ECAI 2020
G.D. Giacomo et al. (Eds.)

doi:10.3233/FAIA200181

Complexity of Possible and Necessary Existence Problems in Abstract Argumentation

Kenneth Skiba[1] and **Daniel Neugebauer**[2] and **Jörg Rothe**[2]

Abstract. This work focuses on generalizing the existence problems for extensions in abstract argumentation to incomplete argumentation frameworks. In this extended model, incomplete or conflicting knowledge about the state of the arguments and attacks are allowed. We propose possible and necessary variations of the existence and nonemptiness problems, originally defined for (complete) argumentation frameworks, to extend these problems to incomplete argumentation frameworks. While the computational complexity of existence problems is already known for the standard model, we provide a full analysis of the complexity for incomplete argumentation frameworks using the most prominent semantics, namely, the conflict-free, admissible, complete, grounded, preferred, and stable semantics. We show that the complexity rises from NP-completeness to Π_2^p-completeness for most "necessary" problem variants when uncertainty is allowed.

1 Introduction

Alice (A) and Bob (B) want to buy a house together. They look at two houses and are now discussing which of them they should buy, weighing up their advantages against their disadvantages.

A: *I liked the big garden in the second house. It would be great for having a barbecue with our friends.*
B: *The garden is great but also a lot of work. I like the location of the first house, so I could ride my bike to work.*
A: *Yes, that would be good, but did you see the mold in the living room? That would very unhealthy for us, or expensive to get rid of.*
B: *Nope, I didn't see that. I was just looking at the giant flat-screen TV set we would have there.*
A: *You do realize that the previous owners will take their stuff with them when they leave? So there won't be a TV set in this house when we move in. And the mold in the living room is really a problem.*
B: *Well, I've actually read that mold spores help to have a quite healthy environment: They kill off all sorts of microbes, including really dangerous human pathogenic viruses.*
A: *What? That's bullshit. Where'd you get that from?*
B: *From the internet, so it must be true.*

Discussions like this can be represented using the well established model of *abstract argumentation frameworks* introduced by Dung [13]. In this model, abstracting away from the actual contents of arguments, a discussion is illustrated by a directed graph whose vertices represent the arguments and whose edges display the attack relation among the arguments. However, Dung's model also has a few shortcomings. For example, it cannot handle incomplete information, like the argument *mold* from Alice that Bob were not aware of. But not only arguments, also attacks can be uncertain due to incomplete information. For instance, Bob believes that no attack against buying his favorite house is coming from the argument *mold*.

Whether or not a definite argument attacks another definite argument very much depends on the agents' individual beliefs, e.g., regarding causality of events. Lifting this from a private dispute among a couple to a matter of public affairs that is crucial for mankind and fiercely discussed these days, let us consider the abatement of carbon emissions. While people might agree that there is global warming, they may have different views on what is causing global warming (i.e., whether it is human-induced or not): While some might view "global warming" attacking the argument that "due to increasing energy demand there is a need to build more coal-fired power plants," others might outright deny the existence of that attack.

For such a scenario, a suitable extension of the model is needed. Baumeister et al. [5] present the concept of *incomplete argumentation framework*, which can handle uncertainty and incomplete information. The idea of uncertainty in abstract argumentation was first studied by Coste-Marquis et al. [10] for uncertain attacks. Later on, Baumeister et al. studied abstract argumentation with uncertain arguments [6] and with uncertain attacks [3]. They then generalized both ideas into one unified model of incomplete-information abstract argumentation [5]. An *incomplete argumentation framework* represents a set of possibilities—we can either include an uncertain element or leave it out of our system, which is captured by so-called *completions*.

The reasoning problems of *verification* (which refers to acceptability of a set of arguments with respect to some semantics) and of either *credulous* or *skeptical acceptance* (both referring to acceptability of a single argument with respect to some semantics) have already been studied for incomplete argumentation frameworks by Baumeister et al. [5, 4]. Completing their work, we tackle the *existence problem* and the *nonempty-existence* (or, *nonemptiness*, for short) *problem* of semantics in incomplete argumentation frameworks. The (nonempty) existence problem refers to the question of whether, given an argumentation framework and a semantics, there exists an acceptable (or, a nonempty acceptable) set of arguments with respect to the given semantics in the first place. While this is trivial for some semantics, it is quite hard to answer for some others. Existence problems are the foundation of reasoning in abstract argumentation, so an analysis of existence problems for incomplete argumentation frameworks is

[1] Institute for Web Science and Technologies, University of Koblenz-Landau, Germany, email: kennethskiba@uni-koblenz.de
[2] Institut für Informatik, Heinrich-Heine-Universität Düsseldorf, Germany, email: {daniel.neugebauer, rothe}@hhu.de

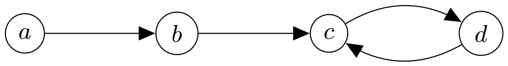

Figure 1. Argumentation framework from Example 1

sorely missing and crucially needed.

Following previous work [5, 4], we look at *possible* and *necessary* variations of our existence problems: Does there exist a completion such that there is a set of arguments satisfying the condition of a given semantics (this is the *possible* problem variant); or, for every possible completion, does there exist a set of arguments satisfying the semantics (this is the *necessary* problem variant)?

This paper is structured as follows. In Section 2, we decribe the formal models of abstract argumentation frameworks and its extension to incomplete argumentation frameworks as well as the existence and nonemptiness problems, followed by a full analysis of the complexity of these two problems for incomplete argumentation frameworks in Section 3. In Section 4, we summarize our results and point out some related work in this field.

2 Model

In this section, we describe the notion of (abstract) argumentation framework formally, which was first presented by Dung [13]. Then we extend this model to *incomplete* argumentation frameworks, as has been done by Baumeister et al. [5].

2.1 Abstract Argumentation Frameworks

An *argumentation framework* AF is a pair $\langle \mathcal{A}, \mathcal{R} \rangle$, where $\mathcal{A}$ is a finite set of arguments and $\mathcal{R}$ is a set of attacks between arguments with $\mathcal{R} \subseteq \mathcal{A} \times \mathcal{A}$. An argument a is said to *attack* b if $(a, b) \in \mathcal{R}$. We call an argument a *acceptable with respect to a set* $S \subseteq \mathcal{A}$ if for each attacker $b \in \mathcal{A}$ of this argument a with $(b, a) \in \mathcal{R}$, there is an argument $c \in S$ which attacks b, i.e., $(c, b) \in \mathcal{R}$; we then say that a is *defended by* c. An argumentation framework $\langle \mathcal{A}, \mathcal{R} \rangle$ can be illustrated by a directed graph with vertex set $\mathcal{A}$ and edge set $\mathcal{R}$.

