On the 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 (DIMES),
University of Calabria, Rende, Italy

Abstract

Dung's Argumentation Framework (AF) has been extended in several directions. An interesting extension, among others, is the *Epistemic Argumentation Framework* (EAF) which allows representing the agent's belief by means of epistemic constraints. An epistemic constraint is a propositional formula over labeled arguments (e.g. **in**(a)) extended with the modal operators **K** and **M** that intuitively state that the agent believes that a given formula is certainly or possibly true, respectively. In this paper, we discuss interesting results recently presented in [1] concerning the complexity of three canonical argumentation problems (i.e. *verification*, *existence*, and *non-empty existence*) in the context of EAF as well as the relationship between EAF and *incomplete AF*, an extension of AF where arguments and attacks may be uncertain.

Keywords

Formal Argumentation Theory, Epistemic Argumentation Framework, Computational Complexity.

1. Introduction

In the last decades, Formal Argumentation has become an important research field in AI [2]. Argumentation has potential applications in several contexts, including e.g. modeling dialogues, negotiation [3, 4], and persuasion [5]. Dung's Argumentation Framework (AF) is a simple yet powerful formalism which is suitable, for instance, for modeling disputes between two or more agents [6]. 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) [6, 7]—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 $\sigma \in \{\text{gr}, \text{co}, \text{pr}, \text{st}, \text{sst}\}$.

Example 1. Consider AF $\Lambda_1 = \langle A_1 = \{a, b, c, d\}, R_1 = \{(a, b), (b, c), (c, d), (d, c)\}\rangle$ 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.
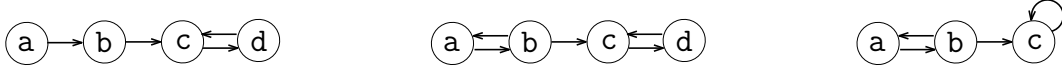
Figure 1: AF Λ_1 of Example 1 (left); AF Λ_3 of Example 3 (center); AF Λ_5 of Example 5 (right).

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 $E_1 = \{a, c\}$ and $E_2 = \{a, d\}$, meaning that either Alice and Carl, or Alice and David will attend the party. □

Argumentation semantics can also be defined in terms of *labelling* [8]. Intuitively, a σ-labelling for an AF is a total function $\mathcal{L}$ assigning to each argument the label **in** if it is accepted, **out** if it is rejected, and **und** if it is undecided under σ. For instance, $\mathcal{L}_1 = \{\mathbf{in}(a), \mathbf{out}(b), \mathbf{in}(c), \mathbf{out}(d)\}$ and $\mathcal{L}_2 = \{\mathbf{in}(a), \mathbf{out}(b), \mathbf{out}(c), \mathbf{in}(d)\}$ 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; this is somehow related to the fact that AF is often viewed as a tool for acceptability evaluation, rather than for directly model domain knowledge as in structured formalisms like ASPIC+ and DeLP [9, 10]. For this reason, Dung's framework has been extended by introducing further constructs, such as preferences [11, 12, 13, 14, 15], weights [16, 17, 18], supports [19, 20, 21, 22], topics [23], constraints [24, 25, 26, 27, 28, 29], qualitative [30, 31, 32] and quantitative uncertainty [33, 34, 35, 36, 37], and general acceptance conditions [38, 39]. Adding these features makes the formalism less abstract and more expressive, and hence closer to knowledge representation. Each of these additions aims to achieve a more comprehensive, natural, and compact way of 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) [26]. 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. $\mathbf{in}(a)$, $\mathbf{out}(c)$) 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 $\Lambda_1 = \langle A_1, R_1 \rangle$ of Example 1, and assume that the party planner believes that Carl will certainly join the party. This can be modeled by EAF $\Delta_2 = \langle A_1, R_1, \varphi \rangle$, where the epistemic constraint $\varphi = \mathbf{Kin}(c)$ 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. □

In the previous example we have a single σ-epistemic labelling sets. However, in general, an EAF may have multiple σ-epistemic labelling sets as shown in the following example.

