. It turns out that verification in iAF can be reduced to verification in EAF, providing a connection between these two AF-based frameworks. It is worth noting that the connection between AF, iAF and EAF carry over to other AF-based frameworks that can be mapped to AF, such as Bipolar AF [26] and AF with recursive attacks and supports [27], among others [59, 45, 1]. However, despite the relationship concerning verification, our complexity analysis also shows that possible and necessary (non-empty) existence behave quite differently for iAF and EAF—this is intuitively due to the different semantics of the two frameworks. For instance, under standard complexity assumption, for some semantics, necessary non-empty existence in iAF cannot be reduced to the corresponding problem in EAF, while possible existence in EAF cannot be reduced to the corresponding problem in iAF.

As in the case of other AF-based frameworks, the complexity of the verification and existence problems are preparatory for the investigation of that of the credulous and skeptical acceptance problems, as e.g. membership algorithms for the verification problem can be used or adapted to solve these problems—we plan to investigate the complexity of EAF acceptability problems in future work. Additional future work will be devoted to considering more general forms of epistemic constraints, such as epistemic constraints allowing to express conditions on aggregates (e.g. the agent believe that at least n arguments from a given set S should be accepted/rejected). Moreover, we plan to explore epistemic constraints in structured argumentation formalisms [20, 41]. Finally, concerning possible implementations, it is worth noting that SAT-based CEGAR algorithms have been successfully used for solving various Σ_2^p-complete problems, including e.g. stable conclusions in ASPIC+ [49] and acceptance in iAF [13]. This suggests that, following that approach, EAF existence and verification problems could be addressed in a similar way. Alternatively, considering the tight relationship between AF and Answer Set Programming (ASP) (see [37]), EAF existence and verification problems could be solved by mapping EAF into Epistemic ASP (EASP) and using current EASP solvers [48, 16].

Acknowledgements

The research reported in the paper was partially supported by the PNRR project "Tech4You (ECS00000009) - Spoke 6", under the NRRP MUR program funded by the NextGenerationEU, and by the PNRR project FAIR - Future AI Research (PE00000013).

References

[1] G. Alfano, A. Cohen, S. Gottifredi, S. Greco, F. Parisi, and G. Simari, 'Dynamics in abstract argumentation frameworks with recursive attack and support relations', in *Proc. of ECAI*, (2020).

[2] G. Alfano, S. Greco, F. Parisi, and I. Trubitsyna, 'Argumentation frameworks with strong and weak constraints: Semantics and complexity', in *Proc. of AAAI*, pp. 6175–6184, (2021).

[3] G. Alfano, S. Greco, F. Parisi, and I. Trubitsyna, 'Incomplete argumentation frameworks: Properties and complexity', in *Proc. of AAAI*, pp. 5451–5460, (2022).

[4] G. Alfano, S. Greco, F. Parisi, and I. Trubitsyna, 'Epistemic abstract argumentation framework: Formal foundations, computation and complexity', in *Proc. of AAMAS*, pp. 409–417, (2023).

[5] L. Amgoud and C. Cayrol, 'On the acceptability of arguments in preference-based argumentation', in *Proc. of UAI*, pp. 1–7, (1998).

[6] L. Amgoud and C. Cayrol, 'Inferring from inconsistency in preference-based argumentation frameworks', *J. Aut. Reas.*, **29**(2), 125–169, (2002).

[7] L. Amgoud, Y. Dimopoulos, and P. Moraitis, 'A unified and general framework for argumentation-based negotiation', in *Proc. of AAMAS*, p. 158, (2007).

[8] Leila Amgoud and Srdjan Vesic, 'Rich preference-based argumentation frameworks', *Int. J. Approx. Reason.*, **55**(2), 585–606, (2014).

[9] O. Arieli, 'Conflict-free and conflict-tolerant semantics for constrained argumentation frameworks', *J. Appl. Log.*, **13**(4), 582–604, (2015).

[10] P. Baroni, M. Caminada, and M. Giacomin, 'An introduction to argumentation semantics', *Knowl. Eng. Rev.*, **26**(4), 365–410, (2011).

[11] R. Baumann, 'Normal and strong expansion equivalence for argumentation frameworks', *Artif. Intell.*, **193**, 18–44, (2012).

[12] R. Baumann and G. Brewka, 'Expanding argumentation frameworks: Enforcing and monotonicity results', in *Proc. of COMMA*, pp. 75–86, (2010).

