Credulous and Skeptical Acceptance in Incomplete Argumentation Frameworks

Dorothea BAUMEISTER, Daniel NEUGEBAUER and Jörg ROTHE
Institut für Informatik, Heinrich-Heine-Universität Düsseldorf, Düsseldorf, Germany

Abstract. We propose natural generalizations of the credulous and skeptical acceptance problems in abstract argumentation for incomplete argumentation frameworks [3]. This continues earlier work on a similar generalization of the verification problem. We provide a full analysis of the computational complexity of the generalized problems for all original semantics, showing that, in almost all cases, acceptance problems for incomplete argumentation frameworks are significantly harder than the respective problems for argumentation frameworks without uncertainty. All our hardness results for the classes NP, coNP, \( \Pi_2^p \) and \( \Sigma_2^p \) are derived from one generic reduction.

Keywords. Incomplete argumentation framework, credulous acceptance, skeptical acceptance, computational complexity

1. Introduction

Abstract argumentation frameworks [11] are a formal model that represents an argumentation by a set of atomic arguments and an attack relation between arguments. Unquantified uncertainty about the existence of particular attacks or arguments in abstract argumentation frameworks was first introduced by Coste-Marquis et al. [6] for the set of attacks and by Baumeister et al. [4] for the set of arguments. Baumeister et al. [3] subsequently generalized both models to incomplete argumentation frameworks, which allow uncertainty about both arguments and attacks. An incomplete argumentation framework can be seen as a representation of a set of possible worlds, called completions, each of which is a standard argumentation framework that shares all definite elements of the incomplete framework and where each of its uncertain elements is either included or excluded. Existing problems for argumentation frameworks can then be generalized to incomplete argumentation frameworks by either asking whether they are satisfied possibly (in at least one completion) or necessarily (in all completions), i.e., whether the uncertainty either can or must be resolved in a way that satisfies the conditions of the given problem. In applications, that answer may help with decisions in strategic scenarios, where the uncertainty represents possible moves. In scenarios where uncertainty represents missing information, the preliminary answer may be sufficient for the task at hand, removing the need to actually resolve the uncertainty.

In this paper, we continue that line of research and turn to the well-understood problems of credulous and skeptical acceptance, which are parameterized by a semantics and, for a given argumentation framework and an argument in that framework, either
ask whether that argument is in *at least one* extension (for credulous acceptance) or *all* extensions (for skeptical acceptance) of the framework with respect to the semantics. For incomplete argumentation frameworks, we study the following four problem combinations, each of which covers interesting questions that arise in different application scenarios.

- **Possible Credulous Acceptance:** Is there any way to accept the given argument?
- **Necessary Credulous Acceptance:** Is the given argument in at least one extension, regardless of how the uncertainty is resolved?
- **Possible Skeptical Acceptance:** Can the uncertainty be resolved in such a way that the given argument is in *all* extensions?
- **Necessary Skeptical Acceptance:** Is the given argument absolutely guaranteed to be accepted?

We continue with formal definitions of the required notions in Section 2, followed by a full analysis of the complexity of possible and necessary acceptance problems in incomplete argumentation frameworks in Section 3, and a conclusion in Section 4.

2. Model

We describe the standard model of (abstract) argumentation framework in Section 2.1, including the skeptical and credulous acceptance problems, and we introduce the more general model of incomplete argumentation framework in Section 2.2.

2.1. Argumentation Frameworks

An argumentation framework $AF = (\mathcal{A}, \mathcal{R})$ consists of a finite set $\mathcal{A}$ of arguments and a binary attack relation $\mathcal{R} \subseteq \mathcal{A} \times \mathcal{A}$ on the arguments, where $(a, b) \in \mathcal{R}$ indicates that $a$ attacks $b$. An argument $a \in \mathcal{A}$ is defended by a set $A \subseteq \mathcal{A}$ of arguments in $AF$ if, for each attacker $b \in \mathcal{A}$ of $a$ with $(b, a) \in \mathcal{R}$, there is a defender $d \in A$ with $(d, b) \in \mathcal{R}$. The characteristic function of $AF$, $F_{AF} : 2^\mathcal{A} \rightarrow 2^\mathcal{A}$, outputs all arguments defended by a given set, i.e., $F_{AF}(A) = \{ a \in \mathcal{A} \mid a \text{ is defended by } A \text{ in } AF \}$.

$F_{AF}^k$ denotes the $k$-fold composition of $F_{AF}$, and $F_{AF}^\ast$ denotes its infinite composition. A set $A \subseteq \mathcal{A}$ is conflict-free (CF) if $(a, b) \notin \mathcal{R}$ for all $a, b \in A$. A conflict-free set $A \subseteq \mathcal{A}$ is further admissible (AD) if $A \subseteq F_{AF}(A)$, complete (CP) if $A = F_{AF}(A)$, grounded (GR) if $A = F_{AF}^\ast(\emptyset)$, preferred (PR) if $A$ is admissible and has no admissible superset, and stable (ST) if for every $b \in \mathcal{A} \setminus A$ there is an $a \in A$ with $(a, b) \in \mathcal{R}$. A set of arguments that satisfies one of these semantics is called an extension of the argumentation framework with respect to that semantics. Every stable extension is preferred, every preferred extension is complete, every complete extension is admissible, and every admissible set is conflict-free. Further, the unique grounded extension is complete. There are argumentation frameworks that have no stable extension, all other extensions are guaranteed to exist.