Example 1. Let $AF = \langle \mathcal{A}, \mathcal{R} \rangle$ with $\mathcal{A} = \{a, b, c, d\}$ and $\mathcal{R} = \{(a, b), (b, c), (c, d), (d, c)\}$ be an argumentation framework. The corresponding graph is shown in Figure 1. Argument b is not acceptable with respect to $\mathcal{A}$, as b is not defended against a's attack. On the other hand, c is acceptable with respect to $\mathcal{A}$, as a defends c against b's attack and c defends itself against d's attack.

In addition to argumentation frameworks, Dung [13] also introduced the notion of *semantics*—properties that a set of arguments should satisfy to be considered acceptable in an argumentation. A set of arguments that satisfies a given semantics is called an *extension*.

The first semantics we consider is *conflict-freeness*. The idea behind it is that a set of acceptable arguments must not contain arguments that contradict (i.e., that attack) each other.

Definition 2. A set $S \subseteq \mathcal{A}$ is *conflict-free* (CF) if there are no arguments a and b in S such that $(a, b) \in \mathcal{R}$.

For every argumentation framework, there is at least one conflict-free extension. In particular, the empty set is always conflict-free.

Definition 3. We call a conflict-free set $S \subseteq \mathcal{A}$ *admissible* (AD) if every argument $a \in S$ is acceptable with respect to S.

Definition 4. A set $S \subseteq \mathcal{A}$ is *complete* (CP) if S is admissible and contains every argument that is acceptable with respect to S.

Definition 5. The *characteristic function* F_{AF} of an argumentation framework $AF = \langle \mathcal{A}, \mathcal{R} \rangle$ is a function $F_{AF} : 2^{\mathcal{A}} \rightarrow 2^{\mathcal{A}}$ defined by

$$F_{AF}(S) = \{a \in \mathcal{A} \mid a \text{ is acceptable with respect to } S\}.$$

A set $S \subseteq \mathcal{A}$ is *grounded* (GR) if S is the least fixed point of the characteristic function of AF.

Definition 6. A set $S \subseteq \mathcal{A}$ is called *preferred* (PR) if it is a maximal admissible set.

Definition 7. We call a conflict-free set $S \subseteq \mathcal{A}$ *stable* (ST) if for each $b \in \mathcal{A} \setminus S$, there is at least one argument $a \in S$ with $(a, b) \in \mathcal{R}$.

Now, let us look at the correlations between the various semantics defined above. The complete, grounded, stable, and preferred semantics also satisfy conflict-freeness and admissibility. Every preferred, grounded, or stable extension is also complete, and every stable extension is preferred.

For every argumentation framework, there always exists a conflict-free, admissible, preferred, complete, and grounded extension. For the stable semantics, existence is not guaranteed. Also, the grounded extension is unique for any argumentation framework, whereas there may be more than one extension for each of the other semantics.

We will study two variants of *existence* decision problems for the above semantics in incomplete argumentation frameworks. For standard argumentation frameworks (in which there is no uncertainty regarding the arguments or attacks), the existence problem was first discussed by Dimopoulos and Torres in [12] and later on by Dunne and Wooldridge [16]. Let $s \in \{\text{CF}, \text{AD}, \text{CP}, \text{GR}, \text{PR}, \text{ST}\}$ be any of the above semantics and define the following problem:

s-EXISTENCE (s-EX)	
Given:	An argumentation framework $AF = \langle \mathcal{A}, \mathcal{R} \rangle$.
Question:	Does there exist a set $S \subseteq \mathcal{A}$ satisfying the conditions specified by semantics s?

The s-NONEMPTINESS (s-NE) problem is a refined version of s-EXISTENCE and asks for the existence of a *nonempty* extension $S \subseteq \mathcal{A}$ for the respective semantics. Some semantics s always have an extension in every argumentation framework, making the s-EXISTENCE problem trivial. The s-NONEMPTINESS problem, on the other hand, is not trivial for any of the semantics considered here.

Both problems coincide for the stable semantics, which never produces empty extensions (provided that the set of arguments $\mathcal{A}$ is nonempty, which we assume).

Observation 8. The problems ST-EX and ST-NE are the same, provided that there is at least one argument.

2.2 Incomplete Argumentation Frameworks

We now turn to the model of *incomplete* argumentation framework that is due to Baumeister et al. [5].

Definition 9. An *incomplete argumentation framework* $\langle \mathcal{A}, \mathcal{A}^?, \mathcal{R}, \mathcal{R}^? \rangle$ partitions both the argument set and the attack set into *definite* ($\mathcal{A}$ and $\mathcal{R}$) and *uncertain* ($\mathcal{A}^?$ and $\mathcal{R}^?$) elements.

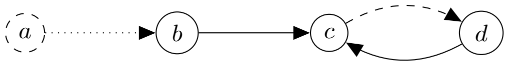

Figure 2. Incomplete argumentation framework from Example 10

An uncertain attack can be between any type of argument, so $\mathcal{R}^? \subseteq (\mathcal{A} \cup \mathcal{A}^?) \times (\mathcal{A} \cup \mathcal{A}^?)$. These uncertain elements can be added to or left out of an argumentation framework. For this operation, we introduce *completions* $AF^* = \langle \mathcal{A}^*, \mathcal{R}^* \rangle$ of an incomplete argumentation framework $\langle \mathcal{A}, \mathcal{A}^?, \mathcal{R}, \mathcal{R}^? \rangle$, which must satisfy $\mathcal{A} \subseteq \mathcal{A}^* \subseteq \mathcal{A} \cup \mathcal{A}^?$ and $\mathcal{R}_{|\mathcal{A}^*} \subseteq \mathcal{R}^* \subseteq (\mathcal{R} \cup \mathcal{R}^?)_{|\mathcal{A}^*}$. We call $\mathcal{R}_{|\mathcal{A}'}$ a *restriction* of an attack relation $\mathcal{R}$ to a set $\mathcal{A}'$ of arguments, with $\mathcal{R}_{|\mathcal{A}'} = \{(a, b) \in \mathcal{R} \mid a, b \in \mathcal{A}'\}$.

Example 10. Let $IAF = \langle \mathcal{A}, \mathcal{A}^?, \mathcal{R}, \mathcal{R}^? \rangle$ be an incomplete argumentation framework with argument sets $\mathcal{A} = \{b, c, d\}$ and $\mathcal{A}^? = \{a\}$ and attack relations $\mathcal{R} = \{(a, b), (b, c), (d, c)\}$ and $\mathcal{R}^? = \{(c, d)\}$. The corresponding graph is shown in Figure 2. We will always draw uncertain arguments and attacks dashed. An interesting observation is the attack (a, b), this attack is drawn dotted, because it is an attack between an uncertain argument a and a certain one, b, while this attack is certain, provided that both arguments occur in the completion. That is, we are forced to add this attack to a completion whenever the argument a is present in it. On the other hand, if a is not in the completion, we must not add this attack to it. Such attacks are called *conditionally definite* by Baumeister et al. [5].