[13] D. Baumeister, M. Järvisalo, D. Neugebauer, A. Niskanen, and J. Rothe, 'Acceptance in incomplete argumentation frameworks', *Artif. Intell.*, 103470, (2021).

[14] D. Baumeister, D. Neugebauer, J. Rothe, and H. Schadrack, 'Verification in incomplete argumentation frameworks', *AI*, **264**, 1–26, (2018).

[15] T. J. M. Bench-Capon, 'Persuasion in practical argument using value-based argumentation frameworks', *J. Log. Comp.*, **13**(3), 429–448, (2003).

[16] M. Bichler, M. Morak, and S. Woltran, 'Single-shot epistemic logic program solving', in *Proc. of IJCAI*, pp. 1714–1720, (2018).

[17] S. Bistarelli, F. Rossi, and F. Santini, 'A novel weighted defence and its relaxation in abstract argumentation', *Int. J. of Approx. Reas.*, **92**, 66–86, (2018).

[18] S. Bistarelli and F. Santini, 'Well-foundedness in weighted argumentation frameworks', in *Proc. of JELIA*, pp. 69–84, (2019).

[19] S. Bistarelli and F. Santini, 'Weighted argumentation', *IfCoLog Journal of Logics and their Applications*, **8**(6), 1589–1622, (2021).

[20] A. Bondarenko, P. M. Dung, R. A. Kowalski, and F. Toni, 'An abstract, argumentation-theoretic approach to default reasoning', *Artif. Intell.*, **93**, 63–101, (1997).

[21] M. C. D. Budán, M. L. Cobo, D. C. Martínez, and G. R. Simari, 'Proximity semantics for topic-based abstract argumentation', *Information Sciences*, **508**, 135–153, (2020).

[22] P. Cabalar, J. Fandinno, and L. F. del Cerro, 'Autoepistemic answer set programming', *Artif. Intell.*, **289**, 103382, (2020).

[23] P. Cabalar, J. Fandinno, and L. F. del Cerro, 'Splitting epistemic logic programs', *Theory Pract. Log. Program.*, **21**(3), 296–316, (2021).

[24] M. Caminada, 'Semi-stable semantics', in *Proc. of COMMA*, pp. 121–130, (2006).

[25] C. Cayrol, A. Cohen, and M.-C. Lagasquie-Schiex, 'Higher-order interactions (bipolar or not) in abstract argumentation: A state of the art', *IfCoLog J. of Log. and their App.*, **8**(6), 1339–1436, (2021).

[26] A. Cohen, S. Gottifredi, A. J. García, and G. R. Simari, 'A survey of different approaches to support in argumentation systems', *Knowl. Eng. Rev.*, **29**(5), 513–550, (2014).

[27] A. Cohen, S. Gottifredi, A. J. Garcia, and G. R. Simari, 'An approach to abstract argumentation with recursive attack and support', *J. Appl. Log.*, **13**(4), 509–533, (2015).

[28] A. Cohen, S. Parsons, E. I. Sklar, and P. McBurney, 'A characterization of types of support between structured arguments and their relationship with support in abstract argumentation', *International Journal of Approximate Reasoning*, **94**, 76–104, (2018).

[29] S. Coste-Marquis, C. Devred, and P. Marquis, 'Constrained argumentation frameworks', in *Proc. of (KR)*, pp. 112–122, (2006).

[30] S. Coste-Marquis, S. Konieczny, P. Marquis, and M. Akli Ouali, 'Weighted attacks in argumentation frameworks', in *Proc. KR*, (2012).

[31] Y. Dimopoulos, J.-G. Mailly, and P. Moraitis, 'Argumentation-based negotiation with incomplete opponent profiles', in *Proc. AAMAS*, pp. 1252–1260, (2019).

[32] S. Doutre and J.-G. Mailly, 'Constraints and changes: A survey of abstract argumentation dynamics', *Arg. & Comp.*, **9**(3), 223–248, (2018).

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

[34] P. E. Dunne and T. J. M. Bench-Capon, 'Coherence in finite argument systems', *Artif. Intell.*, **141**(1/2), 187–203, (2002).

[35] P. E. Dunne, A. Hunter, P. McBurney, S. Parsons, and M. J. Wooldridge, 'Weighted argument systems: Basic definitions, algorithms, and complexity results', *Artif. Intell.*, **175**(2), 457–486, (2011).

[36] W. Dvorák and P. E. Dunne, 'Computational problems in formal argumentation and their complexity', *FLAP*, **4**(8), (2017).