We study the credulous acceptance and skeptical acceptance problems in argumentation frameworks, which were first defined by Dunne and Bench-Capon [12] for the preferred semantics alone, but that have since been adopted for various other semantics. The problems are defined as follows, where $s \in \{\text{CF, AD, CP, GR, PR, ST}\}$ is a placeholder for any of the above semantics.
Given: An argumentation framework \(\langle A, R \rangle\) and an argument \(a \in A\).

Question: Is there an s extension \(E\) of \(\langle A, R \rangle\) with \(a \in E\)?

The stable semantics is a special case, since there may be no stable extension in an argumentation framework. We use the standard formalization of ST-SA which has a “yes” answer for all instances where there is no stable extension, following the convention that a universal quantifier over an empty space defaults to true. This problem was shown to be coNP-complete by Dimopoulos and Torres [10]. However, this means that it is possible for an argument to be skeptically accepted but at the same time not credulously accepted, which may be undesired. An alternative formalization incorporates an exception for instances without stable extensions, treating them as “no”-instances. In this case, the skeptical acceptance problem is even DP-complete (see [13]), where DP (the second level of the boolean hierarchy over NP) is the class of differences of any two NP sets and contains both NP and coNP. We leave the analysis of this variant in the context of incomplete argumentation frameworks to future work.

We make a few observations about the CA and SA problems that will be used later.

Observation 1. Skeptical acceptance for CF and AD is trivial: The empty set is always conflict-free and admissible, so the answer is always “no” for all problem instances.

Observation 2. Since the grounded extension is unique, there is no difference between skeptical and credulous acceptance for the grounded semantics, i.e., GR-CA = GR-SA.

Observation 3. Since the grounded extension is exactly the intersection of all complete extensions, skeptical acceptance of an argument is the same for the grounded and the complete semantics: GR-SA = CP-SA.

Note that Observations 2 and 3 together yield that GR-CA = GR-SA = CP-SA.

Observation 4. The credulous acceptance problem is the same for the admissible and the preferred semantics (AD-CA = PR-CA): An argument is a member of (at least) one preferred extension if and only if it is in (at least) one admissible extension, since every preferred set is admissible and every admissible set is a subset of some preferred set.

The computational complexity of CA and SA for the six semantics considered in this paper was studied by Dimopoulos and Torres [10], Coste-Marquis et al. [7], and Dunne and Bench-Capon [12]. We present their results together with our new findings in Table 2 in Section 4.

2.2. Incomplete Argumentation Frameworks

An incomplete argumentation framework \(\langle A, A^\?, R, R^? \rangle\) splits both the set of arguments and the set of attacks into two disjoint parts, a definite part \((A, R)\) and an uncertain part \((A^?, R^?)\), where both attack types are subsets of \((A \cup A^?) \times (A \cup A^?)\). For uncertain elements (members of \(A^?\) or \(R^?\)), it is not known whether they are part...
of the argumentation—they might be added or removed in the future, or the uncertainty may just represent the limited knowledge of some agent about those elements. Definite arguments (elements of \( \mathcal{A} \)) are known to exist, while definite attacks (elements of \( \mathcal{R} \)) exist if and only if both incident arguments exist, too. To account for this, we call attacks in \( \mathcal{R} \) that are incident to at least one uncertain argument conditionally definite, since these attacks may vanish alongside an incident uncertain argument, while attacks in \( \mathcal{R} \) that are only incident to definite arguments are called definite. If \( \mathcal{A} = \emptyset \), we have a purely attack-incomplete argumentation framework; for \( \mathcal{R} = \emptyset \), a purely argument-incomplete argumentation framework; and \( \mathcal{A} = \mathcal{R} = \emptyset \) yields standard argumentation frameworks without uncertainty. An attack-incomplete argumentation framework may be abbreviated as \( \langle \mathcal{A}, \mathcal{R}, \mathcal{R} \rangle \) and an argument-incomplete argumentation framework as \( \langle \mathcal{A}, \mathcal{A}^?, \mathcal{R} \rangle \).

Example 5. An argumentation framework can be identified with a directed graph by representing arguments as nodes and attacks as directed edges. Figure 1 is a graph representation of the incomplete argumentation framework \( \langle \mathcal{A}, \mathcal{A}^?, \mathcal{R}, \mathcal{R} \rangle \) with \( \mathcal{A} = \{b,c\} \), \( \mathcal{A}^? = \{a,d\} \), \( \mathcal{R} = \{(a,b),(b,b),(d,c)\} \), and \( \mathcal{R}^? = \{(b,c),(c,d)\} \), where definite elements are solid (circles for arguments or arrows for attacks), uncertain elements are dashed, and conditionally definite attacks are dash-dotted.