3 Possible and Necessary Existence

In this section, we will formally define the problems of possible existence and nonemptiness necessary existence and nonemptiness in Section 3.1 and study the former in Section 3.2 and the latter in Section 3.3 in terms of their complexity.

3.1 Problem Definitions and Some Prerequisites

The semantics for incomplete argumentation frameworks are defined on completions. We use the idea of *possibly* and *necessarily* satisfied properties. A property is *possibly* satisfied if there is at least one completion of the incomplete argumentation framework in which this property is satisfied. A property is *necessarily* satisfied if it is satisfied in every completion of the incomplete argumentation framework. Following this scheme, we can introduce the "possible" and "necessary" variants of the existence and nonemptiness problems for incomplete argumentation frameworks.

s-POSSIBLE-EXISTENCE (s-POSEX)	
Given:	An incomplete argumentation framework $IAF = \langle \mathcal{A}, \mathcal{A}^?, \mathcal{R}, \mathcal{R}^? \rangle$.
Question:	Does a completion $AF^* = \langle \mathcal{A}^*, \mathcal{R}^* \rangle$ of IAF exist such that there exists an s-extension in AF^*?

The s-POSSIBLE-NONEMPTINESS (s-POSNE) problem instead asks for the existence of a *nonempty* extension $\emptyset \subset S \subseteq \mathcal{A}^*$ for the respective semantics s. For the *necessary* variations of these problems, s-NECESSARY-EXISTENCE (s-NECEX) and s-NECESSARY-NONEMPTINESS (s-NECNE), we ask if for *all* completions the conditions of the s-EXISTENCE problem and the s-NONEMPTINESS problem, respectively, are satisfied.

Our goal is to study these problems in terms of their computational complexity, as has been done by Baumeister et al. [5] for the possible and necessary verification problems and by Baumeister et al. [4] for the possible and necessary credulous and skeptical acceptance problems. The relevant complexity classes all belong to the polynomial hierarchy introduced by Meyer and Stockmeyer [20, 24], whose zeroth level is P, whose first level consists of NP and coNP, and whose second level consists of $\Sigma_2^p = \mathrm{NP}^{\mathrm{NP}}$ and $\Pi_2^p = \mathrm{coNP}^{\mathrm{NP}}$. We assume the reader to be familiar with the concepts of hardness and completeness based on polynomial-time many-one reducibility, and we refer the reader to, e.g., the textbooks by Papadimitriou [22], and Rothe [23] for more background on computational complexity.

We will use a restricted version of the quantified satisfiability (QSAT) problem for our reductions showing Π_2SAT-hardness. Specifically, employing and extending a general construction based on the work of Dimopoulos and Torres [12] and Baumeister et al. [4], we will use Π_2SAT, the canonical problem for the complexity class Π_2^p, which is defined below. Restricting Π_2SAT even further by fixing $X = \emptyset$ in its definition provides the NP-complete problem 3-SAT, which we will use for showing NP-hardness.

Π_2SAT	
Given:	A 3-CNF formula φ over a set $X \cup Y$ of propositional variables.
Question:	For all assignments τ_X on X, is there an assignment τ_Y on Y such that $\varphi[\tau_X, \tau_Y] = \mathtt{true}$?

In this problem definition, X and Y are disjoint sets of propositional variables, φ is a formula in 3-CNF (conjunctive normal form with at most three literals per clause) over these variables, τ_S is a truth assignment on a set of literals S associated with the variables in X and Y (i.e., a mapping $\tau_S : S \to \{\mathtt{true}, \mathtt{false}\}$), and $\varphi[\tau_S]$ is the truth value that φ evaluates to under the assignment τ_S.

Analogously to the construction of Baumeister et al. [4], we translate a QSAT instance to one of two versions of incomplete argumentation frameworks in Definition 11, the first being purely argument-incomplete (i.e., $\mathcal{R}^? = \emptyset$) and the other purely attack-incomplete (i.e., $\mathcal{A}^? = \emptyset$). We use this construction later in our proofs establishing the complexity of these problems.

Definition 11. Given a Π_2SAT instance (φ, X, Y) with $\varphi = \bigwedge_i c_i$ and $c_i = \bigvee_j \alpha_{i,j}$ for each clause c_i, where $\alpha_{i,j}$ are the literals in clause c_i, an argument-incomplete argumentation framework $\langle \mathcal{A}, \mathcal{A}^?, \mathcal{R} \rangle$ representing (φ, X, Y) is defined by:

$$\begin{aligned}
\mathcal{A} = \; & \{\bar{x}_i, \mid x_i \in X\} \cup \{y_i, \bar{y}_i \mid y_i \in Y\} \cup \{c_i \mid c_i \text{ in } \varphi\} \\
& \cup \{\varphi, d, g\}, \\
\mathcal{R} = \; & \{(x_i, \bar{x}_i) \mid x_i \in X\} \cup \{(y_i, \bar{y}_i), (\bar{y}_i, y_i) \mid y_i \in Y\} \\
& \cup \{(x_k, c_i) \mid x_k \text{ in } c_i\} \cup \{(\bar{x}_k, c_i) \mid \neg x_k \text{ in } c_i\} \\
& \cup \{(y_k, c_i) \mid y_k \text{ in } c_i\} \cup \{(\bar{y}_k, c_i) \mid \neg y_k \text{ in } c_i\} \\
& \cup \{(c_i, \varphi) \mid c_i \in \varphi\} \cup \{(\varphi, d)\} \\
& \cup \{(d, a) \mid a \in \{g, x_i, \bar{x}_i, y_j, \bar{y}_j, c_i\}\}, \\
\mathcal{A}^? = \; & \{x_i \mid x_i \in X\}.
\end{aligned}$$

Similarly, given (φ, X, Y) as above, an attack-incomplete argumentation framework $\langle \mathcal{A}, \mathcal{R}, \mathcal{R}^? \rangle$ representing (φ, X, Y) is defined by:

$$\begin{aligned}
\mathcal{A} = &\ \{x_i, \bar{x}_i \mid x_i \in X\} \cup \{y_i, \bar{y}_i \mid y_i \in Y\} \cup \{c_i \mid c_i \text{ in } \varphi\} \\
&\cup \{\varphi, d, g\}, \\
\mathcal{R} = &\ \{(\bar{x}_i, x_i) \mid x_i \in X\} \cup \{(y_i, \bar{y}_i), (\bar{y}_i, y_i) \mid y_i \in Y\} \\
&\cup \{(x_k, c_i) \mid x_k \text{ in } c_i\} \cup \{(\bar{x}_k, c_i) \mid \neg x_k \text{ in } c_i\} \\
&\cup \{(y_k, c_i) \mid y_k \text{ in } c_i\} \cup \{(\bar{y}_k, c_i) \mid \neg y_k \text{ in } c_i\} \\
&\cup \{(c_i, \varphi) \mid c_i \in \varphi\} \cup \{(\varphi, d)\} \\
&\cup \{(d, a) \mid a \in \{g, x_i, \bar{x}_i, y_j, \bar{y}_j, c_i\}\}, \\
\mathcal{R}^? = &\ \{(g, \bar{x}_i) \mid x_i \in X\}.
\end{aligned}$$

