Proceedings of the 20th International Conference on Principles of Knowledge Representation and Reasoning
Main Track

Argumentative Reasoning in ASPIC$^+$ under Incomplete Information

Daphne Odekerken[1,2] , **Tuomo Lehtonen**[3] , **AnneMarie Borg**[1] ,
Johannes P. Wallner[4] and **Matti Järvisalo**[3]
[1]Utrecht University, The Netherlands
[2]National Police Lab AI, Netherlands Police, The Netherlands
[3]University of Helsinki, Finland
[4]Graz University of Technology, Austria
{d.odekerken, a.borg}@uu.nl, {tuomo.lehtonen, matti.jarvisalo}@helsinki.fi, wallner@ist.tugraz.at

Abstract

Reasoning under incomplete information is an important research direction in AI argumentation. Most computational advances in this direction have so far focused on abstract argumentation frameworks. Development of computational approaches to reasoning under incomplete information in structured formalisms remains to date to a large extent a challenge. We address this challenge by studying the so-called stability and relevance problems—with the aim of analyzing aspects of resilience of acceptance statuses in light of new information—in the central structured formalism of ASPIC$^+$. Focusing on the case of the grounded semantics and an ASPIC$^+$ fragment motivated through application scenarios, we develop exact ASP-based algorithms for stability and relevance in incomplete ASPIC$^+$ theories, and pinpoint the complexity of reasoning about stability (coNP-complete) and relevance (Σ_2^P-complete), further justifying our ASP-based approaches. Empirically, the algorithms exhibit promising scalability, outperforming even a recent inexact approach to stability, with our ASP-based iterative approach being the first algorithm proposed for reasoning about relevance in ASPIC$^+$.

1 Introduction

The study of computational aspects of argumentation is an important research direction in knowledge representation and reasoning (Atkinson et al. 2017; Baroni et al. 2018). Argumentation is intrinsically a dynamic process: justifications for different claims may arise or be withdrawn in light of new information, and the conclusions drawn may subsequently be altered. Indeed, going beyond the thoroughly-studied acceptance problems in static contexts, the study of computational aspects of dynamics in argumentation, with the aim of developing techniques for reasoning in argumentative settings under incomplete information, has recently attracted significant attention. A majority of works in this direction have so far focused on dynamics in abstract argumentation (Baumann and Brewka 2010; Cayrol, de Saint-Cyr, and Lagasquie-Schiex 2010; Maher 2016; Wallner, Niskanen, and Järvisalo 2017; Alfano, Greco, and Parisi 2017; Niskanen and Järvisalo 2020; Mailly and Rossit 2020; Odekerken, Borg, and Bex 2022; Cayrol, Devred, and Lagasquie-Schiex 2007; Baumeister et al. 2018; Baumeister et al. 2021) with fewer computational advances made for the arguably more complex structured argumentation formalisms (Maher 2016; Testerink, Odekerken, and Bex 2019; Odekerken, Borg, and Bex 2020; Borg and Bex 2021; Alfano et al. 2021; Rapberger and Ulbricht 2022; Odekerken et al. 2022a).

We focus on computational aspects of the recently-proposed notions of *stability* (Testerink, Odekerken, and Bex 2019) and *relevance* (Odekerken, Borg, and Bex 2022)—capturing forms of argumentation dynamics motivated by real-world application scenarios (Parsons, Wooldridge, and Amgoud 2002)—in the context of ASPIC$^+$ (Prakken 2010) as one the key structured argumentation formalisms (Besnard et al. 2014). The two notions provide perspectives to reasoning about justifiability of conclusions under incomplete information. Specifically, stability refers to checking whether the justification status of a conclusion cannot be altered by adding new information. Stability hence provides a key point of view, e.g., to argument-based inquiry, where the goal is to gather information on a possible conclusion: once a conclusion is stable, gathering additional information is no longer necessary. Relevance provides a point of view to reasoning about not (yet) stable conclusions. In such cases the choice of which additional information is gathered can play an important role in the efficiency of an inquiry application: to determine which yet unknown information should be investigated to ensure stability of a conclusion, only information that can change the stability status of a conclusion is relevant.

While the notion of stability was originally defined for ASPIC$^+$ (Testerink, Odekerken, and Bex 2019; Odekerken, Borg, and Bex 2020; Odekerken et al. 2022a) and subsequently studied also in the abstract setting (Mailly and Rossit 2020; Odekerken, Borg, and Bex 2022), relevance has to date only been studied in the abstract setting (Odekerken, Borg, and Bex 2022). From a formal perspective, the notions are naturally defined over *incomplete* ASPIC$^+$ theories which, in analogy with incomplete abstract argumentation frameworks (Cayrol, Devred, and Lagasquie-Schiex 2007; Baumeister et al. 2018; Baumeister et al. 2021), allow for modeling a set of possible "future theories", thereby enabling argumentative reasoning under incomplete information. From an application perspective, stability and relevance have been identified as fundamental reasoning problems underlying argument-based inquiry (Par-

sons, Wooldridge, and Amgoud 2002) in crime investigation where structured argumentation provides a natural model to practical and legal rules concerning crime (Odekerken, Borg, and Bex 2020; Odekerken et al. 2022a). For example, in investigating online trade fraud cases, investigative actions come with cost and understanding which actions may yield new information is important (Odekerken et al. 2022a). Despite such concrete application settings, currently the only algorithmic approach to stability in ASPIC$^+$ is a recently-proposed approximative approach which is not guaranteed to identify stability (Odekerken et al. 2022a). Furthermore, to the best of our knowledge, no practical algorithmic approaches to reasoning about relevance have been proposed, and the exact complexity of deciding stability and relevance in ASPIC$^+$ fragments with rule preferences is widely open.

With the main focus on a fragment of ASPIC$^+$ under grounded semantics motivated by applications of stability and relevance in crime investigation, the main contributions of this work encompass both theoretical and algorithmic advances. In terms of theory, we pinpoint the computational complexity of deciding stability and relevance, establishing coNP-completeness and Σ_2^P-completeness, respectively. In terms of algorithms, we develop the first exact approaches to reasoning about stability and relevance in ASPIC$^+$, based on the declarative paradigm of answer set programming (ASP) (Gelfond and Lifschitz 1988; Niemelä 1999), motivated by recent successful ASP-based approaches to reasoning about acceptance in (static) ASPIC$^+$ (Lehtonen, Wallner, and Järvisalo 2020; Lehtonen, Wallner, and Järvisalo 2022). Furthermore, we empirically evaluate an open-source implementation of the algorithms on both real-world and synthetic data, showing promising scalability, with our exact approach to stability outperforming the earlier proposed inexact approach. Formal proofs not included in the paper due to the page limit are available in an online supplement.

2 ASPIC$^+$

We recall ASPIC$^+$ as relevant for our discussion. The basic notion of ASPIC$^+$ is that of an argumentation system. We follow definitions by Modgil and Prakken (2013), incorporating a preorder $\leq$ as defined by Prakken (2010).

Definition 1 (Argumentation system)**.** *An argumentation system (AS) is a pair $AS = (\mathcal{L}, \bar{}, \mathcal{R}, \leq)$ with*

- *$\mathcal{L}$ a finite set of literals,*
- *$\bar{}$ a contradiction function from $\mathcal{L}$ to $2^{\mathcal{L}}$,*
- *$\mathcal{R}$ a finite set of defeasible rules of the form $a_1, \ldots, a_m \Rightarrow c$ with $\{a_1, \ldots, a_m, c\} \subseteq \mathcal{L}$, and*
- *$\leq$ a partial preorder (i.e., a reflexive and transitive binary relation) on $\mathcal{R}$.*

We say that l is a *contradictory* of m iff $m \in \bar{l}$ and $l \in \overline{m}$. Each $l \in \mathcal{L}$ has at least one contradictory and $l \notin \bar{l}$. In examples we will use classical negation as contradiction function: for each $x \in \mathcal{L}$, $\overline{x} = \{\neg x\}$ and $\overline{\neg x} = \{x\}$.

For a defeasible rule $r : a_1, \ldots, a_m \Rightarrow c$, $\texttt{ants}(r) = \{a_1, \ldots, a_m\}$ are the *antecedents* and $\texttt{cons}(r) = c$ is the *consequent* of r.