![Figure 1. Graph representation of the incomplete argumentation framework in Example 5](image)

A completion of an incomplete argumentation framework \( \mathcal{AF} = \langle \mathcal{A}, \mathcal{A}^?, \mathcal{R}, \mathcal{R} \rangle \) is any argumentation framework \( \mathcal{AF}^* = \langle \mathcal{A}^*, \mathcal{R}^* \rangle \) that satisfies \( \mathcal{A} \subseteq \mathcal{A}^* \subseteq \mathcal{A} \cup \mathcal{A}^? \) and \( \mathcal{R}|_{\mathcal{AF}^*} \subseteq \mathcal{R}^* \subseteq (\mathcal{R} \cup \mathcal{R}^?)|_{\mathcal{AF}^*} \). Here, the restriction \( \mathcal{R}|_{\mathcal{AF}^*} \) of an attack relation \( \mathcal{R} \) to \( \mathcal{A}^* \) is defined as \( \mathcal{R}|_{\mathcal{AF}^*} = \{(a,b) \in \mathcal{R} | a \in \mathcal{A}^* \} \). It represents the fact that conditionally definite attacks can only be part of a completion which includes that argument. However, a conditionally definite attack must be present in all completions containing both incident arguments, while an uncertain attack may vanish in a completion that contains both of its incident arguments. If at least one completion of an incomplete argumentation framework \( \mathcal{AF} \) satisfies some property, this property is said to hold possibly for \( \mathcal{AF} \). On the other hand, if all completions of \( \mathcal{AF} \) satisfy a property, it is said to hold necessarily for \( \mathcal{AF} \). Accordingly, we define both a possible and a necessary variant of the \( \mathcal{s} \)-CA and \( \mathcal{s} \)-SA problems for incomplete argumentation frameworks, for each semantics \( \mathcal{s} \) considered here:

<table>
<thead>
<tr>
<th>( \mathcal{s} )-NECESSARY-CREDULOUS-ACCEPTANCE (( \mathcal{s} )-NCA)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Given:</strong></td>
</tr>
<tr>
<td><strong>Question:</strong></td>
</tr>
</tbody>
</table>
We define **s-Possible-Credulous-Acceptance** (s-PCA) analogously to s-NCA, except that we now quantify *existentially* over all completions $AF^s$, and we define **s-Necessary-Skeptical-Acceptance** (s-NSA) analogously to s-PSA, except that we now quantify *universally* over all completions $AF^s$. Note that for the skeptical acceptance problems it is equivalent to ask whether each set of arguments that does not include $a$ is not $s$ in $AF^s$. This alternative quantifier formulation allows us to directly derive upper bounds. Due to Observations 2, 3, and 4, we have the following equalities: $\text{GR-PCA} = \text{GR-PSA} = \text{CP-PSA}$, $\text{GR-NCA} = \text{GR-NSA} = \text{CP-NSA}$, $\text{AD-PCA} = \text{PR-PCA}$, and $\text{AD-NCA} = \text{PR-NCA}$.

Note that we define the semantics of an incomplete argumentation framework through the completions. However, there are other ways for defining semantics of an incomplete argumentation framework as well. Cayrol et al. [5], for instance, restate the basic requirements of conflict-freeness and acceptability in the context of their “partial argumentation frameworks” (PAFs), consider related complexity issues, and establish links between semantics of PAFs and semantics of the completions.

### 3. Complexity Results

The number of completions for a given incomplete argumentation framework is exponential in the number of its uncertain elements. Therefore, possible and necessary problem generalizations are potentially harder than the respective baseline problem. In this section, we provide a full analysis of whether and how the computational complexity of the PCA, NCA, PSA, and NSA problem variants differs from that of CA and SA for the six semantics $\text{CF}$, $\text{AD}$, $\text{ST}$, $\text{CP}$, $\text{GR}$, and $\text{PR}$. For information about the relevant complexity classes of the polynomial hierarchy—in particular, P, NP, coNP, $\Sigma_2^p = \text{coNP}$, $\Sigma_3^p = \text{NP}^{\Sigma_2^p}$, and $\Sigma_3^p = \text{NP}^{\Sigma_2^p}$—as well as the concepts of hardness and completeness, we refer the reader to, e.g., Papadimitriou [14], Stockmeyer [16], and Rothe [15].

#### 3.1. Upper Bounds

We start with some simple P membership results. Since the answer to $\text{CF-SA}$ and $\text{AD-SA}$ is trivially “no” for all completions of an incomplete argumentation framework due to Observation 1, so is the answer to their possible and necessary generalizations s-PSA and s-NSA for $s \in \{\text{CF, AD}\}$, which are therefore in P, too. Further, both the possible and necessary generalizations of $\text{CF-CA}$ are in P, as stated in Proposition 6.

**Proposition 6.** $\text{CF-PCA}$ and $\text{CF-NCA}$ are in P.

The proof of Proposition 6 is omitted due to space limitations. For all remaining problems, from their quantifier representation we can derive upper bounds potentially higher than P. Matching lower bounds in Section 3.2 will prove these bounds to be tight.
representation of skeptical acceptance for the preferred semantics is 
∀we mean polynomially length-bounded existential or universal quantifiers.

The following, whenever we speak of "existential quantifiers" or "universal quantifiers,"

Proof. Due to space limitations, we only prove the last item of this proposition. In

Proposition 7. 1. For \( s \in \{ AD, ST, CP, GR, PR \} \) and for \( s' \in \{ CP, GR \}, \) \( s\)-PCA and
\( s'\)-PSA are in NP.