Example 3. Consider the AF $\Lambda_3 = \langle A_3, R_3 \rangle$ shown in Figure 1 (center), where $A_3 = A_1$, $R_3 = R_1 \cup \{(b, a)\}$, and A_1 and R_1 are as defined in Example 1. The set of its σ-labellings with $\sigma \in \{st, pr, sst\}$ 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 $\mathcal{L}_3 = \{\mathbf{out}(a), \mathbf{in}(b), \mathbf{out}(c), \mathbf{in}(d)\}$. Then, EAF $\Delta_3 = \langle A_3, R_3, \mathbf{Kin}(a) \vee \mathbf{Kin}(d) \rangle$ 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. $\square$

Moreover, 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. $\sigma \in \{gr, co, pr, sst\}$), we can have EAFs having no non-empty σ-epistemic labelling set.

Example 4. The EAF $\Delta_4 = \langle A_3, R_3, \mathbf{K}(\mathbf{in}(a) \wedge \mathbf{in}(b)) \rangle$ (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 $\sigma \in \{gr, co, st, pr, sst\}$. $\square$

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 [40] as well as for several frameworks extending AF, such as *incomplete AF* (iAF) [41, 30, 42], 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 present 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 $\sigma \in \{gr, co, st, pr, sst\}$. 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 discuss 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, we review the AF-based frameworks considered in the paper.

2.1. Argumentation Framework

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

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

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

Given an AF $\langle A, R \rangle$, a set $S \subseteq A$ of arguments is said to be *conflict-free* iff $S \cap Def(S) = \emptyset$; and *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* [6, 7]. 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 $\langle A, R \rangle$, a set $S \subseteq A$ is an *extension* called:

- *complete* (co) iff it is an admissible set and $S = Acc(S)$;
- *preferred* (pr) iff it is a $\subseteq$-maximal complete extension;
- *stable* (st) iff it is a total preferred extension, i.e. a preferred extension s.t. $S \cup Def(S) = A$;
- *semi-stable* (sst) iff it is a preferred extension s.t. $S \cup Def(S)$ is maximal (w.r.t. $\subseteq$);
- *grounded* (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 co(Λ) (resp. pr(Λ), st(Λ), sst(Λ), gr(Λ)). It is well-known that the set of complete extensions forms a complete semilattice w.r.t. $\subseteq$, where gr(Λ) 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 $\mathrm{gr}(\Lambda)$ to denote the grounded extension. For any AF Λ the following inclusion relations hold: *i)* $\mathrm{st}(\Lambda) \subseteq \mathrm{sst}(\Lambda) \subseteq \mathrm{pr}(\Lambda) \subseteq \mathrm{co}(\Lambda)$, *ii)* $\mathrm{gr}(\Lambda) \in \mathrm{co}(\Lambda)$, and *iii)* $\mathrm{st}(\Lambda) \neq \emptyset$ implies that $\mathrm{st}(\Lambda) = \mathrm{sst}(\Lambda)$. 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).

Labelling

The argumentation semantics can be also defined in terms of *labelling* [8]. 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 $\sigma \in \{\mathrm{co}, \mathrm{pr}, \mathrm{st}, \mathrm{sst}, \mathrm{gr}\}$ 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.