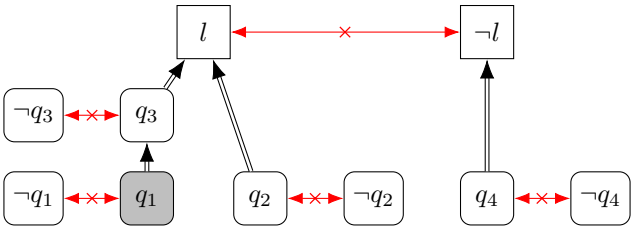

Figure 1: Example AT $T = (AS, \mathcal{K})$. Each square is a literal in $\mathcal{L}$: rounded squares are queryable literals, literals in $\mathcal{K}$ are shaded, double-lined arrows are rules, single-lined arrows negations.

An argumentation system gives rise to arguments with respect to a knowledge base. Here we assume that knowledge bases consist of axioms and are therefore consistent.

Definition 2 (Knowledge base)**.** *A knowledge base $\mathcal{K} \subseteq \mathcal{L}$ over an argumentation system $AS = (\mathcal{L}, \bar{}, \mathcal{R}, \leq)$ is a consistent set of literals, i.e., for any $l, m \in \mathcal{K}$ we have $l \notin \overline{m}$.*

Definition 3 (Argumentation theory)**.** *An argumentation theory (AT) $T = (AS, \mathcal{K})$ consists of an argumentation system AS and a knowledge base $\mathcal{K}$ over AS.*

An argumentation theory gives rise to arguments as follows.

Definition 4 (Arguments)**.** *The set of arguments Arg_T that an AT T gives rise to contains the arguments obtained applying the following implications finitely many times.*

- *If $c \in \mathcal{K}$, then $c \in Arg_T$ is an observation-based argument.*
- *If there is a rule $r : c_1, \ldots, c_m \Rightarrow c$ in $\mathcal{R}$ and $A_i \in Arg_T$ with $\texttt{conc}(A_i) = c_i$ for each $i = 1..m$, then $A_1, \ldots, A_m \Rightarrow c$ is a rule-based argument in Arg_T.*

For an observation-based argument c, the set of premises is $\texttt{prem}(c) = \{c\}$ the set of (defeasible) rules $\texttt{defrules}(c) = \emptyset$, the conclusion is $\texttt{conc}(c) = c$, and the set of subarguments is $\texttt{sub}(A) = \{c\}$. For a rule-based argument A, we have $\texttt{prem}(A) = \texttt{prem}(A_1) \cup \cdots \cup \texttt{prem}(A_m)$; $\texttt{defrules}(A) = \{r\} \cup \texttt{defrules}(A_1) \cup \cdots \texttt{defrules}(A_m)$; $\texttt{conc}(A) = c$; and $\texttt{sub}(A) = \texttt{sub}(A_1) \cup \cdots \cup \texttt{sub}(A_m) \cup \{A\}$. Furthermore, the top rule $\texttt{top-rule}(A)$ is r.

An argument with conclusion c is referred to as "an argument for c" and an argument with $\texttt{defrules}(A) \subseteq R \subseteq \mathcal{R}$ by "an argument based on R".

Example 1. *Let $T = (AS, \mathcal{K})$ (see Figure 1) be the AT over $AS = (\mathcal{L}, \bar{}, \mathcal{R}, \leq)$ where $\mathcal{L}$ consists of the literals q_1, q_2, q_3, q_4, l and their negations, $\mathcal{K} = \{q_1\}$, $\mathcal{R}$ consists of $q_1 \Rightarrow q_3$, $q_3 \Rightarrow l$, $q_2 \Rightarrow l$, $q_4 \Rightarrow \neg l$, and $\leq = \emptyset$. The set of arguments Arg_T consists of the observation-based argument q_1 and the rule-based arguments $q_1 \Rightarrow q_3$ and $[q_1 \Rightarrow q_3] \Rightarrow l$.*

In ASPIC$^+$, attacks between arguments are based on the structure of the arguments. We consider rebuttal attacks, where arguments attack each other on the conclusion of a defeasible inference.

Definition 5 (Rebuttal attack)**.** *Argument A rebuts argument B (on B') iff $\texttt{conc}(A) \in \overline{\texttt{conc}(B')}$ for some rule-based argument $B' \in \texttt{sub}(B)$.*

Example 2. *Recall the AT $T = (AS, \mathcal{K})$ from Example 1. None of the arguments in Arg_T attack any other argument in Arg_T. Alternatively, in the AT $T' = (AS, \mathcal{K} \cup \{\neg q_3\})$, there is an argument for $\neg q_3$ that attacks the argument for l (on the argument for q_3) and is not attacked by any argument in $Arg_{T'}$. Further, $T'' = (AS, \mathcal{K} \cup \{q_4\})$. contains two additional arguments compared to Arg_T: q_4 and $q_4 \Rightarrow \neg l$. The argument for $\neg l$ attacks the argument for l and vice versa.*

Not all rebuttals succeed as defeats: in ASPIC$^+$, this depends on the preference relation between arguments.

Definition 6 (Defeat)**.** *Argument A defeats argument B iff A rebuts B on B' and $A \not\prec B$.*

To compare arguments using $\prec$ (where $A \prec B$ iff $A \preceq B$ and $B \not\preceq A$), a notion for orderings that uses the partial preorder $\leq$ on $\mathcal{R}$ is needed. Four orderings were proposed by Modgil and Prakken (2013), based on combinations of the weakest and last-link principles and the choice of elitist or democratic comparisons on sets. For the ASPIC$^+$ fragment we consider, the elitist and democratic coincide. Here we focus on the last-link ordering.

Definition 7 (Last-link principle)**.** *Let A and B be two arguments on the basis of an AT. Under the last-link principle, $B \preceq A$ iff A is observation-based, or both A and B are rule-based and $\texttt{top-rule}(B) \leq \texttt{top-rule}(A)$.*

Each argumentation theory gives rise to an abstract argument framework. Semantics for argumentation theories are defined through semantics for the abstract frameworks (Dung 1995).

Definition 8 (AFs corresponding to ATs)**.** *An abstract argumentation framework (AF) defined by an AT $T = (AS, \mathcal{K})$ is a pair $\langle \mathcal{A}, \mathcal{C} \rangle$ with $\mathcal{A} = Arg_T$ and $\mathcal{C}$ the defeat relation on $\mathcal{A}$ determined by T.*

The specific ASPIC$^+$ instantiation we focus on in this work is motivated through applications in criminal investigation, as an extension of the instantiation used for inquiry dialogue at the Netherlands Police (Odekerken et al. 2022a). In such inquiry dialogues it is essential that reasoning is based on observations that are considered certain, which justifies only considering axioms and no attackable premises in the knowledge base. Furthermore, excluding strict rules makes it more feasible for police employees without background in computational argumentation to adapt or create rule sets—the design of argumentation theories with strict rules would require in-depth expertise to ensure that the rationality postulates (Caminada and Amgoud 2007) are satisfied. Requiring literals and rules to be finite is not restrictive for applications in e.g. criminal investigation where it is natural that only a limited number of rules and literals are used for capturing domain-specific information. Finally, all notions of conflict in the police use-case (Odekerken et al. 2022a) can be modelled using rebuttal attacks. It should be noted that, generalizing on (Odekerken et al. 2022a), we also allow preferences on rules in the instantiation considered in this work. Furthermore, we focus on grounded semantics. This is again motivated by practical applications: in criminal investigation, it is convenient to adopt a single-status semantics with a strong sceptical flavour, which is how grounded semantics can be intuitively characterized.

Definition 9 (Grounded extension)**.** *Let $F = \langle \mathcal{A}, \mathcal{C} \rangle$ be an AF and $S \subseteq \mathcal{A}$.*

- *S is conflict-free in F iff $(X, Y) \notin \mathcal{C}$ for each $X, Y \in S$.*
- *S defends $X \in \mathcal{A}$ in F iff for each $Y \in \mathcal{A}$ with $(Y, X) \in \mathcal{C}$, there is a $Z \in S$ with $(Z, Y) \in \mathcal{C}$.*
- *S is admissible in F iff S is conflict-free and S defends each $X \in S$.*
- *S is a complete extension of F iff S is admissible and, for each $X \in \mathcal{A}$, $X \in S$ if S defends X.*
- *The (unique) grounded extension S of F is the subset-minimal complete extension of F.*

For an AT T and corresponding argumentation framework F, $G(T)$ denotes the grounded extension of F.