2. For \( s \in \{ ST, CP, GR \}, \) \( s\)-NSA and \( GR\)-NCA are in \( coNP. \)

3. For \( s \in \{ AD, ST, CP, PR \}, \) \( s\)-NCA is in \( \Pi_2^P. \)

4. \( ST\)-PSA is in \( \Sigma_2^P. \)

5. \( PR\)-NSA is in \( \Pi_2^P \) and \( PR\)-PSA is in \( \Sigma_2^P. \)

Proof. Due to space limitations, we only prove the last item of this proposition. In
the following, whenever we speak of “existential quantifiers” or “universal quantifiers,”
we mean polynomially length-bounded existential or universal quantifiers. A quantifier
representation of skeptical acceptance for the preferred semantics is “\( \forall \delta' \subseteq (\mathcal{A} \setminus \{ a \}) : \exists \delta' \supseteq \delta' \) : (\( \delta' \) is not AD in \( \mathcal{A}F \)) or (\( \delta' \) is AD in \( \mathcal{A}F \))”, where \( \mathcal{A}F = (\mathcal{A}, \mathcal{R}) \) and \( a \in \mathcal{A}. \)
For \( PR\)-NSA, this is preceded by a universal quantifier over completions which collapses
with the leading universal quantifier and provides \( \Pi_2^P \) membership. For \( PR\)-PSA, it is pre-
ceded by an existential quantifier over completions and provides \( \Sigma_2^P \) membership. \( \Box \)

3.2. Lower Bounds

Any hardness of the problems CA and SA is directly inherited by their possible and
necessary generalizations. For several of these generalizations, the upper bound from
Section 3.1 coincides with the lower bound inherited from CA and SA (cf. Table 2,
columns 2 and 5), which is stated in Corollary 8.

Corollary 8. 1. For \( s \in \{ AD, ST, CP, PR \}, \) \( s\)-PCA is NP-hard.

2. \( ST\)-NSA is \( coNP\)-hard.

3. \( PR\)-NSA is \( \Pi_2^P\)-hard.

For our proofs of the remaining hardness results, we reduce from different versions
of the satisfiability problem for quantified boolean formulas (QSAT), which are known
to be hard for different classes in the polynomial hierarchy. Table 1 gives a short defini-
tion of all used problems along with their complexity, where \( X, Y, \) and \( Z \) are disjoint sets
of propositional variables, \( \varphi \) denotes a formula in 3-CNF (conjunctive normal form with
at most three literals per clause) over the respective variables, \( \tau_5 \) is a truth assignment on
a set of literals \( S \) with \( \tau_5 : S \rightarrow \{ true, false \}, \) and \( \varphi|\tau_5 \) \) is the truth value that \( \varphi \) evaluates
to under \( \tau_5 \).

Table 1. Overview of quantified SAT problems used for hardness reductions

<table>
<thead>
<tr>
<th>instance</th>
<th>question</th>
<th>complexity</th>
</tr>
</thead>
<tbody>
<tr>
<td>3-SAT ((\varphi, X))</td>
<td>( \exists \tau_5 : \varphi</td>
<td>\tau_5 ) = true</td>
</tr>
<tr>
<td>( \Sigma_2) SAT ((\varphi, Y, X))</td>
<td>( \exists \tau_5 : \forall \tau_7 : \varphi</td>
<td>\tau_5, \tau_7 ) = false</td>
</tr>
<tr>
<td>( \Pi_2) SAT ((\varphi, X, Y, Z))</td>
<td>( \exists \tau_5 : \forall \tau_7 : \exists \tau_9 : \varphi</td>
<td>\tau_5, \tau_7, \tau_9 ) = true</td>
</tr>
<tr>
<td>3-UNSAT ((\varphi, X))</td>
<td>( \forall \tau_5 : \varphi</td>
<td>\tau_5 ) = false</td>
</tr>
<tr>
<td>( \Pi_2) SAT ((\varphi, X, Y))</td>
<td>( \forall \tau_5 : \exists \tau_7 : \varphi</td>
<td>\tau_5, \tau_7 ) = true</td>
</tr>
</tbody>
</table>

Table 1. Overview of quantified SAT problems used for hardness reductions

For our proofs of the remaining hardness results, we reduce from different versions
of the satisfiability problem for quantified boolean formulas (QSAT), which are known
to be hard for different classes in the polynomial hierarchy. Table 1 gives a short defini-
tion of all used problems along with their complexity, where \( X, Y, \) and \( Z \) are disjoint sets
of propositional variables, \( \varphi \) denotes a formula in 3-CNF (conjunctive normal form with
at most three literals per clause) over the respective variables, \( \tau_5 \) is a truth assignment on
a set of literals \( S \) with \( \tau_5 : S \rightarrow \{ true, false \}, \) and \( \varphi|\tau_5 \) \) is the truth value that \( \varphi \) evaluates
to under \( \tau_5 \).
we associate a given truth assignment \( \tau \) of variables with a set \( \mathcal{A} \) of arguments.

For an incomplete argumentation framework \( \mathcal{A} \), consider the attack-incomplete argumentation framework \( \mathcal{A} = \langle \mathcal{A}, \mathcal{R} \rangle \) (left) or an argument-incomplete argumentation framework \( \mathcal{A} = \langle \mathcal{A}, \mathcal{R}', \mathcal{R} \rangle \) (right):

\[
\mathcal{A} = \{ x, y, z, \text{ for } x \in X \}
\]

\[
\mathcal{R} = \{ (x, y), (y, z), (z, x) \}
\]

\[
\mathcal{R}' = \{ (x, z) \}
\]

\[
\mathcal{A} = \{ x, y, z, \text{ for } x \in X \}
\]

\[
\mathcal{R} = \{ (x, y), (y, z), (z, x) \}
\]

\[
\mathcal{R}' = \{ (x, z) \}
\]

All arguments \( x, y, \) and \( x, y, \) are called literal arguments and all arguments \( c_i \) clause arguments. All arguments in pairs \( a, a \) are called counterparts of each other. Argument \( g \) is without effect in the argument-incomplete version and only included there for uniformity.