Example 5. Let $\Lambda_5 = \langle \mathrm{A}_5, \mathrm{R}_5 \rangle$ be an AF where $\mathrm{A}_5 = \{\mathsf{a}, \mathsf{b}, \mathsf{c}\}$ and $\mathrm{R}_5 = \{(\mathsf{a}, \mathsf{b}), (\mathsf{b}, \mathsf{a}), (\mathsf{b}, \mathsf{c}), (\mathsf{c}, \mathsf{c})\}$ whose graph is shown in Figure 1 (right). AF Λ_5 has three complete extensions: $E_1 = \emptyset, E_2 = \{\mathsf{a}\}, E_3 = \{\mathsf{b}\}$, whose corresponding complete labellings are $\mathcal{L}_1 = \langle \emptyset, \emptyset, \{\mathsf{a}, \mathsf{b}, \mathsf{c}\} \rangle, \mathcal{L}_2 = \langle \{\mathsf{a}\}, \{\mathsf{b}\}, \{\mathsf{c}\} \rangle$, and $\mathcal{L}_3 = \langle \{\mathsf{b}\}, \{\mathsf{a}, \mathsf{c}\}, \emptyset \rangle$. 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 pr-labelling set is $\{\mathcal{L}_2, \mathcal{L}_3\}$, the st- and sst-labelling set is $\{\mathcal{L}_3\}$, while the gr-labelling set is $\{\mathcal{L}_1\}$. $\qquad \square$

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 $\sigma \in \{\mathrm{gr}, \mathrm{co}, \mathrm{st}, \mathrm{pr}, \mathrm{sst}\}$:

- 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 [40] for a survey) and is summarized in Tables 1 and 2.

2.2. Incomplete Argumentation Framework

We now recall the incomplete argumentation framework [41, 43].

Definition 1 (iAF). An *incomplete AF* (iAF) is a tuple $\langle A, B, R, T \rangle$, where A and B are disjoint sets of arguments, and R and T are disjoint sets of attacks between arguments in $A \cup B$. 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 $\langle A, B, R, T \rangle$ is said to be an *arg-iAF* iff $T = \emptyset$, i.e. it does not contain uncertain attacks. We may omit the empty set T and use a triple $\langle A, B, R \rangle$ to denote an arg-iAF.

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

Existence problems in iAF have been investigated in [42]. 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;
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 [42].

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

Example 6. Let $\Delta_6 = \langle A_5 \setminus \{\mathtt{b}\}, \{\mathtt{b}\}, R_5, \emptyset \rangle$ be an iAF, where $\langle A_5, R_5 \rangle$ is the AF of Example 5. The completion $\langle A_5, R_5 \rangle$ 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. $\quad\square$
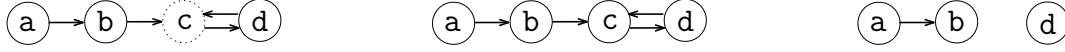
Figure 2: Arg-iAF Δ_7 of Example 7 (left), and its completions Λ'_7 (center) and Λ''_7 (right).

It is worth noting that, for any iAF, PEX_{st} is NP-complete and NEX_{st} is Π^p_2-complete. These results follow from Proposition 18 and Theorem 21 in [42] by observing that $\text{PEX}_{\text{st}}(\Delta)$ (resp. $\text{NEX}_{\text{st}}(\Delta)$) is true iff $\text{PEX}^{\neg\emptyset}_{\text{st}}(\Delta)$ (resp. $\text{NEX}^{\neg\emptyset}_{\text{st}}(\Delta)$) 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 [41, 43]. Given an iAF $\Delta = \langle \text{A}, \text{B}, \text{R}, \text{T} \rangle$, a set of arguments $S \subseteq (\text{A} \cup \text{B})$, and a semantics $\sigma \in \{\text{gr}, \text{co}, \text{pr}, \text{st}, \text{sst}\}$,

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 $\Delta_7 = \langle\{\text{a},\text{b},\text{d}\}, \{\text{c}\}, \{(\text{a},\text{b}), (\text{b},\text{c}), (\text{c},\text{d}), (\text{d},\text{c})\}, \emptyset\rangle$ whose graph is shown in Figure 2 (left), where the uncertain argument is represented by a dotted circle. Δ_7 has 2 completions: $\Lambda'_7 = \langle\{\text{a},\text{b},\text{c},\text{d}\}, \{(\text{a},\text{b}), (\text{b},\text{c}), (\text{c},\text{d}), (\text{d},\text{c})\}\rangle$ and $\Lambda''_7 = \langle\{\text{a},\text{b},\text{d}\}, \{(\text{a},\text{b})\}\rangle$, also shown in Figure 2. Under semantics $\sigma \in \{\text{st}, \text{pr}, \text{sst}\}$, AF Λ'_7 has two extensions, $E_1 = \{\text{a},\text{d}\}$ and $E_2 = \{\text{a},\text{c}\}$, 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. $\qquad \square$