Example 3. *In the AT $T = (AS, \mathcal{K})$ of Example 1, all arguments in Arg_T are undefeated and therefore in the grounded extension. Adding $\neg q_3$ to the knowledge base results in $T' = (AS, \mathcal{K} \cup \{\neg q_3\})$: in this AT, the arguments for q_1 and $\neg q_3$ are undefeated and therefore in the grounded extension, while the argument for l is defeated by an argument in the grounded extension. As for $T'' = (AS, \mathcal{K} \cup \{q_4\})$, neither the argument for l, nor the argument for $\neg l$ is in the grounded extension.*

In ASPIC$^+$, a statement is justified under grounded semantics if and only if there is a justified argument for the statement (Modgil and Prakken 2013, Definition 15). However, applications may in cases require a more fine-grained distinction between different types of justifications (Odekerken et al. 2022a). To this end, we consider four distinct justification statuses, including the special status *unsatisfiable* for literals for which there is no argument.

Definition 10 (Justification status)**.** *Let $T = (AS, \mathcal{K})$ be an AT where $AS = (\mathcal{L}, \bar{}, \mathcal{R}, \leq)$ and let $\langle \mathcal{A}, \mathcal{C} \rangle$ be the AF defined by T. The justification status of $l \in \mathcal{L}$ in T is*

- *unsatisfiable iff there is no argument for l in $\mathcal{A}$;*
- *defended iff there is an argument for l in $\mathcal{A}$ that is in the grounded extension $G(T)$;*
- *out iff there exists an argument for l in $\mathcal{A}$, but each argument for l in $\mathcal{A}$ is defeated by an argument in $G(T)$;*
- *blocked iff there is an argument for l in $\mathcal{A}$, no argument for l is in $G(T)$, and there is an argument for l that is not defeated by any argument in $G(T)$.*

The *defended* status corresponds to the justified status of conclusions of arguments (Modgil and Prakken 2013, Definition 15). Conclusions of arguments that are not justified can be either *out* or *blocked*. For some intuition, a literal that is *out* is not justifiable (every argument for the literal is defeated by the grounded extension). A literal that is *blocked* is not justified under the grounded semantics we focus in this paper, but might be justifiable for semantics other than grounded (Baroni, Caminada, and Giacomin 2011).

Example 4 (Justification statuses). *For argumentation system AS from Example 1, $\neg l$ is unsatisfiable wrt $T = (AS, \{q_1\})$; l is defended wrt $T = (AS, \{q_1\})$; l is out wrt $T' = (AS, \{q_1, \neg q_3\})$; and l is blocked wrt $T'' = (AS, \{q_1, q_4\})$.*

3 Stability and Relevance

We turn to the main focus of this work: stability and relevance. Stability can be seen as a dynamic variant on the justification status defined in the previous section: the justification status determines if a literal l is justified *given current information*. However, there are situations in which more information can be added, which possibly results in a change of l's justification status. If additional information cannot influence l's justification status, then we say that l is *stable*. We impose some restrictions on the allowed additions on the knowledge base, by distinguishing between queryable and non-queryable literals. Queryables are a specific set of literals that can be obtained (i.e. added to the axioms of the knowledge base) by querying the environment.

Definition 11 (Queryables). *Given an AT $T = (AS, \mathcal{K})$ with $AS = (\mathcal{L}, \bar{}, \mathcal{R}, \leq)$, a set of queryables $\mathcal{Q}$ is a set of literals such that $\mathcal{K} \subseteq \mathcal{Q} \subseteq \mathcal{L}$ and if $q \in \mathcal{Q}$ then $\overline{q} \subseteq \mathcal{Q}$.*

A set of queryables restricts the literals that can be added to the axioms of a knowledge base. Note that Definition 11 requires that all contradictories of each literal in $\mathcal{Q}$ are also in $\mathcal{Q}$. Adding a queryable literal q to the knowledge base of an AT $T = (AS, \mathcal{K})$ (where $\overline{q} \cap \mathcal{K} = \emptyset$) results in a new AT $T' = (AS, \mathcal{K} \cup \{q\})$. The set of all argumentation theories that can be obtained by adding queryables to the knowledge base is the set of future argumentation theories.

Definition 12 (Future argumentation theories). *Let $T = (AS, \mathcal{K})$ be an AT and $\mathcal{Q}$ a set of queryables. We say that AT $T' = (AS, \mathcal{K}')$ is a future argumentation theory of T, denoted by $T \sqsubseteq_Q T'$, if $\mathcal{K} \subseteq \mathcal{K}' \subseteq \mathcal{Q}$.*

We define a strict variant $T \sqsubset_Q T'$ by $T \sqsubseteq_Q T'$ and $T' \not\sqsubseteq_Q T$. By definition, $T \sqsubseteq_Q T$. Also note that, since all future argumentation theories are argumentation theories in the sense of Definition 3, the axioms in their knowledge base must be consistent.

We distinguish four types of stability, relative to the four justification statuses from Definition 10.

Definition 13 (j-stability). *Let $T = (AS, \mathcal{K})$ be an AT and $\mathcal{Q}$ is a set of queryables. Given a literal $l \in \mathcal{L}$ and a justification status j in $\{$unsatisfiable, defended, out, blocked$\}$, l is stable-j in T wrt $\mathcal{Q}$ iff l is j in T' for each T' with $T \sqsubseteq_Q T'$.*

Example 5 (Stability statuses). *Consider the argumentation system AS from Example 1 and let $\mathcal{Q} = \{q_1, q_2, q_3, q_4, \neg q_1, \neg q_2, \neg q_3, \neg q_4\}$. We have that $\neg l$ is stable-unsatisfiable wrt $(AS, \{q_1, \neg q_4\})$ and $\mathcal{Q}$; l is stable-defended wrt $(AS, \{q_1, q_3, \neg q_4\})$ and $\mathcal{Q}$; l is stable-out wrt $(AS, \{q_1, \neg q_2, \neg q_3\})$ and $\mathcal{Q}$; and l is stable-blocked wrt $(AS, \{q_1, q_3, q_4\})$ and $\mathcal{Q}$.*

When a literal does not have a stable status, i.e., there is a future AT that changes the justification status of the literal, a natural question to ask is which queryables are *relevant* for making the literal stable, i.e., which queryables should be added to the knowledge base in order to obtain an AT where this literal is stable. This is captured by the notion of relevance, recently introduced in the context of incomplete (abstract) argumentation frameworks (Odekerken, Borg, and Bex 2022). Here we propose an analogous definition of relevance for ASPIC$^+$, based on the notion of minimal stable future ATs, i.e., future ATs where the knowledge base is minimally expanded and the considered literal is stable.

Definition 14 (Minimal stable-j future theory). *Let $T = (AS, \mathcal{K})$ be an AT, $\mathcal{Q}$ be a set of queryables, and j be a justification status. Given an $l \in \mathcal{L}$, a minimal stable-j future theory for l wrt T and $\mathcal{Q}$ is an AT T' with $T \sqsubseteq_Q T'$ s.t. (i) l is stable-j in T', and (ii) there is no T'' such that l is stable-j in T'' and $T \sqsubseteq_Q T'' \sqsubset_Q T'$.*

Example 6 (Minimal stable-j future theory). *Consider again the AT $T = (AS, \mathcal{K})$ from Example 1 with $\mathcal{Q} = \{q_1, q_2, q_3, q_4, \neg q_1, \neg q_2, \neg q_3, \neg q_4\}$. We have that $\neg l$ is stable-unsatisfiable wrt $T' = (AS, \{q_1, q_2, \neg q_4\})$ and $\mathcal{Q}$, but T' is not a minimal stable-unsatisfiable future theory for $\neg l$ wrt T and $\mathcal{Q}$, since $\neg l$ would also be stable without q_2. The future AT $(AS, \{q_1, \neg q_4\})$ is minimal stable-unsatisfiable. There are two minimal stable-defended future theories for l wrt $(AS, \{q_1\})$ and $\mathcal{Q}$: $(AS, \{q_1, q_3, \neg q_4\})$ and $(AS, \{q_1, q_2, \neg q_4\})$.*

Literals in the knowledge base of a minimal stable-j future theory that do not occur in the original knowledge base are considered relevant.