For an incomplete argumentation framework \( \mathcal{A} \) created according to Definition 9, we associate a given truth assignment \( \tau \) on \( X \) with a completion \( \mathcal{A} \) of \( \mathcal{A} \). For an attack-incomplete argumentation framework \( \mathcal{A} = \langle \mathcal{A}, \mathcal{R}, \mathcal{R}' \rangle \), that completion has \( \mathcal{A} = \mathcal{A} \) and \( (g, x) \in \mathcal{R} \). For all arguments \( c_i \) in \( \mathcal{A} \), that completion has \( \mathcal{A} = \mathcal{A} \) and \( \mathcal{R} \). Further, we identify an assignment \( \tau \) on a set \( S = \{ x_1, \ldots, x_n \} \subseteq (X \cup Y) \) of variables with a set \( \mathcal{A} = \{ \tau \} \) of arguments in the completion, namely, \( \mathcal{A} = \{ x \} \) is an assignment on \( X \) and \( \mathcal{A} \) is a subset of the grounded extension and therefore contained in all complete extensions.

**Lemma 10.** Let \( \mathcal{A} \) be a \( \mathcal{A} \) instance \( \langle \mathcal{A}, \mathcal{R}, \mathcal{R}' \rangle \) or \( \mathcal{A} \) be an incomplete argumentation framework created for it according to Definition 9, and let \( \tau \) be an attack-incomplete argumentation framework \( \mathcal{A} \) of \( \mathcal{A} \) according to \( \tau \). In \( \mathcal{A} \), \( g \) attacks each argument \( x \) for which

\[
g \text{ is always unattacked and therefore clearly in the grounded extension. Consider the attack-incomplete argumentation framework } \mathcal{A} = \langle \mathcal{A}, \mathcal{R}, \mathcal{R}' \rangle \text{ and the completion } \mathcal{A} \text{ of } \mathcal{A} \text{ according to } \tau . \]
\( \tau_X(x_i) = true \), thus defending its counterpart \( x_i \), so these \( x_i \) are included in the grounded extension. All \( \bar{x}_j \) for which \( \tau_X(x_j) = false \) remain unattacked and are themselves included in the grounded extension, while they attack their counterparts \( x_j \), which thus are not included. Consider now the argument-incomplete argumentation framework \( AF = \langle \mathcal{A}, \mathcal{A}', \mathcal{R} \rangle \) and its completion \( AF^\tau \). Each argument \( x_i \) is always unattacked and therefore in the grounded extension if and only if it is included in the completion, which is the case for \( \tau_X(x_i) = true \). Each \( \bar{x}_j \) is only attacked by its counterpart \( x_j \) and therefore in the grounded extension if and only if that \( x_j \) is excluded from the completion, which is the case for \( \tau_X(x_j) = false \).

We now show a crucial correspondence between assignments in a QSAT instance and sets of arguments in the respective incomplete argumentation framework.

**Lemma 11.** Given a QSAT instance \( (\mathcal{Q}, X, Y) \) and full assignments \( \tau_X \) and \( \tau_Y \) \( (\tau_Y \) only if applicable). Let \( AF \) be an incomplete AF created for \( \mathcal{Q}, X, Y) \) following Definition 9, let \( AF^\tau \) be its completion corresponding to \( \tau_X \) and let \( \mathcal{A}^\tau \) \( \tau_X \), \( \tau_Y \) be the set of literal arguments corresponding to the total assignment.

- If \( \mathcal{Q}[\tau_X, \tau_Y] = true \), then \( \mathcal{A}^\tau \) is admissible, complete, preferred, and stable in \( AF^\tau \), and for \( Y = \emptyset \) also grounded.
- If \( \mathcal{Q}[\tau_X, \tau_Y] = false \), then \( \mathcal{A}^\tau \) is admissible, complete, preferred, and stable in \( AF^\tau \), and for \( Y = \emptyset \) also grounded.