The complexity of PV_σ and NV_σ for iAF has been investigated in [41] for $\sigma \in \{\text{gr}, \text{co}, \text{st}, \text{pr}\}$ and in [31] for the semi-stable semantics. The complexity results are summarized in Table 1.

2.3. Epistemic Argumentation Framework

In this section, we review the *Epistemic Argumentation Framework* [26], which extends Dungs' framework with epistemic constraints.

Given an AF $\Lambda = \langle \text{A}, \text{R} \rangle$, 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 $\lambda_A = \{\mathbf{in}(a), \mathbf{out}(a), \mathbf{und}(a) \mid a \in \text{A}\}$ 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).

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 t (`true`), and every atom not occurring in $\mathcal{L}(S)$ with f (`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:

	AF	iAF		EAF	
σ	V_σ	PV_σ	NV_σ	PV_σ	NV_σ
co	P	NP-c	P	NP-c	coNP-c
st	P	NP-c	P	NP-c	coNP-c
pr	coNP-c	Σ_2^p-c	coNP-c	Σ_2^p-c	Π_2^p-c
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.

- $\varphi = \mathtt{t}$,
- $\varphi = \mathbf{K}\psi$ 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 $\mathcal{L}^S \models \mathbf{K}\neg\varphi$,
- $\mathcal{L}^S \models \mathbf{M}(\varphi_1 \vee \varphi_2)$ iff $\mathcal{L}^S \models \mathbf{M}\varphi_1$ or $\mathcal{L}^S \models \mathbf{M}\varphi_2$,
- $\mathcal{L}^S \models \neg\mathbf{K}\varphi$ iff $\mathcal{L}^S \models \mathbf{M}\neg\varphi$,
- $\mathcal{L}^S \models \mathbf{K}(\varphi_1 \wedge \varphi_2)$ 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 $\emptyset$). 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, $\mathcal{L} \not\models (\mathbf{Kin}(\mathtt{a}) \vee \mathbf{Kin}(\mathtt{d}))$. □

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 discuss their complexity.

σ	AF		iAF				EAF		
	EX_σ	$\mathrm{EX}_\sigma^{\neg\emptyset}$	PEX_σ	NEX_σ	$\mathrm{PEX}_\sigma^{\neg\emptyset}$	$\mathrm{NEX}_\sigma^{\neg\emptyset}$	$\mathrm{PEX}_\sigma/\mathrm{NEX}_\sigma$	$\mathrm{PEX}_\sigma^{\neg\emptyset}$	$\mathrm{NEX}_\sigma^{\neg\emptyset}$
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.

Definition 3 (Possible/Necessary Verification). Given an EAF Δ, a semantics $\sigma \in \{\mathrm{gr}, \mathrm{co}, \mathrm{st}, \mathrm{pr}, \mathrm{sst}\}$, 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 $\mathrm{PV}_\sigma(\Delta, \mathcal{L})$ (resp. $\mathrm{NV}_\sigma(\Delta, \mathcal{L})$) to denote the output of problem PV_σ (resp. NV_σ) over such instance.

Example 9. Let $\Delta_3 = \langle A_3, R_3, \mathbf{Kin}(a) \vee \mathbf{Kin}(d) \rangle$ be the EAF of Example 3. Recall that Δ_3 has two σ-epistemic labelling sets (under semantics $\sigma \in \{\mathrm{pr}, \mathrm{st}, \mathrm{sst}\}$): $\{\mathcal{L}_1, \mathcal{L}_2\}$ 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. □

As shown in Table 1, that summarizes the complexity of the possible and necessary verification problems for EAF [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).