Definition 15 (j-relevance). *Let $T = (AS, \mathcal{K})$ be an AT with $AS = (\mathcal{L}, \bar{}, \mathcal{R}, \leq)$, let $\mathcal{Q}$ be a set of queryables and let j be a justification status. Given $l \in \mathcal{L}$ and $q \in Q$ with $q \notin \mathcal{K}$ and $\overline{q} \cap \mathcal{K} = \emptyset$, we say that q is j-relevant for l wrt T and $\mathcal{Q}$ iff there is a minimal stable-j future theory $T' = (AS, \mathcal{K}')$ for l wrt T and $\mathcal{Q}$ such that $q \in \mathcal{K}'$.*

Example 7 (j-relevance). *Continuing Example 1, for the AT $T = (AS, \mathcal{K})$ and $\mathcal{Q} = \{q_1, q_2, q_3, q_4, \neg q_1, \neg q_2, \neg q_3, \neg q_4\}$ we find that $\neg q_4$ is the only literal that is unsatisfiable-relevant for $\neg l$ wrt T. The literals that are defended-relevant for l wrt T and $\mathcal{Q}$ are $\{q_2, q_3, \neg q_4\}$.*

Note that it is possible that a queryable and its negation are both relevant for a given topic literal:

Example 8. *Consider the AT $T = (AS, \mathcal{K})$ where $AS = (\mathcal{L}, \bar{}, \mathcal{R}, \leq)$, $\mathcal{L} = \{q, \neg q, l, \neg l\}$, $\mathcal{R} = \{q \Rightarrow l, \neg q \Rightarrow l\}$ and $\mathcal{K} = \emptyset$. Suppose that $\mathcal{Q} = \{q, \neg q\}$. Then both q and $\neg q$ are defended-relevant for l wrt T since l is unsatisfiable in T and defended in both $(AS, \{q\})$ and $(AS, \{\neg q\})$.*

4 Complexity Results

As the main complexity-theoretic contributions, we pinpoint the complexity of deciding for a given literal the (i) justification status of the literal, (ii) stability of the literal and (iii) relevance of the literal, under the four different justification statuses in the ASPIC$^+$ fragment considered. Specifically, we show that the justification status of a literal is decidable in polynomial time. Moreover, we show that the stability problems for the four justification statuses are coNP-

complete and establish Σ_2^P-completeness for deciding relevance. We begin with the complexity of the justification problem, which lays the ground for the other results.

4.1 Rephrasing Grounded Semantics

Note that P-membership for the justification problem is not immediately clear from the definitions, since Definition 9 specifies the grounded extension in terms of arguments and an AT can have a number of arguments that is not bounded polynomially in the AT size. An example that exhibits an exponential number of arguments for given rule based theories is given by Strass, Wyner, and Diller (2019). In order to provide polynomial-time decidability, we reformulate the grounded extension, in terms of sets of *rules* rather than sets of *arguments*. In order to do so, we use notions of applicability, defeats and defence on sets of rules.

First, a rule is applicable by a set of rules if it is possible to construct an argument using only rules from the set.

Definition 16 (Applicable by rule set)**.** *Given an AT $T = (AS, \mathcal{K})$ and a set of defeasible rules $D \subseteq \mathcal{R}$, define $Arg_T(D)$ as the set of all arguments that can be constructed using $\mathcal{K}$ and D. We say that a rule $r \in \mathcal{R}$ is applicable by D iff there is an argument A based on $D \cup \{r\}$ with $r \in \texttt{defrules}(A)$.*

Turning to the general case with preferences included, we now define defeat in terms of rule sets (which is comparable to Definition 6 for defeat of arguments).

Definition 17 (Defeat by rule set)**.** *Given an AT $T = (AS, \mathcal{K})$, a set of defeasible rules $D \subseteq \mathcal{R}$, and a rule $r \in \mathcal{R}$, we say that D defeats r iff (i) there is some r' in D such that $\texttt{cons}(r') \in \overline{\texttt{cons}(r)}$, r' is applicable by D, and $r' \not< r$; or (ii) there is some l in $\mathcal{K}$ such that $l \in \overline{\texttt{cons}(r)}$.*

In words, a set of defeasible rules D defeats a single rule r if (i) there is a rule r' applicable by D that has as consequent a contradictory of the consequent of r and is not strictly less preferred to r, or (ii) that the knowledge base contains an axiom contradictory to the consequent of r. Intuitively, in the former case, a rule-based argument with r as top rule is defeated by a rule-based argument that has as its defeasible rules only rules in D and r' as its top rule. Then the rebut succeeds as a defeat. In the latter case, an observation-based argument directly defeats any argument with r as its top rule.

Example 9. *Continuing Example 3, consider again AT $T'' = (AS, \mathcal{K} \cup \{q_4\})$. Then $D = \{(q_1 \Rightarrow q_3), (q_3 \Rightarrow l)\}$ defeats rule $q_4 \Rightarrow \neg l$, since the consequence of one of the rules in D and the latter rule are contradictory to each other. The set $D' = \{q_4 \Rightarrow \neg l\}$ defeats $q_3 \Rightarrow l$. If we strictly prefer $q_3 \Rightarrow l$ to $q_4 \Rightarrow \neg l$ then D' does not defeat $q_3 \Rightarrow l$.*

Stated as follows, there is a correspondence between defeats by rule sets and defeats by arguments.

Proposition 1. *Given an AT $T = (AS, \mathcal{K})$, the corresponding AF $\langle \mathcal{A}, \mathcal{C} \rangle$, and a set of defeasible rules D and an argument $A \in \mathcal{A}$, it holds that at least one $B \in Arg_T(D)$ defeats A if and only if D defeats a rule $r \in \texttt{defrules}(A)$.*

Analogously to defeat, we also introduce a notion of defence in terms of rule sets. An argument A is defended by a set of arguments S if each argument B defeating A is defeated by some argument in S. In other words, each argument that is not defeated by any argument in S must not defeat A. We rephrase this aspect into defence on rule sets.

Definition 18 (Defence by rule set)**.** *Given an $T = (AS, \mathcal{K})$, a set of defeasible rules $D \subseteq \mathcal{R}$, and a rule $r \in \mathcal{R}$, let U be the set of rules in $\mathcal{R}$ that are not defeated by D. Then r is defended by D iff U does not defeat r.*

In the following formal results, we show the correspondence between defence by arguments (Definition 9) and defence by rule sets (Definition 18).

Proposition 2. *Given an $T = (AS, \mathcal{K})$, the corresponding AF $\langle \mathcal{A}, \mathcal{C} \rangle$, and a set of defeasible rules D, and an argument $A \in \mathcal{A}$, it holds that $Arg_T(D)$ defends A if and only if D defends every rule $r \in \texttt{defrules}(A)$.*

Based on the notion of defence for rule sets, we next define a counterpart of Dung's fundamental lemma (Dung 1995, Lemma 10) for rule sets rather than argument sets.

Proposition 3. *Let $T = (AS, \mathcal{K})$ be an AT where $AS = (\mathcal{L}, \bar{}, \mathcal{R}, \leq)$, $R \subseteq \mathcal{R}$ be a set of defeasible rules such that (i) each rule $r \in R$ is applicable by R and (ii) $Arg_T(R)$ is admissible. Let r and r' be rules in $\mathcal{R}$ defended by R. Then (1) $Arg_T(R \cup \{r\})$ is admissible and (2) $R \cup \{r\}$ defends r'.*

Towards defining the grounded extension without computing arguments, we define a characteristic function for rule sets alternative to the "classical" version of the characteristic function (Dung 1995, Definition 16).

Definition 19. *Let $T = (AS, \mathcal{K})$ be an AT where $AS = (\mathcal{L}, \bar{}, \mathcal{R}, \leq)$ and $D \subseteq \mathcal{R}$ a rule set. Then $def_T(D) = \{r \in \mathcal{R} \mid r \text{ is applicable and defended by } D\}$.*

We denote i applications of def_T on $\emptyset$ by $def_T^i(\emptyset)$ for $i > 0$ and define $def_T^0(\emptyset) = \emptyset$.

By Proposition 4, iterating the characteristic function starting from the empty set gives the grounded extension.

Proposition 4. *Given an AT $T = (AS, \mathcal{K})$ where $AS = (\mathcal{L}, \bar{}, \mathcal{R}, \leq)$, let C be the least fixpoint of def_T. Then $G(T) = Arg_T(C)$.*

Example 10. *Consider again Example 3 and AT $T'' = (AS, \mathcal{K} \cup \{q_4\})$. It holds that $\{q_1 \Rightarrow q_3\}$ is the least fixpoint of $def_{T''}$. Therefore $G(T)$ contains arguments for q_1 and q_3, but not for, e.g., q_4, l or $\neg l$.*

4.2 Complexity of Justification

We show that one can compute the least fixpoint in $|\mathcal{R}|/2$ iterations, starting with the empty set of rules. At the fixpoint we conclude a rule to be defended or defeated.