Arguments c_i are called *clause arguments* and arguments $z_i \subseteq \alpha_{i,j}$ are called *literal arguments*. For every literal argument z_i, there is a counterargument $\bar{z}_i$. We construct a completion AF^{τ_X} for a given truth assignment τ_X on X from the constructed incomplete argumentation frameworks above. For the argument-incomplete case, we construct $AF^{\tau_X} = \langle \mathcal{A}^{\tau_X}, \mathcal{R}^{\tau_X} \rangle$, with $x_i \in \mathcal{A}^{\tau_X} \Leftrightarrow \tau_X(x_i) = \texttt{true}$. For the attack-incomplete case, we generate $AF^{\tau_X} = \langle \mathcal{A}^{\tau_X}, \mathcal{R}^{\tau_X} \rangle$, with $\mathcal{A}^{\tau_X} = \mathcal{A}$ and $(g, \bar{x}_i) \in \mathcal{R}^{\tau_X} \Leftrightarrow \tau_X(x_i) = \texttt{true}$. For full assignments τ_X and τ_Y, we denote a corresponding set of arguments $\mathcal{A}^{\tau_X}[\tau_X, \tau_Y] = \{x_i \mid \tau_X(x_i) = \texttt{true}\} \cup \{\bar{x}_i \mid \tau_X(x_i) = \texttt{false}\} \cup \{y_j \mid \tau_Y(y_j) = \texttt{true}\} \cup \{\bar{y}_j \mid \tau_Y(y_j) = \texttt{false}\}$ in the completion AF^{τ_X}.

Lemma 12. *Let (φ, X, Y) be a Π_2SAT-instance, let $\langle \mathcal{A}, \mathcal{R}, \mathcal{R}^? \rangle$ or $\langle \mathcal{A}, \mathcal{A}^?, \mathcal{R} \rangle$ be an incomplete argumentation framework created for the instance according to Definition 11. Let τ_X be an assignment on X. In the completion AF^{τ_X}, the argument φ has to be in every nonempty extension that satisfies the semantics $s \in \{\mathrm{AD}, \mathrm{CP}, \mathrm{GR}, \mathrm{PR}, \mathrm{ST}\}$.*

Proof. The argument d attacks every argument except φ, but we cannot use d in any extension because it attacks itself and therefore violates conflict-freeness. So if there is a nonempty s-extension, we need to attack the argument d and φ is the only argument attacking d. ❑

Lemma 13. *Let (φ, X, Y) be a given Π_2SAT-instance and let τ_X and τ_Y be assignments on X and Y, respectively. Let IAF be an incomplete AF created by Definition 11 for (φ, X, Y). Let AF^{τ_X} be its completion corresponding to τ_X and let $\mathcal{A}^{\tau_X}[\tau_X, \tau_Y]$ be the set of literal arguments corresponding to the total assignment. If $\varphi[\tau_X, \tau_Y] = \texttt{true}$, then there exists an extension $\mathcal{A}^{\tau_X}[\tau_X, \tau_Y] \cup \{g, \varphi\}$, which is admissible, complete, preferred, and stable.*

Proof. Assume that $\varphi[\tau_X, \tau_Y] = \texttt{true}$. From Lemma 12 we know that φ is in every nonempty extension of AF^{τ_X}. We show that $\mathcal{E} = \mathcal{A}^{\tau_X}[\tau_X, \tau_Y] \cup \{g, \varphi\}$ is a stable extension in $\mathcal{A}^{\tau_X}$. This will imply that this extension is also admissible, complete, and preferred.

For an extension $\mathcal{E}$ to satisfy the stable semantics, it has to attack every attacker of $\varphi \in \mathcal{E}$ and every argument not in the extension. These attackers are the clause arguments c_i. Every clause argument is attacked by its literal arguments. Since we have a satisfying assignment, for every clause at least one literal is satisfied. So, for every clause argument, there is at least one literal argument attacking it in $\mathcal{A}^{\tau_X}[\tau_X, \tau_Y]$. A literal argument z_i over $X \cup Y$ is part of the extension if it has a corresponding literal that is satisfied. For the Y arguments, it is straightforward that we can use either y_i or $\bar{y}_i$. For these arguments, their attackers are always attacked. Up to this point, this construction works on both variants, the purely argument-incomplete and the purely attack-incomplete variant. The only distinction is the X part of the literals.

Consider the argument-incomplete variant first. An argument x_i is part of a completion if the corresponding literal is satisfied. This argument x_i is then only attacked by d and also attacks its counterargument $\bar{x}_i$. If the literal is not satisfied, x_i is then not part of the completion and $\bar{x}_i$ is only attacked by d. Therefore, we can use the arguments corresponding to the assignment of the literals in the extension. In this variant, we have to use the argument g in the extension because it is only attacked by d, which cannot be used in any conflict-free extension.

In the attack-incomplete variant, argument x_i is attacked by its counterargument $\bar{x}_i$. But if the literal x_i is satisfied, then this counterargument is attacked by g and g is only attacked by d. So argument g cannot be attacked by any extension and we have to use this argument in any stable extension. If x_i is not satisfied, then this attack between g and $\bar{x}_i$ is not in the completion and we can then use $\bar{x}_i$ in the extension. Therefore, every argument is either attacked by arguments in $\mathcal{E}$ or is a member of $\mathcal{E}$. So in both variants, the extension contains φ, τ_X, τ_Y, and g, and this extension is stable. ❑

With Lemmas 12 and 13, we see that the two constructions of Definition 11 behave similarly and can therefore be interchanged. We will only use the argument-incomplete case, yielding the hardness of the problems for the general case with uncertainty for arguments *and* attacks.

We start our complexity analysis by stating all cases where existence of an extension is trivial.

Proposition 14. *For $s \in \{\mathrm{CF}, \mathrm{AD}, \mathrm{CP}, \mathrm{PR}, \mathrm{GR}\}$, s-PosEx and s-NecEx are trivial.*

Proof. The empty set always exists and this set fulfills all the properties needed for each of the given semantics s. Hence, the answer to this problem will always be "yes." ❑

Next, we will provide hardness results for the remaining (nontrivial) existence and nonemptiness problems.

3.2 Possible Existence and Nonemptiness

Before presenting our hardness results for the possible existence and nonemptiness problems, we start with an easy polynomial-time algorithm for conflict-freeness.[3]

Proposition 15. CF-NecNe *and* CF-PosNe *are in* P.