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 **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 $\Delta = \langle A, R, \varphi \rangle$ 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 discuss the complexity of the existence problems in EAF [1]. We start with some results that will be useful to analyze the complexity of the considered problems. We first observe that possible and necessary existence in EAF coincide, i.e. $\mathsf{PEX}_\sigma(\Delta) = \mathsf{NEX}_\sigma(\Delta)$ for any EAF Δ and semantics $\sigma \in \{\mathtt{gr}, \mathtt{co}, \mathtt{st}, \mathtt{pr}, \mathtt{sst}\}$. 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.

Moreover, it is worth noting that the grounded-epistemic labelling set is unique, that is, for any EAF $\langle A, R, \varphi \rangle$ the $\mathtt{gr}$-epistemic labelling set is $\{\mathcal{L}(\mathtt{gr}(\langle A, R \rangle))\}$ if $\mathcal{L}(\mathtt{gr}(\langle A, R \rangle)) \models \varphi$; $\emptyset$ otherwise. Although the presence of constraints in EAF breaks the meet-semilattice of complete extensions, reasoning under the grounded semantics remains tractable. Indeed, checking whether a given labelling set satisfies a given epistemic formula can be decided in polynomial time, and checking whether an EAF admits a non-empty grounded-epistemic labelling set can be done in polynomial time as well.

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.

As shown in Table 2, that summarizes the complexity of the possible and necessary non-empty existence problems for EAF, 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).
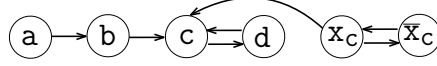
Figure 3: AF underlying EAF Δ_{10}, corresponding to iAF of Example 7.

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 result 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 [31, 41, 30]. Let $\Delta = \langle A, B, R \rangle$ be an arg-iAF. We use $eaf(\Delta) = \langle A^*, R^*, \varphi \rangle$ to denote the EAF obtained from Δ as follows: $A^* = A \cup B \cup \{x_b, \overline{x_b} \mid b \in B\}$; $R^* = R \cup \{(x_b, \overline{x_b}), (\overline{x_b}, x_b), (x_b, b) \mid b \in B\}$; and $\varphi = \mathbf{K}\Big(\bigwedge_{b \in B} \big(\neg \mathbf{und}(x_b) \big) \Big)$.

Let $\Delta = \langle A, B, R \rangle$ be an iAF, $S \subseteq A \cup B$ a set of arguments, $\sigma \in \{\texttt{co}, \texttt{pr}, \texttt{st}, \texttt{sst}\}$ 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 $eaf(\Delta)$ [1]. We illustrate it with the following example.

Example 10. Consider the iAF Δ_7 of Example 7 and the corresponding EAF $\Delta_{10} = eaf(\Delta_7) = \langle \{a, b, c, d, x_c, \overline{x}_c\}, \{(a, b), (b, c), (c, d), (d, c), (x_c, \overline{x}_c), (\overline{x}_c, x_c), (x_c, c)\}, \varphi = \mathbf{K}\neg\mathbf{und}(x_c) \rangle$, whose underlying AF is shown in Figure 3. For $\sigma \in \{\texttt{st}, \texttt{pr}, \texttt{sst}\}$, Δ_{10} has the σ-epistemic labelling set $\{\mathcal{L}'_1 = \{\mathbf{in}(a), \mathbf{out}(b), \mathbf{out}(c), \mathbf{in}(d), \mathbf{out}(x_c), \mathbf{in}(\overline{x}_c)\}, \mathcal{L}'_2 = \{\mathbf{in}(a), \mathbf{out}(b), \mathbf{in}(c), \mathbf{out}(d), \mathbf{out}(x_c), \mathbf{in}(\overline{x}_c)\}, \mathcal{L}''_1 = \{\mathbf{in}(a), \mathbf{out}(b), \mathbf{out}(c), \mathbf{in}(d), \mathbf{in}(x_c), \mathbf{out}(\overline{x}_c)\}$, whose labellings correspond (modulo arguments x_c and $\overline{x}_c$) to σ-labellings of Δ_7. □