Proposition 5. *Given an AT $T = (AS, \mathcal{K})$ where $AS = (\mathcal{L}, \bar{}, \mathcal{R}, \leq)$, the least fixpoint of def_T is reached in at most $|\mathcal{R}|/2$ iterations.*

The least fixpoint of def_T allows for directly inferring the justification status of a literal.

Proposition 6. *Given an AT $T = (AS, \mathcal{K})$ where $AS = (\mathcal{L}, \bar{}, \mathcal{R}, \leq)$, and C be the least fixpoint of def_T. A literal $l \in \mathcal{L}$ is*

- *unsatisfiable if there is no argument $A \in Arg_T$ with $\texttt{conc}(A) = l$,*
- *defended if there is an argument $A \in Arg_T$ with $\texttt{conc}(A) = l$ and $\texttt{defrules}(A) \subseteq C$,*
- *out if no argument in Arg_T with conclusion l is based on U, where U is the set of rules not defeated by C, and*
- *blocked otherwise.*

Putting our results together, it holds that we can infer the justification status of a literal in polynomial time for each of the four justification statuses.

Theorem 1. *Let j be the unsatisfiable, defended, out, or blocked justification status. The problem of deciding whether a literal has justification status j is in P.*

4.3 Complexity of Stability

Polynomial-time decidability of justification (Theorem 1) has implications on the complexity of stability. Specifically, to decide whether a literal is stable wrt a justification status, we can proceed as follows: non-deterministically guess a future theory and deterministically check (in polynomial time by Theorem 1) the justification status of the targeted literal. Thus, the complementary problem, i.e., a literal is not stable wrt a justification status, is in NP. In addition to membership in coNP, we can infer coNP-hardness from earlier results (Odekerken et al. 2022a) which imply hardness for the case without preferences.

Proposition 7. *Deciding whether a literal is j-stable in an AT is coNP-complete for each justification status $j \in \{$unsatisfiable, defended, out, blocked$\}$. Hardness holds even without preferences.*

4.4 Complexity of Relevance

We turn to the problem of deciding whether a given queryable is j-relevant for a given literal for a justification status j: the problem turns out to be Σ_2^P-complete for each of the four justification statuses. We first show an auxiliary result that characterizes relevance of literals in terms of checking (non-)stability. Intuitively, we can verify that a queryable q is j-relevant for a literal l if we are able to find an AT in which the literal is not stable-j, but when adding q to the axioms, stability holds.

Lemma 1. *Let $T = (AS, \mathcal{K})$ be an AT, let $\mathcal{Q}$ be a set of queryables and let j be a justification status. Given a literal $l \in \mathcal{L}$ and a queryable literal $q \in Q$ where $q \notin \mathcal{K}$ and $\overline{q} \cap \mathcal{K} = \emptyset$, q is j-relevant for l wrt T and $\mathcal{Q}$ iff*

- *there is an AT $T' = (AS, \mathcal{K}')$ with $T \sqsubseteq_Q T'$ such that l is not stable-j wrt T' and*
- *l is stable-j wrt $(AS, \mathcal{K}' \cup \{q\})$.*

A direct use of Lemma 1 is to show membership results in Σ_2^P for relevance for all justification statuses considered in this paper. We also prove hardness via a reduction from quantified Boolean formulas.

Theorem 2. *Deciding whether a queryable is j-relevant for a literal in an AT wrt a set of queryables is Σ_2^P-complete for each $j \in \{$unsatisfiable, defended, out, blocked$\}$. Hardness holds even without preferences.*

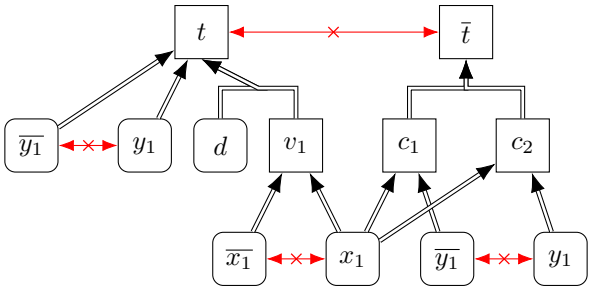

Figure 2: Illustration of the reduction used in Theorem 2 for the formula $\phi = (x_1 \vee y_1) \wedge (x_i \vee \neg y_1)$. The queryables $\overline{y_1}$ and y_1 are displayed twice for readability.

Proof sketch for defended status. Membership in Σ_2^P follows from Lemma 1: a positive instance can be verified with two calls to an oracle for stability, which is coNP-complete by Proposition 7. For Σ_2^P-hardness, we detail here a proof for the defended justification status. We reduce from the Σ_2-SAT problem of deciding for a formula ϕ in CNF, quantified over X and Y where $X = \{x_1, \ldots, x_n\}$ and $Y = \{y_1, \ldots, y_m\}$ are pairwise disjoint sets, if there is an assignment τ_X to variables in X such that for each assignment τ_Y to variables in Y, $\phi[\tau_X, \tau_Y] = \text{False}$. Construct the following AT T and queryables (see Figure 2), with $C = c_1 \wedge \cdots \wedge c_p$ the set of clauses in ϕ, and $\overline{X} = \{\overline{x} \mid x \in X\}$, $\overline{Y} = \{\overline{y} \mid y \in Y\}$ and $\overline{C} = \{\overline{c} \mid c \in C\}$. Let $V = \{v_i \mid x_i \in X\}$ and $\overline{V} = \{\overline{v_i} \mid x_i \in X\}$. Define $\mathcal{Q} = X \cup \overline{X} \cup Y \cup \overline{Y} \cup \{d, \overline{d}\}$, language $\mathcal{L} = \mathcal{Q} \cup C \cup \overline{C} \cup V \cup \overline{V} \cup \{t, \overline{t}\}$, $\mathcal{K} = \emptyset$, contraries $\overline{} = \{(x, \overline{x}), (\overline{x}, x) \mid x \in X \cup Y \cup V \cup C \cup \{d, t\}\}$, and

$$\begin{aligned}\mathcal{R} = \{&(d, v_1, \ldots, v_n \Rightarrow t)\} \cup \\ &\{(x \Rightarrow c) \mid x \in c\} \cup \{(\overline{x} \Rightarrow c) \mid \neg x \in c\} \cup \\ &\{(y \Rightarrow c) \mid y \in c\} \cup \{(\overline{y} \Rightarrow c) \mid \neg y \in c\} \cup \\ &\{(c_1, \ldots, c_p \Rightarrow \overline{t})\} \cup \\ &\{(x_i \Rightarrow v_i), (\overline{x_i} \Rightarrow v_i) \mid x_i \in X\} \\ &\{(y \Rightarrow t), (\overline{y} \Rightarrow t) \mid y \in Y\}.\end{aligned}$$

The AT $T = (AS, \mathcal{K})$ and $\mathcal{Q}$ can be constructed in polytime wrt ϕ. In addition, (ϕ, X, Y) is a satisfiable Σ_2-SAT instance iff d is defended-relevant for t wrt T. □

Summarizing, the complexities of the justification, stability, and relevance problems exhibit a clear jump: from "in P" to coNP-completeness and to Σ_2^P-completeness. We note that our hardness results hold even without the use of preferences. The result for deciding the justification status required a reformulation in terms of defeasible rules, a representation bounded polynomially in terms of a given AT. The membership results all rely on this reformulation.

5 Algorithms for Stability and Relevance

Complementing our complexity results, we develop declarative algorithms for deciding stability and relevance based on the declarative paradigm of answer set programming (ASP) (Gelfond and Lifschitz 1988; Niemelä 1999).