[37] U. Egly, S. A. Gaggl, and S. Woltran, 'ASPARTIX: implementing argumentation frameworks using answer-set programming', in *Proc. of ICLP*, pp. 734–738, (2008).

[38] J. Fandinno, W. Faber, and M. Gelfond, 'Thirty years of epistemic specifications', *CoRR*, **abs/2108.07669**, (2021).

[39] B. Fazzinga, S. Flesca, and F. Furfaro, 'Revisiting the notion of extension over incomplete abstract argumentation frameworks', in *Proc. of IJCAI*, pp. 1712–1718, (2020).

[40] *Handbook of Formal Argumentation*, eds., D. Gabbay, M. Giacomin, G. R. Simari, and M. Thimm, volume 2, College Public., 2021.

[41] A. J. Garcia, H. Prakken, and G. R. Simari, 'A comparative study of some central notions of ASPIC+ and DeLP', *Theory Pract. Log. Program.*, **20**(3), 358–390, (2020).

[42] M. Gelfond, 'Strong introspection', in *Proc. of AAAI*, (1991).

[43] M. Gelfond, 'New semantics for epistemic specifications', in *Proc. of LPNMR Conf.*, volume 6645, pp. 260–265, (2011).

[44] M. G. Escañuela Gonzalez, M. C. D. Budán, G. I. Simari, and G. R. Simari, 'Labeled bipolar argumentation frameworks', *Journal of Artificial Intelligence Research*, **70**, 1557–1636, (2021).

[45] S. Gottifredi, A. Cohen, A. J. Garcia, and G. R. Simari, 'Characterizing acceptability semantics of argumentation frameworks with recursive attack and support relations', *Artif. Intell.*, **262**, 336–368, (2018).

[46] A. Herzig and A. Yuste-Ginel, 'On the epistemic logic of incomplete argumentation frameworks', in *Proc. of KR*, pp. 681–685, (2021).

[47] S. Kaci, L. W. N. van der Torre, and S. Villata, 'Preference in abstract argumentation', in *Proc. COMMA*, pp. 405–412, (2018).

[48] A. P. Leclerc and P. T. Kahl, 'A survey of advances in epistemic logic program solvers', *CoRR*, **abs/1809.07141**, (2018).

[49] T. Lehtonen, J.P. Wallner, and M. Järvisalo, 'Computing stable conclusions under the weakest-link principle in the ASPIC+ argumentation formalism', in *Proc. of KR*, (2022).

[50] S. Modgil, 'Reasoning about preferences in argumentation frameworks', *Artif. Intell.*, **173**(9-10), 901–934, (2009).

[51] S. Modgil and H. Prakken, 'A general account of argumentation with preferences', *Artif. Intell.*, **195**, 361–397, (2013).

[52] A. Niskanen, J. P. Wallner, and M. Järvisalo, 'Extension enforcement under grounded semantics in abstract argumentation', in *Proc. KR*, pp. 178–183, (2018).

[53] C. H. Papadimitriou, *Computational complexity*, Addison-Wesl, 1994.

[54] H. Prakken, 'Models of persuasion dialogue', in *Argumentation in Artificial Intelligence*, 281–300, (2009).

[55] C. Sakama and T. C. Son, 'Epistemic argumentation framework: Theory and computation', *J. Artif. Intell. Res.*, **69**, 1103–1126, (2020).

[56] Y.-D. Shen and T. Eiter, 'Constraint monotonicity, epistemic splitting and foundedness are too strong in answer set programming', *CoRR*, **abs/2010.00191**, (2020).

[57] K. Skiba, D. Neugebauer, and J. Rothe, 'Complexity of nonempty existence problems in incomplete argumentation frameworks', *IEEE Intell. Syst.*, **36**(2), 13–24, (2021).

[58] *Handbook of Epistemic Logic*, eds., H. van Ditmarsch, J. Y. Halpern, W. van der Hoek, and B. P. Kooi, kings College Pubn, 2015.

[59] S. Villata, G. Boella, D. M. Gabbay, and L. W. N. van der Torre, 'Modelling defeasible and prioritized support in bipolar argumentation', *Ann. Math. Artif. Intell.*, **66**(1-4), (2012).

[60] J. P. Wallner, A. Niskanen, and M. Järvisalo, 'Complexity results and algorithms for extension enforcement in abstract argumentation', *J. of Artif. Intell. Res.*, **60**, 1–40, (2017).