Proof. A set with only one argument and no self-attack is always conflict-free and also nonempty. Thus, in an AF, we only need to find one argument without a self-attack to get a nonempty conflict-free extension in that AF. Conversely, if all arguments in an argumentation framework have a self-attack, then no nonempty set of arguments can be conflict-free.

Using these considerations, CF-PosNe can be solved by checking whether all arguments—both certain and uncertain—in the given incomplete AF have (conditionally) definite self-attacks. If yes, the answer to CF-PosNe is "no," and vice versa. Similarly, $IAF \in$ CF-NecNe if and only if IAF has a definite argument with no definite or uncertain self-attack. ❑

[3] Note that Proposition 15 contains also a result on a necessary-nonemptiness problem, CF-NecNe. Those problems are actually considered only in Section 3.3. However, since this proof is closely related to the proof for CF-PosNe, we provide both results together in the present subsection.

Proposition 16. ST-PosEx *is* NP-*complete.*

Proof. To prove NP-membership, we look at the quantifier representation. ST-PosEx can be written as

$$\exists \text{ completion } AF^* \text{ of } IAF : \exists \mathcal{E} \subseteq \mathcal{A}^* : \mathcal{E} \text{ is stable in } AF^*?$$

These two polynomial bounded existential quantifiers can be merged into one existential quantifier, with an inner predicate in P, so the problem is in NP.

Hardness of this problem can be deduced from the NP-hardness of the ST-Ex problem, which was shown by Chvátal [9]. ❑

From Observation 8 we know that the existence problem is the same problem as the nonemptiness problem for the stable semantics. From Proposition 16 it thus follows that ST-PosNe is NP-complete.

Corollary 17. ST-PosNe *is* NP-*complete.*

Proposition 18. *For* $s \in \{\text{AD}, \text{CP}, \text{PR}\}$*,* s-PosNe *is* NP-*complete.*

Proof. Again, we have two existential quantifiers in our problem definition, which can be merged into one to provide an NP upper bound for each of the problems considered.

The NP lower bounds are inherited from the s-Ne problems for the same semantics s, which were shown to be NP-complete by Dimopoulos and Torres [12]. ❑

On the other hand, for the grounded semantics, possible nonemptiness is easy to solve, though not in a trivial way.

Theorem 19. GR-PosNe *is in* P.

Proof. It is apparent that an AF has a nonempty grounded extension if and only if it has at least one unattacked argument—if the AF has an unattacked argument, then this argument is a member of the grounded extension, and if it doesn't, then no argument can be acceptable with respect to the empty set, so the empty set is the grounded extension. Therefore, the GR-PosNe problem is equivalent to asking whether an incomplete AF has a completion which has an unattacked argument.

First, given an incomplete argumentation framework $IAF = \langle \mathcal{A}, \mathcal{A}^?, \mathcal{R}, \mathcal{R}^? \rangle$, we search in the minimal completion $AF^* = \langle \mathcal{A}, \mathcal{R}|_{\mathcal{A}} \rangle$ that only contains the definite parts of the incomplete argumentation framework. If AF^* has an unattacked argument, the answer for the GR-PosNe instance is clearly "yes." Otherwise, for every uncertain argument $x \in \mathcal{A}^?$, let $AF^*_x = \langle \mathcal{A} \cup \{x\}, \mathcal{R}|_{\mathcal{A} \cup \{x\}} \rangle$ be the completion that contains only x and all definite elements. If any of these completions AF^*_x has an unattacked argument, we can again answer "yes." If none of them have an unattacked argument, it is clear that no completion can have one, since adding several uncertain arguments simultaneously or adding additional attacks cannot produce unattacked arguments that were not unattacked before. Therefore, we can answer "no" in this case.

This algorithm requires to check at most $|\mathcal{A}^?| + 1$ completions for unattacked arguments, which can be done in polynomial time. ❑

3.3 Necessary Existence and Nonemptiness

We now turn to the necessary existence and nonemptiness problems (besides CF-NecNe, which has already been handled in Proposition 15, and the trivial cases of necessary existence problems handled in Proposition 14) and show that most of them are hard for the second level of the polynomial hierarchy. We start with the upper bounds for the nonemptiness problems.

Proposition 20. *For* $s \in \{\text{AD}, \text{CP}, \text{PR}, \text{ST}\}$*,* s-NecNe *is in* Π_2^p*.*

Proof. To show membership, we can look at the quantifier representation. We can formulate the problems as

$$\forall \text{ completion } AF^* \text{ of } IAF : \exists \mathcal{E} \subseteq \mathcal{A}^* : \mathcal{E} \text{ is } s \text{ in } AF^*?$$

For this representation, we use polynomially length-bounded quantifiers. The existentially quantified inner part is the problem s-Nonemptiness—this problem is NP-complete, as shown by Dimopoulos and Torres [12]. Hence, we have an existential quantifier followed by a statement checkable in polynomial time. This expression is preceded by a universal quantifier over completions. Both quantifiers yield a problem in Π_2^p. ❑

Theorem 21. ST-NecEx *is* Π_2^p*-complete.*

Proof. The Π_2^p upper bound of this problem follows immediately from Observation 8 and Proposition 20.

To show Π_2^p-hardness of this problem, we reduce from the problem Π_2SAT. Given a Π_2SAT instance (φ, X, Y), we create an argument-incomplete argumentation framework $IAF = \langle \mathcal{A}, \mathcal{A}^?, \mathcal{R} \rangle$ according to Definition 11.

Assume that for every assignment τ_X on X there exists an assignment τ_Y on Y such that $\varphi[\tau_X, \tau_Y]$ is $\texttt{true}$. Then we have a "yes"-instance of Π_2SAT and we need to show that there is always a stable extension $\mathcal{E}$. In Lemma 13, we have shown that if $\varphi[\tau_X, \tau_Y] = \texttt{true}$ then there is an extension, namely $\mathcal{E} = \mathcal{A}^{\tau_X}[\tau_X, \tau_Y] \cup \{g, \varphi\}$, that satisfies the stable semantics.

For the other direction of the proof, assume there exists a stable extension $\mathcal{E}$. The argument d cannot be in $\mathcal{E}$, because it attacks itself. Therefore, this extension contains φ, as it is the only argument that attacks d and for an extension to be stable, for each argument that it does not contain, it must contain an argument attacking the outside argument. The arguments c_i cannot be in $\mathcal{E}$, since they all attack φ and because for an extension to be stable, it must be conflict-free.