From the above-mentioned result, 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. Discussion of Related Work and Conclusion

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 [44]. 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) [45, 46]. Epistemic logic programs, firstly proposed in [45], extend disjunctive logic programs under the stable model semantics with modal constructs

called subjective literals [47, 48, 49, 46]. 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 [50, 49, 51].

Constraints have been also used in the context of dynamic AFs to refer to the enforcement of some change [52]. 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 [53, 54, 55, 56]. This is different from the EAF approach [26] 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 [26], a difference between Constrained AF [24] 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 $\sigma \in \{\texttt{gr}, \texttt{co}, \texttt{pr}, \texttt{st}, \texttt{sst}\}$. 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 $E \in \sigma(\langle A, R, C \rangle) \setminus \sigma(\langle A, R \rangle)$. Differently, EAF prescribe σ-labellings that are σ-labellings of underlying AF.

AF with epistemic attacks (EAAF) has been recently introduced in [28], 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 [26], 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 [57, 58]. 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 [59]. Such kind of preferences can be encoded into EAF, possible through reductions relying on additional arguments and attacks [60]. 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.

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 connections between AF, iAF and EAF carry over to other AF-based frameworks that can be mapped to AF [61], such as Bipolar AF [62] and AF with recursive attacks and supports [63]. However, despite the relationship concerning verification, the 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.

References

[1] G. Alfano, S. Greco, D.Mandaglio, F. Parisi, I. Trubitsyna, Complexity of verification and existence problems in epistemic argumentation framework, in: Proc. of ECAI, 2023, pp. 77–84.

[2] D. Gabbay, M. Giacomin, G. R. Simari, M. Thimm (Eds.), Handbook of Formal Argumentation, volume 2, College Public., 2021.

[3] L. Amgoud, Y. Dimopoulos, P. Moraitis, A unified and general framework for argumentation-based negotiation, in: Proc. of AAMAS, 2007, p. 158.

[4] Y. Dimopoulos, J. Mailly, P. Moraitis, Argumentation-based negotiation with incomplete opponent profiles, in: Proc. AAMAS, 2019, pp. 1252–1260.

[5] H. Prakken, Models of persuasion dialogue, in: Argumentation in Artificial Intelligence, 2009, pp. 281–300.

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

[7] M. Caminada, Semi-stable semantics, in: Proc. of COMMA, 2006, pp. 121–130.

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

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

[10] G. Alfano, S. Greco, F. Parisi, G. I. Simari, G. R. Simari, Incremental computation for structured argumentation over dynamic delp knowledge bases, Artif. Intell. 300 (2021) 103553.

[11] L. Amgoud, C. Cayrol, On the acceptability of arguments in preference-based argumentation, in: Proc. of UAI, 1998, pp. 1–7.

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

[13] G. Alfano, S. Greco, F. Parisi, I. Trubitsyna, On preferences and priority rules in abstract argumentation, 2022.

[14] G. Alfano, S. Greco, F. Parisi, I. Trubitsyna, Abstract argumentation framework with conditional preferences, in: Proc. of AAAI, 2023, pp. 6218–6227.

[15] G. Alfano, S. Greco, F. Parisi, I. Trubitsyna, Preferences and constraints in abstract argumentation, in: Proc. of IJCAI, 2023, pp. 3095–3103.

[16] S. Bistarelli, F. Santini, Well-foundedness in weighted argumentation frameworks, in: Proc. of JELIA, 2019, pp. 69–84.

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

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

[19] A. Cohen, S. Parsons, E. I. Sklar, 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 (2018) 76–104.