**Proof.** Assume that \( \mathcal{Q}[\tau_X, \tau_Y] = true \). We know that \( \mathcal{A}^\tau \) is a subset of the grounded extension of \( AF^\tau \). We show that \( \mathcal{E} = \mathcal{A}^\tau \) is stable in \( AF^\tau \). It is easy to see from Definition 9 that \( \mathcal{E} \) is conflict-free, since there are no attacks between literal arguments for distinct literals, \( \mathcal{Q}, \) or \( g \). Further, \( \mathcal{E} \) attacks each argument in \( \mathcal{A}^\tau \) \( \mathcal{E} \). Argument \( \mathcal{Q} \) is attacked by \( \mathcal{Q} \in \mathcal{E} \). Each literal argument from \( X \) that does not occur in \( \mathcal{E} \) is either excluded from the completion, attacked by \( g \), or attacked by its counterpart in \( \mathcal{E} \). Each literal argument from \( Y \) that is not in \( \mathcal{E} \) is attacked by its counterpart in \( \mathcal{E} \). For each clause argument \( \bar{c}_i \), we know by assumption that the corresponding clause \( c_i \) in \( \mathcal{Q} \) is satisfied by the total assignment, since \( \mathcal{Q}[\tau_X, \tau_Y] = true \). Since \( c_i \) is satisfied, at least one literal in \( c_i \) must be satisfied. By construction of \( \mathcal{E} \) we know that at least one literal argument corresponding to a literal in \( c_i \) is in \( \mathcal{E} \), and by construction of \( AF \), this argument attacks the clause argument \( \bar{c}_i \). In total, this means that all clause arguments are attacked by \( \mathcal{E} \), and we proved that \( \mathcal{E} \) is stable in \( AF^\tau \). Since \( \mathcal{E} \) is stable, it is also preferred, complete, and admissible. For \( Y = \emptyset \), the set \( \mathcal{A}^\tau \) is a subset of the grounded extension by Lemma 10, already attacks all clause arguments and thus defends \( \mathcal{Q} \), so \( \mathcal{A}^\tau \) is the grounded extension of \( AF^\tau \).

Now assume that \( \mathcal{Q}[\tau_X, \tau_Y] = false \). Let \( \mathcal{E}' = \mathcal{A}^\tau \) be the subset \( \mathcal{C} = \{ \bar{c}_i \mid \bar{c}_i \in \mathcal{A}^\tau \} \) of \( \mathcal{E} \) is non-empty. Since \( \mathcal{Q}[\tau_X, \tau_Y] = false \), there is at least one clause \( c'_i \) in \( \mathcal{Q} \) that is not satisfied by the total assignment, so none of the literals in \( c'_i \) is satisfied. These literals correspond to literal arguments in \( AF \), which are the only arguments in \( AF \) that attack the clause argument \( \bar{c}_i \). By construction of \( \mathcal{E}' \), we know that none of these arguments are in \( \mathcal{E}' \), so \( \mathcal{E}' \) does not attack \( \bar{c}_i \) and thus \( \bar{c}_i \in \mathcal{C} \). We now show that \( \mathcal{E}' \) is stable in \( AF^\tau \). Again, \( \mathcal{E}' \) is clearly conflict-free. All literal arguments from \( X \) that do not occur in \( \mathcal{E}' \) are again either excluded from the completion, attacked by \( g \), or attacked by their
counterpart in $\delta'$. Each literal argument from $Y$ that is not in $\delta'$ is attacked by its counterpart in $\delta'$. Each clause argument that is not in $C$ is attacked by some $d \in \mathscr{A}^X[\tau_Y, \tau_Y]$ due to the definition of $C$. Finally, argument $\varphi$ is attacked by all arguments in $C \subseteq \delta'$, of which there is at least one since $C \neq \emptyset$. Since $\delta'$ is stable, it is also preferred, complete, and admissible. For $Y = \emptyset$, the set $\mathcal{A}^X[\tau_X] \cup \{g\}$, which is a subset of the grounded extension due to Lemma 10, already attacks all clause arguments in $\mathcal{A}^X \setminus C$ and thus defends all arguments in $C$, which in turn defend $\varphi$, so $\delta'$ is the grounded extension of $AF^X$. 

**Theorem 12.** GR-PCA is NP-hard.

**Proof.** We reduce from 3-SAT. Let $(\varphi, X)$ be a 3-SAT instance. If $(\varphi, X) \in 3$-SAT, we have $\exists \tau_X : \varphi[\tau_X] = \text{true}$, so by Lemma 11 there exists a completion of the corresponding argumentation framework $AF$ where $\varphi$ is in the grounded extension, and we have $(AF, \varphi) \in \text{GR-PCA}$. If $(\varphi, X) \notin 3$-SAT, we have $\forall \tau_X : \varphi[\tau_X] = \text{false}$, so $\varphi$ is in the grounded extension of all completions of the corresponding argumentation framework $AF$, so $\varphi$ cannot be in the grounded extension of any completion, and we have $(AF, \varphi) \notin \text{GR-PCA}$. 

Together with Observations 2 and 3, the following corollary follows immediately.

**Corollary 13.** GR-PSA and CP-PSA are NP-hard.

**Theorem 14.** GR-NCA is coNP-hard.

**Proof.** We reduce from 3-UNSAT. Let $(\varphi, X)$ be a 3-UNSAT instance. If $(\varphi, X) \in 3$-UNSAT, we have $\forall \tau_X : \varphi[\tau_X] = \text{false}$, so by Lemma 11, $\varphi$ is in the grounded extension of all completions of the corresponding argumentation framework $AF$ and we have $(AF, \varphi) \in \text{GR-NCA}$. If $(\varphi, X) \notin 3$-UNSAT, we have $\exists \tau_X : \varphi[\tau_X] = \text{true}$, so there exists a completion of the corresponding argumentation framework $AF$ where $\varphi$ is in the grounded extension, so $\varphi$ cannot be in the grounded extensions of all completions, and we have $(AF, \varphi) \notin \text{GR-NCA}$. 

Again, Observations 2 and 3 immediately give the following corollary.