The only arguments left are the literal arguments $z_i \in X \cup Y$ and the argument g. It is obvious that we need either z_i or $\bar{z}_i$ for a stable extension and cannot use both. Since we have a stable extension, every c_i must be attacked by at least one z_i in $\mathcal{E}$. Now, there could still be some literals z_j for which neither argument z_j nor argument $\bar{z}_j$ are in $\mathcal{E}$. For each of these, either z_j or $\bar{z}_j$ (but not both) must be in $\mathcal{E}$. Finally, g is in $\mathcal{E}$, since φ defends it against its only attacker d.

Consequently, we can define a total assignment based on the stable extension $\mathcal{E}$ depending on whether the literal arguments are in or out of $\mathcal{E}$: If an argument z_i is in $\mathcal{E}$, we set the corresponding literal $\alpha_{i,j}$ to $\texttt{true}$; and if it is not, we set $\neg\alpha_{i,j}$ to $\texttt{true}$. We know that this assignment must be a satisfying assignment, since every clause argument is attacked by $\mathcal{E}$ and, accordingly, every clause is satisfied by the assignment. ❑

Example 22. Consider the Π_2SAT instance $\varphi = (x_1 \vee \neg y_1) \wedge (y_1 \vee \neg y_2)$, with $X = \{x_1\}$ and $Y = \{y_1, y_2\}$. We get $\mathcal{A} = \{\bar{x}_1, y_1, \bar{y}_1, y_2, \bar{y}_2, c_1, c_2, \varphi, d, g\}$ and $\mathcal{A}^? = \{x_1\}$ by using our construction. This instance is a "yes"-instance of Π_2SAT, since for every assignment τ_X on X, there is an assignment τ_Y on Y such that $\varphi[\tau_X, \tau_Y] = \texttt{true}$—in particular, the assignment $\tau_Y(y_1) = \texttt{false}$ and $\tau_Y(y_2) = \texttt{false}$ satisfies φ irrespective of the assignment on X used.

Figure 3 shows the corresponding graph. This ST-NecEx instance is a "yes" instance, too: For τ_X^1 with $\tau_X^1(x_1) = \texttt{true}$, the

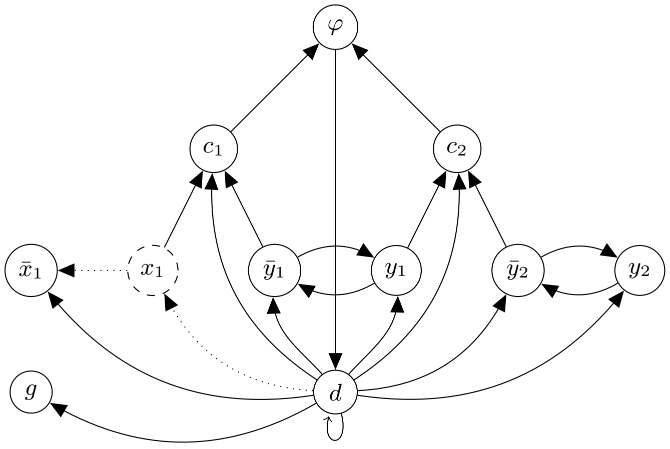

Figure 3. Graph for Example 22

set $\{\varphi, x_1, \bar{y}_1, \bar{y}_2, g\}$ is stable in the completion $AF^{\tau_X^1}$, and for τ_X^2 with $\tau_X^1(x_1) = \texttt{false}$, the set $\{\varphi, \bar{x}_1, \bar{y}_1, \bar{y}_2, g\}$ is stable in the completion $AF^{\tau_X^2}$, so there exists a stable extension in every completion.

From Observation 8 we know that the existence problem is the same as the nonemptiness problem for the stable semantics. With Theorem 21 we thus obtain that ST-NECNE is Π_2^p-complete.

Corollary 23. ST-NECNE *is* Π_2^p*-complete.*

Theorem 24. AD-NECNE*,* CP-NECNE*, and* PR-NECNE *each are* Π_2^p*-complete.*

Proof. The Π_2^p upper bounds have been shown in Proposition 20 already.

To prove the Π_2^p lower bounds, we can use the same construction as in the proof of Theorem 21. Specifically, we construct an argument-incomplete $IAF = \langle \mathcal{A}, \mathcal{A}^?, \mathcal{R} \rangle$ from a Π_2SAT instance (φ, X, Y) according to Definition 11. We know from Lemma 12 that φ has to be in every nonempty extension, which satisfies the admissible semantics. We also know that $\mathcal{E} = \mathcal{A}^{\tau_X}[\tau_X, \tau_Y] \cup \{g, \varphi\}$ is an admissible extension if $\varphi[\tau_X, \tau_Y] = \texttt{true}$. For the other direction of this proof, we can use the same idea as in Theorem 21.

With the admissible extension we can also prove the CP and PR cases, because here, the admissible extension contains every argument that is acceptable with regard to $\mathcal{E}$. So this extension is complete. Also, every argument is either in $\mathcal{E}$ and acceptable, or not in $\mathcal{E}$ and not acceptable. Consequently, we cannot add any arguments to our extension $\mathcal{E}$ and, therefore, this extension is maximal admissible, that is, it is preferred. If a nonadmissible completion exists for the incomplete argumentation framework, then at least one c_i is not attacked. Hence, φ cannot be acceptable by this extension and we cannot find any nonempty preferred or complete extension. ❑

Finally, we turn to necessary nonemptiness for the grounded semantics. While the necessary existence problem, GR-NECEX, was shown to be efficiently solvable on purely trivial grounds in Proposition 14, GR-NECNE is also efficiently solvable but by an algorithm that is far from trivial.

Theorem 25. GR-NECNE *is in* P.

Proof. The idea behind this proof is to prevent the existence of a grounded extension. Assume we have an incomplete argumentation framework. We start with a completion that contains no uncertain elements and then add arguments or attacks, which can possibly destroy a grounded set.

To this end, we first look at two extreme cases separately, concerning attack-incompleteness and argument-incompleteness, and then combine them so as to capture the general case.

Case 1 (attack-incompleteness): For purely attack-incomplete argumentation frameworks, we can just add all possible attacks to the argumentation framework. In this maximal completion, the set of unattacked arguments is minimal, because, by adding attacks to a completion, the number of unattacked arguments can decrease or maybe stay the same, but it cannot grow.

If there exists a nonempty grounded extension in the maximal completion, then there must be a nonempty grounded extension in every completion, since every argument that is unattacked in the maximal completion is also unattacked in any other completion. Constructing the maximal completion and testing if there exists a nonempty grounded extension in the maximal completion is possible in (deterministic) polynomial time.

Case 2 (argument-incompleteness): For purely argument-incomplete argumentation frameworks, we add all arguments to our completion that are attacked by at least one other argument contained in this completion. We begin with the argumentation framework $AF_0 = \langle \mathcal{A}_0, \mathcal{R} \rangle$, where we only use the arguments that are certain. That is, $\mathcal{A}_0 = \mathcal{A}$ for any incomplete argumentation framework $AF^* = \langle \mathcal{A}, \mathcal{A}^?, \mathcal{R}, \mathcal{R}^? \rangle$.