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

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

[22] G. Alfano, S. Greco, F. Parisi, A meta-argumentation approach for the efficient computation of stable and preferred extensions in dynamic bipolar argumentation frameworks, Intelligenza Artificiale 12 (2018) 193–211.

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

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

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

[26] C. Sakama, T. C. Son, Epistemic argumentation framework: Theory and computation, J. Artif. Intell. Res. 69 (2020) 1103–1126.

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

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

[29] G. Alfano, S. Greco, F. Parisi, I. Trubitsyna, Stable semantics for epistemic abstract argumentation framework, in: Proc. of Arg&App@KR, 2023, pp. 11–25.

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

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

[32] G. Alfano, M. Calautti, S. Greco, F. Parisi, I. Trubitsyna, Explainable acceptance in probabilistic and incomplete abstract argumentation frameworks, Artif. Intell. 323 (2023) 103967.

[33] H. Li, N. Oren, T. J. Norman, Probabilistic argumentation frameworks, in: Proc. of TAFA, 2011, pp. 1–16.

[34] A. Hunter, Some foundations for probabilistic abstract argumentation, in: Proc. of COMMA, 2012, pp. 117–128.

[35] B. Fazzinga, S. Flesca, F. Parisi, Efficiently estimating the probability of extensions in abstract argumentation, in: Proc. of SUM, 2013, pp. 106–119.

[36] B. Fazzinga, S. Flesca, F. Parisi, On the complexity of probabilistic abstract argumentation frameworks, ACM Trans. Comput. Log. 16 (2015) 22:1–22:39.

[37] G. Alfano, M. Calautti, S. Greco, F. Parisi, I. Trubitsyna, Explainable acceptance in probabilistic abstract argumentation: Complexity and approximation, in: Proc. of KR, 2020, pp. 33–43.

[38] G. Brewka, S. Woltran, Abstract dialectical frameworks, in: Proc. of KR, 2010.

[39] G. Alfano, S. Greco, F. Parisi, I. Trubitsyna, On acceptance conditions in abstract argumentation frameworks, Inf. Sci. 625 (2023) 757–779.

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

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

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

[43] B. Fazzinga, S. Flesca, F. Furfaro, Revisiting the notion of extension over incomplete abstract argumentation frameworks, in: Proc. of IJCAI, 2020, pp. 1712–1718.

[44] H. van Ditmarsch, J. Y. Halpern, W. van der Hoek, B. P. Kooi (Eds.), Handbook of Epistemic Logic, kings College Pubn, 2015.

[45] M. Gelfond, Strong introspection, in: Proc. of AAAI, 1991.

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

[47] M. Gelfond, New semantics for epistemic specifications, in: Proc. of LPNMR Conf., volume 6645, 2011, pp. 260–265.

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

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

[50] Y. Shen, T. Eiter, Constraint monotonicity, epistemic splitting and foundedness are too strong in answer set programming, CoRR abs/2010.00191 (2020).

[51] A. Herzig, A. Yuste-Ginel, On the epistemic logic of incomplete argumentation frameworks, in: Proc. of KR, 2021, pp. 681–685.

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

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

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

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

[56] A. Niskanen, J. P. Wallner, M. Järvisalo, Extension enforcement under grounded semantics in abstract argumentation, in: Proc. KR, 2018, pp. 178–183.

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

[58] S. Modgil, Reasoning about preferences in argumentation frameworks, Artif. Intell. 173 (2009) 901–934.

[59] L. Amgoud, S. Vesic, Rich preference-based argumentation frameworks, Int. J. Approx. Reason. 55 (2014) 585–606.

[60] S. Kaci, L. W. N. van der Torre, S. Villata, Preference in abstract argumentation, in: Proc. COMMA, 2018, pp. 405–412.

[61] G. Alfano, S. Greco, F. Parisi, I. Trubitsyna, On the semantics of abstract argumentation frameworks: A logic programming approach, Theory and Practice of Logic Programming 20 (2020) 703–718.

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

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