**Corollary 15.** GR-NSA and CP-NSA are coNP-hard.

**Theorem 16.** For $s \in \{\text{AD, CP, ST, PR}\}$, s-NCA is $\Pi_2^P$-hard.

**Proof.** We reduce from $\Pi_2$SAT. Let $(\varphi, X, Y)$ be a $\Pi_2$SAT instance. If $(\varphi, X, Y) \in \Pi_2$SAT, we have $\forall \tau_X : \exists \tau_Y : \varphi[\tau_X, \tau_Y] = \text{true}$, so by Lemma 11, for all completions of the corresponding argumentation framework $AF$, there is a $\tau_Y$ such that the set $\mathcal{A}^X[\tau_X, \tau_Y] \cup \{\varphi, \varphi\}$ is admissible, complete, preferred, and stable, so $(AF, \varphi) \in s$-NCA for $s \in \{\text{AD, CP, ST, PR}\}$. If $(\varphi, X, Y) \notin \Pi_2$SAT, we have $\exists \tau_X : \forall \tau_Y : \varphi[\tau_X, \tau_Y] = \text{false}$, so there is a completion $AF_X^\tau$ of the corresponding argumentation framework $AF$ where $\mathcal{A}^X[\tau_X, \tau_Y] \cup \{\varphi, \varphi\} \cup \{c_i \mid d_i \in \mathcal{A}^X[\tau_X, \tau_Y] : (d, c_i) \in \mathcal{R}^X\}$ is stable for any choice of $\tau_Y$. This means that $\varphi$ cannot be a member of any admissible set in that completion—and therefore neither in a complete, stable, or preferred set—so $(AF, \varphi) \notin s$-NCA for $s \in \{\text{AD, CP, ST, PR}\}$. 


Figure 2. \( \text{PR-PSA instance created from clauses } c_1 = x_1 \lor \neg y_1 \lor \neg z_1 \text{ and } c_2 = y_1 \lor \neg y_2 \lor z_1 \text{ following a construction of } \text{Dunne and Bench-Capon [12].} \) Add either the framed part at the top to create an argument-incomplete argumentation framework or the one at the bottom for an attack-incomplete argumentation framework. A slight modification that uses \( c'_2 = y_1 \lor \neg y_2 \) instead of \( c_2 \) can be obtained by excluding the dotted attack \((z_1, c_2)\).

**Theorem 17.** ST-PSA is \( \Sigma^p_2 \)-hard.

**Proof.** We reduce from \( \Sigma_2 \text{ SAT} \). Let \((\varphi, X, Y)\) be a \( \Sigma_2 \text{ SAT} \) instance. If \((\varphi, X, Y) \in \Sigma_2 \text{ SAT} \), we have \( \exists \tau_X : \forall \tau_Y : \varphi[\tau_X, \tau_Y] = \text{false} \), so by Lemma 11, there is a completion \( \text{AF}^{\tau_X} \) of the corresponding argumentation framework \( \text{AF} \) where \( \forall \tau_X : \varphi[\tau_X, \tau_Y] \in \text{ST-PSA} \). If \((\varphi, X, Y) \notin \Sigma_2 \text{ SAT} \), we have \( \forall \tau_X : \exists \tau_Y : \varphi[\tau_X, \tau_Y] = \text{true} \), so for all completions of the corresponding argumentation framework \( \text{AF} \), there is some \( \tau_Y \) such that the set \( \text{AF}^{\tau_X} \) is stable in \( \tau_Y \) and \( \tau_Y \) is stable for any choice of \( \tau_Y \). Therefore, \((\varphi, X, Y) \notin \text{ST-PSA} \).

**Theorem 18.** \( \text{PR-PSA} \) is \( \Sigma^p_2 \)-hard.

**Proof.** Due to space constraints, we only sketch the proof of Theorem 18. To prove \( \Sigma^p_2 \)-hardness, we can extend a reduction that Dunne and Bench-Capon [12, Def. 13] used to prove \( \Pi^p_2 \)-hardness of \( \text{PR-SA} \) (see also the related work of Atkinson et al. [1]). Given an instance \((\varphi, X, Y, Z)\) of \( \Sigma_3 \text{ SAT} \), create an incomplete argumentation framework \( \text{AF} \) according to their construction using \( Y \) for their \( x \)-arguments, \( Z \) for their \( y \)-arguments, and clause arguments and \( \varphi \) for their gate arguments. In addition, create arguments from literals \( X \) along with argument \( g \) the same way as in our Definition 9.

If \((\varphi, X, Y, Z) \in \Sigma_3 \text{ SAT} \), there is a completion of \( \text{AF} \) in which, by their result, argument \( \varphi \) is skeptically preferred, so \((\varphi, X, Y, Z) \in \text{PR-PSA} \). If \((\varphi, X, Y, Z) \notin \Sigma_3 \text{ SAT} \), for all completions of \( \text{AF} \), there is a preferred extension that does not include \( \varphi \), so \((\varphi, X, Y, Z) \notin \text{PR-PSA} \).