Step by step, we now add uncertain arguments that are attacked by arguments from the previous step, i.e., $AF_{i+1} = \langle \mathcal{A}_{i+1}, \mathcal{R} \rangle$ with

$$\mathcal{A}_{i+1} = \mathcal{A}_i \cup \{a \mid (b, a) \in \mathcal{R} \wedge b \in \mathcal{A}_i\}.$$

With the addition of uncertain arguments, we may also add some conditionally definite attacks to the completion. So we add no unattacked argument and possibly add attacks to some previously unattacked arguments. Consequently, we minimize the set of unattacked arguments.

When looking at the algorithm from Modgil and Caminada [21] for generating grounded sets, we can see that the starting points are the unattacked arguments. In the modified completion we do not add new starting points. By adding more arguments to a completion, some previously unattacked arguments can be attacked by these new arguments, because these new arguments may have conditionally definite attacks against other arguments.

If there exists a nonempty grounded extension in the thus modified argumentation framework, then there must be a nonempty grounded extension in every other completion, and the process of checking if there is a grounded extension in a complete argumentation framework can again be done in (deterministic) polynomial time.

Case 3 (general case): For the general case, we can just combine both solutions. We first add all uncertain attacks to the completion according to Case 1, and then add arguments according to Case 2 to the completion. Therefore, we minimize the set of unattacked arguments and if there exists a grounded extension in this case, there exists a grounded extension in every other completion as well. ❑

Example 26. Now, let us have a look at an example for the problem GR-NECNE. Let $IAF = \langle \mathcal{A}, \mathcal{A}^?, \mathcal{R}, \mathcal{R}^? \rangle$ be an incomplete argumentation framework with $\mathcal{A} = \{a, b, c, d, g\}$, $\mathcal{R} =$

Table 1. Summary of complexity results. NP-c stands for "NP-complete" and Π_2^p-c for "Π_2^p-complete." Results due to this paper are marked by the corresponding theorem, proposition, or corollary, and results due to previous work are marked by the respective reference.

	Ex	**Ne**	**PosEx**		**PosNe**		**NecEx**		**NecNe**	
CF	trivial	trivial	trivial	(Prop. 14)	in P	(Prop. 15)	trivial	(Prop. 14)	in P	(Prop. 15)
GR	trivial	in P [21]	trivial	(Prop. 14)	in P	(Thm. 19)	trivial	(Prop. 14)	in P	(Thm. 25)
AD	trivial	NP-c [12]	trivial	(Prop. 14)	NP-c	(Prop. 18)	trivial	(Prop. 14)	Π_2^p-c	(Thm. 24)
CP	trivial	NP-c [12]	trivial	(Prop. 14)	NP-c	(Prop. 18)	trivial	(Prop. 14)	Π_2^p-c	(Thm. 24)
PR	trivial	NP-c [12]	trivial	(Prop. 14)	NP-c	(Prop. 18)	trivial	(Prop. 14)	Π_2^p-c	(Thm. 24)
ST	NP-c [9]	NP-c [9]	NP-c	(Prop. 16)	NP-c	(Cor. 17)	Π_2^p-c	(Thm. 21)	Π_2^p-c	(Cor. 23)

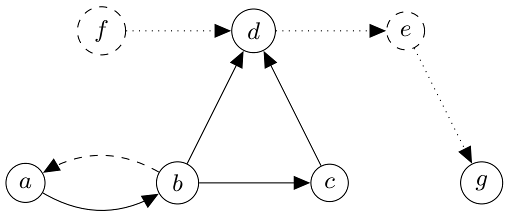

Figure 4. Graph for Example 26

$\{(a,b),(b,c),(c,d),(b,d),(d,e),(e,g),(f,d)\}$, $\mathcal{A}^? = \{e,f\}$, and $\mathcal{R}^? = \{(b,a)\}$. The corresponding graph is depicted in Figure 4. The grounded extension for the minimal completion that drops all uncertain elements is $\{a,c,g\}$. The arguments a and g have no attackers, so they must be in the grounded extension, and a also defends c against b. Looking first at the uncertain attacks (Case 1), we add the attack (b,a). Since argument a now is attacked, it must leave and so can no longer defend c; therefore, the grounded extension only contains g now. For the uncertain arguments (Case 2), we first look at the argument f. We do not add this argument to the completion because it has no attackers. Otherwise, the set of unattacked arguments would get bigger, increasing the chance for a nonempty grounded exension. But we would add e to the completion, because d attacks e, so e is already attacked in the completion. When we combine both cases (Case 3), we first add the attack between (b,a) and then the argument e to the completion. Here, the grounded extension is empty at the end, showcasing that the algorithm successfully identified a completion with no nonempty grounded extension and correctly provides a "no" answer to this GR-NecNe instance.

4 Conclusion

The contribution of this work is to solve the open questions of the existence and nonemptiness problems for incomplete argumentation frameworks. We pinpointed the complexity of two variations of both problem, the possible and the necessary variant. A summary of all results can be found in Table 1. When we compare the complexity of the incomplete problems with their corresponding complete versions, we can see that the complexity for the possible cases stays the same (due to collapsing existential quantifiers), whereas the complexity of the necessary cases grows to Π_2^p for several semantics. The s-PosNe and s-NecNe problems for $s \in \{\mathrm{CF}, \mathrm{GR}\}$ have the same complexity as the respective base problem for standard argumentation frameworks, while not relying on collapsing quantifiers.

Beside the original semantics we covered in this work, there are more semantics like the stage semantics [25], the CF2 semantics [1], the semi-stable semantics [7], or the ideal semantics [14]. The complexity of IAF generalizations of existence problems for these semantics is an interesting task for future work.

Other models that also represent uncertainty in AFs include Control AFs (CAFs) by Dimopoulos et al. [11] which also incorporate uncertainty in the form of a *control part* and an *uncertain part*, similar to uncertain arguments in IAFs. The results obtained in our paper could potentially be adapted for CAFs. Cayrol et. al. [8] introduce attack-incomplete AFs as *Partial AFs* (PAFs). Instead of reducing the semantics of PAFs to the semantics of AFs via completions (like IAFs do), they define new semantics for Partial AFs, which coincide with the completion-based approach for conflict-freeness, but generally not for the more advanced semantics. Maher [19] presents a model of *strategic argumentation*, which simulates a game between a proponent P and an opponent O, where P tries to make an argument a accepted, while O tries the opposite. In this model, there are three sets of arguments, $\mathcal{A}_{Com}$ (common knowledge), $\mathcal{A}_P$ (arguments of P), and $\mathcal{A}_O$ (arguments of O), where $\mathcal{A}_P$ and $\mathcal{A}_O$ are comparable to uncertain arguments in IAFs. However, this model allows no uncertain attacks between arguments; every attack is definite. Further, IAFs have some similarities to AF expansion due to Baumann and Brewka [2], where new arguments and new attacks incident to at least one new argument may be added. Opposed to IAFs, the set of new arguments and attacks is not fixed, and new attacks among existing arguments are not allowed.