Listing 1: Module π_{common}

```
1 literal(L) ← head(_,L). literal(L) ← body(_,L).
2 literal(L) ← axiom(L). literal(L) ← ctr(L,_).
3 rule(R) ← head(R,_).
4 ctr(X,Y) ← ctr(Y,X).
5 derivable(L) ← axiom(L).
6 derivable(L) ← head(R,L), applicable_rule(R).
7 applicable_rule ← rule(R), derivable(L) : body(R,L).
8 unsat(L) ← not derivable(L), literal(L).
```

5.1 Encoding Justification Status

In this section we present ASP encodings for deciding the justification status of literals for a given AT $T = (AS, \mathcal{K})$. These encodings will be used for computing stability and relevance. The AT (and queryable set $\mathcal{Q}$) is represented as the set of facts $\mathtt{AT}(T)$, defined as follows:

$$\{\mathbf{axiom}(a). \mid a \in \mathcal{K}\} \cup$$
$$\{\mathbf{queryable}(a). \mid a \in \mathcal{Q}\} \cup$$
$$\{\mathbf{head}(r, b). \mid r \in \mathcal{R}, b = head(r)\} \cup$$
$$\{\mathbf{body}(r, b). \mid r \in \mathcal{R}, b \in body(r)\} \cup$$
$$\{\mathbf{ctr}(a, b). \mid b \in \overline{a}, a \in \mathcal{L}\} \cup$$
$$\{\mathbf{preferred}(a, b). \mid b \leq a, a, b \in \mathcal{R}\}.$$

Listing 1 is used by all encodings. Lines 1–3 collect literals and rules in the AT, and Line 4 enforces that contradiction is symmetric. Lines 5–7 determine which literals are derivable, and Line 8 collects unsatisfiable literals. Line 7 uses a conditional literal "$\mathbf{derivable}(L) : \mathbf{body}(R, L)$", representing a list of $\mathbf{derivable}(L)$ whenever $\mathbf{body}(R, L)$ holds.

We present two separate ASP encodings for deciding the justification status of literals: one ($\pi_{<\text{-}just}$) taking rule preferences into account, the other (π_{just}) assuming that $\leq = \emptyset$. Whereas π_{just} is conceptually simpler and can be used in a comparison to the stability algorithm by (Odekerken et al. 2022a), $\pi_{<\text{-}just}$ is more generally applicable.

No preferences For ATs without rule preferences, arguments can only be in the grounded extension if they are defended by observation-based arguments:

Lemma 2 (Odekerken et al. 2022b, Lemmas 4–5)**.** *Given an AT $T = (AS, \mathcal{K})$ with $AS = (\mathcal{L}, \overline{\ }, \mathcal{R}, \leq)$ where $\leq = \emptyset$, an argument $A \in Arg_T$ is in $G(T)$ iff each argument defeating A is defeated by an observation-based argument, and defeated by an argument in $G(T)$ iff A is defeated by an observation-based argument.*

This property is exploited in the following proposition, in which we collect rules U not defeated by axioms and defended rules def not defeated by arguments based on U.

Proposition 8. *Given an $T = (AS, \mathcal{K})$ with $\leq = \emptyset$, let $U = \{r \in \mathcal{R} \mid \overline{\mathtt{cons}(r)} \cap \mathcal{K} = \emptyset\}$. A literal l is labelled*

- *defended if there is an argument A for l such that $\mathtt{defrules}(A) \subseteq def$, where $def = \{r \in \mathcal{R} \mid$ there is no argument for $\overline{\mathtt{cons}(r)}$ based on $U\}$, and*
- *out if l is not unsatisfiable and there is no argument A for l with $\mathtt{defrules}(A) \subseteq U$*

The encoding for assigning the justification label of each literal is the program $\pi_{just} = \pi_{common} \cup \Delta_{just}$. The module Δ_{just} (Listing 2) assigns the justification labels other than unsatisfiable, following Proposition 8. Lines 1–4 of Δ_{just} collect the rules that are undefeated by axioms and the literals that can be concluded via them. Line 5 states that a literal is out if it is derivable, but not concluded via undefeated rules. Lines 6–9 of Δ_{just} collect defended literals by considering derivations from rules that are undefeated by rules undefeated by axioms. Finally, yet unlabelled literals are labelled as blocked in Line 10 (Odekerken et al. 2022a).

Preferences For ATs with preferences, we present an encoding $\pi_{<\text{-}just} = \pi_{common} \cup \Delta_{<\text{-}just}$ based on the least fixpoint of the defence operator (Definition 19), from which one can infer the justification labels by Proposition 6. The module $\Delta_{<\text{-}just}$ (Listing 3) encodes a sequence of applications of the defence operator with explicit indices (up to $|\mathcal{R}|/2$, per Proposition 5). Line 1 encodes transitivity of preferences and Lines 2–3 when a rule is not strictly less preferred than another. Lines 4–6 set the iteration upper bound. For clarity, for each iteration i we denote here the set of defended rules by D^i and rules not defeated by D^i by U^i (corresponding to **defended_rule** and **undefeated** in $\Delta_{<\text{-}just}$). On Lines 7–8, a literal is deemed defended on iteration i if it can be derived by only using rules from D^i. On Lines 9–10, D^i is identified as the applicable rules that are not defeated by U^{i-1}, corresponding to the defence operator. The rules that D^i defeats are identified on Lines 11–12, following Definition 17: $r \in \mathcal{R}$ is defeated if either an axiom contradicts r or D^i induces an argument whose top rule is not less preferred than r and that concludes $\overline{\mathtt{cons}(r)}$. Based on rules defeated by D^i, the undefeated rules U^i and the literals derivable from U^i are identified on Lines 13–15. Defeats from U^i are identified on Lines 16–17. Finally, Lines 18–21 label the literals based on the final iteration.

5.2 Encoding Stability

The stability status of a literal is obtained by checking if there is a future AT where the literal is not j for a justification status j. We implement this by conjoining our encodings for justification with a non-deterministic guess of future ATs. This is achieved by guessing a future theory by

Listing 2: Module Δ_{just}

```
1 defeated(R) ← head(R,X), axiom(Y), ctr(X,Y).
2 undefeated(L) ← axiom(L).
3 undefeated(L) ← head(R,L), undefeated_rule(R).
4 undefeated_rule(R) ← rule(R), not defeated(R),
      undefeated(L) : body(R,L).
5 out(L) ← derivable(L), not undefeated(L).
6 defeated_by_undefeated(R) ← head(R,X), undefeated(Y),
      ctr(X,Y).
7 defended(L) ← axiom(L).
8 defended(L) ← head(R,L), defended_rule(R).
9 defended_rule(R) ← not defeated_by_undefeated(R),
      rule(R), defended(L) : body(R,L).
10 blocked(L) ← literal(L), not unsat(L),
      not out(L), not defended(L).
```

Listing 3: Module $\Delta_{<\text{-}just}$

```
1  preferred(X,Z) ← preferred(X,Y), preferred(Y,Z).
2  strictly_less_preferred(X,Y) ← not preferred(X,Y),
       preferred(Y,X).
3  no_less_preferred(X,Y) ← rule(X), rule(Y),
       not strictly_less_preferred(X,Y).
4  n_rules(N) ← #count{X : rule(X)} = N.
5  max_iterations(N) ← n_rules(M), N=(M+1)/2.
6  iteration(0..N) ← max_iterations(N).
7  defended(X,I) ← axiom(X), iteration(I).
8  defended(X,I) ← head(R,X), defended_rule(R,I).
9  defended_rule(R,I) ← iteration(I), usable_rule(R,I),
       rule(R), defended(X,I) : body(R,X).
10 usable_rule(R,I) ← iteration(J), rule(R),
       not defeated_by_undefeated(R,J), J+1=I.
11 defeated(R,I) ← head(R,X), axiom(Y), ctr(X,Y),
       iteration(I).
12 defeated(R,I) ← head(R,X), defended_rule(DR,I),
       head(DR,Y), ctr(X,Y), no_less_preferred(DR,R).
13 derived_from_undefeated(X,I) ← axiom(X),
       iteration(I).
14 derived_from_undefeated(X,I) ← head(R,X),
       undefeated(R,I).
15 undefeated(R,I) ← iteration(I),
       rule(R), not defeated(R,I),
       derived_from_undefeated(X,I) : body(R,X).
16 defeated_by_undefeated(R,I) ← head(R,X), axiom(Y),
       ctr(X,Y), iteration(I).
17 defeated_by_undefeated(R,I) ← head(R,X),
       undefeated(IR,I), head(IR,Y), ctr(X,Y),
       no_less_preferred(IR,R).
18 defended_rule(R) ← defended_rule(R,N),
       max_iterations(N).
19 defended(X) ← defended(X,N), max_iterations(N).
20 out(L) ← derivable(L), max_iterations(N),
       not derived_from_undefeated(L,N).
21 blocked(L) ← literal(L), not unsat(L), not out(L),
       not defended(L).
```

$\{\mathbf{axiom}(L)\} \leftarrow \mathbf{queryable}(L)$. Consistency is enforced via the constraint $\leftarrow \mathbf{axiom}(L), \mathbf{axiom}(N), \mathbf{ctr}(L, N)$. We refer to these rules as Δ_{stab}. The encoding for checking stability without preferences is $\pi_{stab} = \Delta_{stab} \cup \pi_{just}$, and for the case with preferences is $\pi_{<\text{-}stab} = \Delta_{stab} \cup \pi_{<\text{-}just}$. One can obtain the stability statuses of all literals via the cautious reasoning mode readily available in modern ASP solvers, directly computing the intersection of all answer sets to a given program: the literals that are **j** in the cautious solution are stable-j. If the stability status of a single literal is of interest, it can be decided with one ASP solver call. With the addition of the constraint $\leftarrow \mathbf{j}(l)$ for literal $l \in \mathcal{L}$ and justification status $j \in \{$*unsatisfiable*, *defended*, *out*, *blocked*$\}$, the resulting program does not have an answer set iff l is stable-j.