**Example 19.** Consider a \( \Sigma_3 \text{ SAT} \) instance \((\varphi, \{x_1\}, \{y_1, y_2\}, \{z_1\})\), where \( \varphi = c_1 \land c_2 \) with \( c_1 = x_1 \lor \neg y_1 \lor \neg z_1 \) and \( c_2 = y_1 \lor \neg y_2 \lor z_1 \). Figure 2 displays a graph representation of the incomplete argumentation framework created for this instance of \( \Sigma_3 \text{ SAT} \): For \( \tau_X(x_1) = \text{true} \), any assignment \( \tau_Y \) on \( \{y_1, y_2\} \), and \( \tau_Z(z_1) = \text{true} \), we have \( \varphi[\tau_X, \tau_Y, \tau_Z] = \text{true} \). Accordingly, in the completion \( \text{AF}^{\tau_X} \) all preferred extensions are of the form \( \{g, x_1, z_1, \varphi\} \cup \text{AF}[\tau_Y] \) for some \( \tau_Y \), so \( \varphi \) is skeptically preferred.
Table 2. Overview of existing and new complexity results for credulous and skeptical acceptance problems, where existing results are ascribed to the respective source (references are numbers in brackets). Results marked with an asterisk (*) are straight-forward and and results due to this paper are marked by their theorem numbers.

<table>
<thead>
<tr>
<th></th>
<th>s-CA</th>
<th>s-PCA</th>
<th>s-NCA</th>
<th>s-SA</th>
<th>s-PSA</th>
<th>s-NSA</th>
</tr>
</thead>
<tbody>
<tr>
<td>CF</td>
<td>∈ P *</td>
<td>∈ P 6</td>
<td>∈ P 6</td>
<td>trivial*</td>
<td>trivial*</td>
<td>trivial*</td>
</tr>
<tr>
<td>AD</td>
<td>NP-c. [10]</td>
<td>NP-c. 8</td>
<td>Π_1^c. 16</td>
<td>trivial*</td>
<td>trivial*</td>
<td>trivial*</td>
</tr>
<tr>
<td>ST</td>
<td>NP-c. [10]</td>
<td>NP-c. 8</td>
<td>Π_1^c. 16</td>
<td>coNP-c. [10]</td>
<td>Σ_2^c. 17</td>
<td>coNP-c. 8</td>
</tr>
<tr>
<td>GR</td>
<td>∈ P *</td>
<td>NP-c. 12</td>
<td>coNP-c. 14</td>
<td>∈ P *</td>
<td>NP-c. 13</td>
<td>coNP-c. 15</td>
</tr>
<tr>
<td>PR</td>
<td>NP-c. [10]</td>
<td>NP-c. 8</td>
<td>Π_1^c. 16</td>
<td>Π_2^c. [12]</td>
<td>Σ_2^c. 18</td>
<td>Π_2^c. 8</td>
</tr>
</tbody>
</table>

When changing \( c_2 \) to \( c_2' = y_1 \lor \neg y_2 \) and \( \phi' = c_1 \land c_2' \), we obtain a “no” instance. For \( \tau_Y \) with \( \tau_Y(y_1) = false \) and \( \tau_Y(y_2) = true \), along with any assignments \( \tau_X \) and \( \tau_Z \), we have \( \phi' [\tau_X, \tau_Y, \tau_Z] = false \). In the corresponding argumentation framework, in both completions either \( \{g, x_1, y_1, y_2, \hat{c}_2\} \) or \( \{g, x_1, \hat{y}_1, y_2, \hat{c}_2\} \) is a preferred extension that does not include \( \psi \), so \( \phi \) is not skeptically preferred.

4. Conclusions and Relations to Other Models

Table 2 summarizes all complexity results of this paper and compares them to the existing results for CA and SA.

Compared to the possible and necessary variants of the verification problem for incomplete argumentation frameworks [3], which are not harder to solve than the respective baseline problem for many semantics, in this paper we observe a jump in complexity of necessary credulous acceptance and possible skeptical acceptance in almost all cases. This indicates that the presence of uncertainty, as described by attack incompleteness or argument incompleteness or both, is very likely to make acceptance problems harder.

Possible-credulous acceptance problems in incomplete argumentation frameworks are related to extension enforcement problems [2,8], where the question is, given an argumentation framework and a subset of its arguments, how the attacks and/or the arguments of the argumentation framework can be modified most efficiently such that the given set becomes part of an extension. Instances for acceptance problems in incomplete argumentation frameworks and for enforcement problems coincide if the incomplete argumentation framework has only uncertain attacks and no uncertain arguments, and if the enforcement instance allows only changes to the attack relation and its given subset is a singleton. However, enforcement problems aim at finding a minimal number of changes to the argumentation framework, which is not an aim in incomplete argumentation frameworks. On the other hand, the question of whether acceptance of the target argument can at all be achieved is trivially true in most variants of enforcement, while this is the key question for possible-credulous acceptance problems in incomplete argumentation frameworks.

The model of incomplete argumentation frameworks is further closely related to the recently proposed control argumentation frameworks (CAF) [9], which use a similar, yet much more specific formalism of uncertain elements in argumentation frameworks. Though technically similar, neither model can be fully expressed by the other: CAFs have no feature to represent uncertain attacks in “possible” problem variants, while in-
complete argumentation frameworks cannot express uncertain attacks where the attack itself is known, but its direction is not. However, there are various special cases where both models coincide. For example, “possible” problem variants in purely argument-incomplete argumentation frameworks can be represented by CAFs using their control-part, while “necessary” problem variants in incomplete argumentation frameworks can be represented by CAFs using their uncertain-part. The results of this paper may therefore be useful for the complexity analysis of similar problems in CAFs.

Acknowledgments: We thank the reviewers for many helpful suggestions. This work has been supported in part by the project “Online Participation,” funded by the NRW Ministry for Innovation, Science, and Research.

References