Existence and nonemptiness problems are not the only interesting problems for incomplete argumentation frameworks. As mentioned earlier, Baumeister et al. [5] studied *verification* problems. A second question that was already studied for incomplete argumentation frameworks are *acceptance* problems. There are two important variants of them: One is *credulous acceptance*, where the question is whether a given argument belongs to *some* extension for a certain semantics; the other is *skeptical acceptance*, where the question is whether an argument is in *every* extension satisfying a certain semantics. For this problem, Baumeister et al. [4] showed that the complexity grows in the incomplete case compared to the complete case.

Another variation of our model could be to add weights to the uncertain elements. So far, we tried to minimize extensions only with respect to the number of arguments in them. Also, in the model used here, arguments and attacks are treated essentially equally, which may not always be intended, especially not with uncertainty in the model. Previous work from Dunne et al. [15], who looked at weighted attacks, could help to better refine our model. Similarly, Li et al. [18] followed a probabilistic approach, using probabilities both for arguments and attacks, and later on, Fazzinga et al. [17] studied the complexity of problems defined in this probabilistic model.

Acknowledgements: The research reported here was supported by the Deutsche Forschungsgemeinschaft under grants KE 1413/11-1 and RO 1202/14-2.

REFERENCES

[1] P. Baroni, M. Giacomin, and G. Guida, 'SCC-recursiveness: A general schema for argumentation semantics', *Artificial Intelligence*, **168**(1-2), 162–210, (2005).

[2] R. Baumann and G. Brewka, 'Expanding argumentation frameworks: Enforcing and monotonicity results', in *Proceedings of the 3rd International Conference on Computational Models of Argument*, pp. 75–86. IOS Press, (September 2010).

[3] D. Baumeister, D. Neugebauer, and J. Rothe, 'Verification in attack-incomplete argumentation frameworks', in *Proceedings of the 4th International Conference on Algorithmic Decision Theory*, pp. 341–358. Springer-Verlag *Lecture Notes in Artificial Intelligence #9346*, (September 2015).

[4] D. Baumeister, D. Neugebauer, and J. Rothe, 'Credulous and skeptical acceptance in incomplete argumentation frameworks', in *Proceedings of the 7th International Conference on Computational Models of Argument*, pp. 181–192. IOS Press, (2018).

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

[6] D. Baumeister, J. Rothe, and H. Schadrack, 'Verification in argument-incomplete argumentation frameworks', in *Proceedings of the 4th International Conference on Algorithmic Decision Theory*, pp. 359–376. Springer-Verlag *Lecture Notes in Artificial Intelligence #9346*, (September 2015).

[7] M. Caminada, 'Semi-stable semantics', in *Proceedings of the 1st International Conference on Computational Models of Argument*, pp. 121–130. IOS Press, (September 2006).

[8] C. Cayrol, C. Devred, and M. Lagasquie-Schiex, 'Handling ignorance in argumentation: Semantics of partial argumentation frameworks', in *Proceedings of the 9th European Conference on Symbolic and Quantitative Approaches to Reasoning and Uncertainty*, pp. 259–270. Springer-Verlag *Lecture Notes in Artificial Intelligence #4724*, (November 2007).

[9] V. Chvátal, 'On the computational complexity of finding a kernel', *Report CRM-300, Centre de Recherches Mathématiques, Université de Montréal*, (1973).

[10] S. Coste-Marquis, C. Devred, S. Konieczny, M. Lagasquie-Schiex, and P. Marquis, 'On the merging of Dung's argumentation systems', *Artificial Intelligence*, **171**(10), 730–753, (2007).

[11] Y. Dimopoulos, J. Mailly, and P. Moraitis, 'Control argumentation frameworks', in *Proceedings of the 32nd AAAI Conference on Artificial Intelligence*, pp. 4678–4685. AAAI Press, (February 2018).

[12] Y. Dimopoulos and A. Torres, 'Graph theoretical structures in logic programs and default theories', *Theoretical Computer Science*, **170**(1), 209–244, (1996).

[13] P. Dung, 'On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games', *Artificial Intelligence*, **77**(2), 321–357, (1995).

[14] P. Dung, P. Mancarella, and F. Toni, 'A dialectic procedure for sceptical, assumption-based argumentation', in *Proceedings of the 1st International Conference on Computational Models of Argument*, pp. 145–156. IOS Press, (September 2006).

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

[16] P. Dunne and M. Wooldridge, 'Complexity of abstract argumentation', in *Argumentation in Artificial Intelligence*, eds., I. Rahwan and G. Simari, chapter 5, 85–104, Springer, (2009).

[17] B. Fazzinga, S. Flesca, and F. Parisi, 'On the complexity of probabilistic abstract argumentation', in *Proceedings of the 23rd International Joint Conference on Artificial Intelligence*, pp. 898–904. AAAI Press/IJCAI, (August 2013).

[18] H. Li, N. Oren, and T. Norman, 'Probabilistic argumentation frameworks', in *Proceedings of the 1st International Workshop on Theory and Applications of Formal Argumentation*, pp. 1–16. Springer-Verlag *Lecture Notes in Artificial Intelligence #7132*, (July 2011).

[19] M. Maher, 'Resistance to corruption of strategic argumentation', in *Proceedings of the 30th AAAI Conference on Artificial Intelligence*, pp. 1030–1036. AAAI Press, (February 2016).

[20] A. Meyer and L. Stockmeyer, 'The equivalence problem for regular expressions with squaring requires exponential space', in *Proceedings of the 13th IEEE Symposium on Switching and Automata Theory*, pp. 125–129. IEEE Computer Society Press, (1972).

[21] S. Modgil and Caminada M., 'Proof theories and algorithms for abstract argumentation frameworks', in *Argumentation in Artificial Intelligence*, 105–129, Springer, (2009).

[22] C. Papadimitriou, *Computational Complexity*, Addison-Wesley, second edn., 1995.

[23] J. Rothe, *Complexity Theory and Cryptology. An Introduction to Cryptocomplexity*, EATCS Texts in Theoretical Computer Science, Springer-Verlag, 2005.

[24] L. Stockmeyer, 'The polynomial-time hierarchy', *Theoretical Computer Science*, **3**(1), 1–22, (1976).

[25] B. Verheij, 'Two approaches to dialectical argumentation: Admissible sets and argumentation stages', in *Proceedings of the 8th Dutch Conference on Artificial Intelligence*, pp. 357–368, (1996).