5.3 ASP-Based Algorithm for Relevance

For j-relevance, we detail ASP-based counterexample-guided abstraction refinement (CEGAR) (Clarke, Gupta, and Strichman 2004; Clarke et al. 2003) algorithms, using our stability encodings as subprocedures. In CEGAR, an NP-abstraction as an overapproximation of the solution space is iteratively refined by drawing candidates from this space and verifying if the candidate is an actual solution. Candidate solutions are computed with an ASP solver. If there is a candidate solution, another ASP solver call is made to check if there is a counterexample to the candidate being a solution. If there is no counterexample, the candidate is an actual solution. Otherwise the abstraction is refined by analyzing the counterexample and the search is continued.

We assume as input an $T = (AS, \mathcal{K})$, a queryable q, a literal l, and a justification status j. We present an algorithm that decides if q is j-relevant for l wrt T. The idea is to find a $\mathcal{K}'$ such that $q \in \mathcal{K}'$ and l is j-stable wrt $(AS, \mathcal{K}')$ but not wrt $(AS, \mathcal{K}' \setminus \{q\})$, in which case q is j-relevant for l by Lemma 1. We first show properties based on which we can narrow the search space upon finding counterexamples.

Proposition 9. *Let $T = (AS, \mathcal{K})$ be an AT, $\mathcal{Q}$ a set of queryables and j a justification status. Given $l \in \mathcal{L}$ and $q \in \mathcal{Q}$ where $q \notin \mathcal{K}$ and $\overline{q} \cap \mathcal{K} = \emptyset$,*

- *if $T' = (AS, \mathcal{K}') \sqsupseteq_Q T$ such that l is not stable-j wrt T', then for each $\mathcal{K}'' \subseteq \mathcal{K}'$, l is not stable-j wrt $(AS, \mathcal{K}'')$;*
- *if $T' = (AS, \mathcal{K}') \sqsupseteq_Q T$ such that l is stable-j wrt T' and $q \notin \mathcal{K}'$, then for each consistent $\mathcal{K}'' \supseteq \mathcal{K}'$, l is stable-j wrt $(AS, \mathcal{K}'' \setminus \{q\})$.*

Our approach is presented as Algorithm 1. A candidate is a consistent set of queryables $Q \subseteq \mathcal{Q}$ such that $q \in Q$ and $\mathcal{K} \subseteq Q$ (Line 3). Given Q (Line 4), it is checked if l is stable-j with respect to (AS, Q) (Line 5). If so, it is checked whether l is stable-j wrt $Q \setminus \{q\}$ (Line 8). If the first condition holds and the second does not, Q is a witnessing set of queryables for q being j-relevant for l (Line 10). In other cases, we refine the abstraction depending on which type of counterexample is found, based on Proposition 9. If the first check fails, all subsets of the counterexample obtained in the check are refined out (Line 6). In particular, the call produces a $Q' \supset Q$ such that l is not stable-j wrt (AS, Q') and no subset of Q' can be the set of queryables $\mathcal{K}'$ that would show that q is j-relevant for l. If the second condition holds, supersets of Q can be ruled out, because then $Q \setminus \{q\}$ is a counterexample for any superset of Q as well (Line 9). The refinements on Lines 6 and 9 are accomplished by

Algorithm 1: ASP-based CEGAR for relevance

Require: AT $T = (AS, \mathcal{K})$, $q \in \mathcal{Q}$, $l \in \mathcal{L}$ and $j \in \{defended, out, blocked, unsatisfiable\}$
Ensure: return YES if q is j-relevant for q, NO otherwise

1: $\pi_c := \pi_{candidate}$
2: $\pi_v := \pi_{(<\text{-})stab} \cup \{\leftarrow not\, \mathbf{j}(l)\}$
3: **while** $I := \pi_c \cup \{\leftarrow not\, \mathbf{axiom}(q)\}$ is satisfiable **do**
4: $\quad Q := \{q' \in \mathcal{Q} \mid \mathbf{axiom}(q') \in I\}$
5: $\quad$ **if** $C := \pi_v \cup \{\mathbf{axiom}(q') \mid q' \in Q\}$ is satisfiable
6: $\quad\quad$ **then** $\pi_c := \pi_c \cup no\_subsets(C)$
7: $\quad$ **else**
8: $\quad\quad$ **if** $\pi_v \cup \{\mathbf{axiom}(q') \mid q' \in Q \setminus \{q\}\}$ unsatisfiable
9: $\quad\quad\quad$ **then** $\pi_c := \pi_c \cup no\_supersets(Q)$
10: $\quad\quad$ **else return** YES
11: **return** NO

ASP constraints $\leftarrow not\ \mathbf{axiom}(q_1), ..., not\ \mathbf{axiom}(q_n)$ for $not\ \mathbf{axiom}(q_i) \in C$, and $\leftarrow \mathbf{axiom}(q_1), ..., \mathbf{axiom}(q_m)$ for $q_i \in Q$, respectively.

6 Empirical Evaluation

We empirically evaluate the ASP-based approaches to deciding stability and relevance, using Clingo (Gebser et al. 2016) as the ASP solver and its incremental (multi-shot) features for implementing the CEGAR algorithms for relevance. Our implementation is available in open source at https://bitbucket.org/coreo-group/raspic. Our approach provides the first algorithm for relevance in ASPIC$^+$ and for stability in ASPIC$^+$ with preferences, and the first exact algorithm for stability without preferences. For stability, we compare the ASP approach to a polynomial-time inexact algorithm (Odekerken et al. 2022a, Algorithm 4) as the key earlier approach proposed for the problem for instances without rule preferences. The inexact approach is sound (all stable results are indeed stable) but not complete (the algorithm may report unstability for stable literals). The experiments were run on 2.60-GHz Intel Xeon E5-2670 57-GB machines with RHEL 8 under a per-instance 10-min time and 16-GB memory limit.

As benchmarks, we consider both real-world and synthetic data. The *real-world benchmarks* are based on the argumentation system $AS = (\mathcal{L}, ^-, \mathcal{R}, \leq)$ and queryables $\mathcal{Q}$ used in an inquiry system for the intake of online trade fraud at the Netherlands Police (Odekerken et al. 2022a) with $|\mathcal{L}| = 60$, $|\mathcal{Q}| = 30$, $|\mathcal{R}| = 43$ and $\leq = \emptyset$. The rules form a tree-like structure, without cycles. To generate stability instances, we obtained knowledge bases by randomly sampling 25 subsets of each size between 1 and 15 from $\mathcal{Q}$ and the empty knowledge base, for a total of 376 instances. We computed the stability status of each literal. For relevance, we randomly selected one literal from a set of "topics" (literals whose status is of interest) and one queryable. For a further scalability study, we also consider *synthetic data*. For this, we generated ATs and queryable sets that are parametrised by the size of the language $|\mathcal{L}| \in \{50, 100, 150, 200, 250, 500, 1000, 2500\}$ and rule set size $|\mathcal{R}| \in \{\frac{1}{2}|\mathcal{L}|, |\mathcal{L}|, \frac{3}{2}|\mathcal{L}|\}$. These ATs have a similar tree-like structure as the real-world benchmarks with 1125 instances per $|\mathcal{L}|$.

Results Table 1 provides an overview of the performance of our ASP approach (with and without preferences) on the task of computing the stability of *each* literal in a given AT. As shown, without preferences our exact approach, taking less than a second on each instance, outperforms the inexact algorithm. Additionally, while the ASP approach is exact, the inexact algorithm mislabelled 69 out of 1689 topic literals in the real-world instances and 109 out of 714431 topic literal in the synthetic instances. For the case with preferences, deciding stability becomes empirically harder, with the largest instances taking a few minutes to solve. We observed that runtimes similarly increase with more rules.

Table 2 shows results for deciding relevance both with and without preferences. For the real-world data without preferences, our approach can decide relevance of a query in 0.26 seconds on average, with a maximum of 5.8 seconds. Our algorithm for relevance also scales well to reasonable-size instance, solving all instances with up to 100 literals. We observe a high variance in runtimes on large instances, with many instances solved instantly while some time out, suggesting that the structure of instances plays a significant role in runtime performance. Overall, the results suggest that our ASP approach is applicable in real-world applications.

		#solved (mean runtime (s) over solved)					
Data	$\lvert\mathcal{L}\rvert$	In-exact		ASP		ASP under prefs	
Synthetic	50	1125	(0.16)	1125	(0.01)	1125	(0.09)
	100	1125	(0.18)	1125	(0.02)	1125	(0.27)
	150	1125	(0.20)	1125	(0.03)	1125	(0.56)
	200	1125	(0.23)	1125	(0.05)	1125	(0.98)
	250	1125	(0.25)	1125	(0.06)	1125	(1.51)
	500	1125	(0.37)	1125	(0.12)	1125	(6.37)
	1000	1125	(0.66)	1125	(0.24)	1125	(28.65)
	2500	1125	(2.17)	1125	(0.6)	1047	(209.80)
Real	60	376	(0.16)	376	(0.02)	376	(0.09)

Table 1: Number of solved instances and mean runtimes over solved instances for detecting stability of all literals.

7 Conclusions

With motivations in real-world applications, we established the complexity of stability and relevance—two related notions dealing with argumentation dynamics—in a specific fragment of the structured argumentation formalism ASPIC$^+$. While stability was recently proposed in the realm of ASPIC$^+$, our work constitutes the first study of relevance in this context. Complementing and motivated by the NP-completeness and Σ_2^p-completeness results, we developed the first exact algorithms for stability and relevance based on the declarative programming paradigm of answer set programming. The algorithms exhibit promising scalability in practice, and allow for reasoning about stability and relevance efficiently in a real-world setting concerning argument-based inquiry. Extending the complexity analysis and algorithms to cover further semantics and ASPIC$^+$ more generally is a promising direction for further work.

		#solved (mean runtime (s) over solved)			
Data	$\lvert\mathcal{L}\rvert$	ASP no prefs		ASP under prefs	
Synthetic	50	1125	(0.2)	1125	(0.2)
	100	1125	(5.4)	1125	(19.4)
	150	831	(2.1)	809	(2.1)
	200	820	(1.2)	824	(1.5)
	250	844	(0.6)	831	(1.4)
	500	817	(0.2)	809	(5.9)
	1000	837	(0.3)	816	(24.0)
	2500	1125	(4.9)	898	(220.8)
Real	60	376	(0.26)	376	(0.36)

Table 2: Number of solved instances and mean runtimes over solved instances for defended-relevance.

Acknowledgments

This work has been financially supported in part by University of Helsinki Doctoral Programme in Computer Science DoCS, Austrian Science Fund (FWF) P35632, and Academy of Finland grants 322869 and 356046. The authors wish to thank the Finnish Computing Competence Infrastructure (FCCI) for supporting this project with computational and data storage resources.

References

Alfano, G.; Greco, S.; Parisi, F.; Simari, G. I.; and Simari, G. R. 2021. Incremental computation for structured argumentation over dynamic DeLP knowledge bases. *Artificial Intelligence* 300:103553.

Alfano, G.; Greco, S.; and Parisi, F. 2017. Efficient computation of extensions for dynamic abstract argumentation frameworks: An incremental approach. In Sierra, C., ed., *Proc. IJCAI*, 49–55. ijcai.org.

Atkinson, K.; Baroni, P.; Giacomin, M.; Hunter, A.; Prakken, H.; Reed, C.; Simari, G.; Thimm, M.; and Villata, S. 2017. Towards artificial argumentation. *AI magazine* 38(3):25–36.

Baroni, P.; Gabbay, D.; Giacomin, M.; and van der Torre, L., eds. 2018. *Handbook of Formal Argumentation*. College Publications.

Baroni, P.; Caminada, M.; and Giacomin, M. 2011. An introduction to argumentation semantics. *The Knowledge Engineering Review* 26(4):365–410.

Baumann, R., and Brewka, G. 2010. Expanding argumentation frameworks: Enforcing and monotonicity results. In Baroni, P.; Cerutti, F.; Giacomin, M.; and Simari, G. R., eds., *Proc. COMMA*, volume 216 of *Frontiers in Artificial Intelligence and Applications*, 75–86. IOS Press.

Baumeister, D.; Neugebauer, D.; Rothe, J.; and Schadrack, H. 2018. Verification in incomplete argumentation frameworks. *Artificial Intelligence* 264:1–26.

Baumeister, D.; Järvisalo, M.; Neugebauer, D.; Niskanen, A.; and Rothe, J. 2021. Acceptance in incomplete argumentation frameworks. *Artificial Intelligence* 295:103470.

Besnard, P.; Garcia, A.; Hunter, A.; Modgil, S.; Prakken, H.; Simari, G.; and Toni, F. 2014. Introduction to structured argumentation. *Argument & Computation* 5(1):1–4.

Borg, A., and Bex, F. 2021. Enforcing sets of formulas in structured argumentation. In Bienvenu, M.; Lakemeyer, G.; and Erdem, E., eds., *Proc. KR*, 130–140. IJCAI.

Caminada, M., and Amgoud, L. 2007. On the evaluation of argumentation formalisms. *Artificial Intelligence* 171(5-6):286–310.

Cayrol, C.; de Saint-Cyr, F. D.; and Lagasquie-Schiex, M.-C. 2010. Change in abstract argumentation frameworks: Adding an argument. *Journal of Artificial Intelligence Research* 38:49–84.

Cayrol, C.; Devred, C.; and Lagasquie-Schiex, M. 2007. Handling ignorance in argumentation: Semantics of partial argumentation frameworks. In Mellouli, K., ed., *Proc. ECSQARU*, volume 4724 of *Lecture Notes in Computer Science*, 259–270. Springer.

Clarke, E. M.; Grumberg, O.; Jha, S.; Lu, Y.; and Veith, H. 2003. Counterexample-guided abstraction refinement for symbolic model checking. *Journal of the ACM* 50(5):752–794.

Clarke, E. M.; Gupta, A.; and Strichman, O. 2004. SAT-based counterexample-guided abstraction refinement. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* 23(7):1113–1123.

Dung, P. M. 1995. On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games. *Artificial Intelligence* 77:321–357.

Gebser, M.; Kaminski, R.; Kaufmann, B.; Ostrowski, M.; Schaub, T.; and Wanko, P. 2016. Theory solving made easy with Clingo 5. In *Technical Communications of ICLP*, OASICS, 2:1–2:15. Schloss Dagstuhl - Leibniz-Zentrum für Informatik.

Gelfond, M., and Lifschitz, V. 1988. The stable model semantics for logic programming. In *Proc. ICLP/SLP*, 1070–1080. MIT Press.

Lehtonen, T.; Wallner, J. P.; and Järvisalo, M. 2020. An answer set programming approach to argumentative reasoning in the ASPIC+ framework. In Calvanese, D.; Erdem, E.; and Thielscher, M., eds., *Proc. KR*, 636–646. IJCAI.

Lehtonen, T.; Wallner, J. P.; and Järvisalo, M. 2022. Computing stable conclusions under the weakest-link principle in the ASPIC+ argumentation formalism. In Kern-Isberner, G.; Lakemeyer, G.; and Meyer, T., eds., *Proc. KR*, 215–225. IJCAI.

Maher, M. J. 2016. Resistance to corruption of strategic argumentation. In Schuurmans, D., and Wellman, M. P., eds., *Proc. AAAI*, 1030–1036. AAAI Press.

Mailly, J.-G., and Rossit, J. 2020. Stability in abstract argumentation. In Martinez, M. V., and Varzinczak, I., eds., *Proc. NMR*, 93–99.

Modgil, S., and Prakken, H. 2013. A general account of argumentation with preferences. *Artificial Intelligence* 195:361–397.

Niemelä, I. 1999. Logic programs with stable model semantics as a constraint programming paradigm. *Annals of Mathematics and Artificial Intelligence* 25(3-4):241–273.

Niskanen, A., and Järvisalo, M. 2020. Algorithms for dynamic argumentation frameworks: An incremental sat-based approach. In Giacomo, G. D.; Catalá, A.; Dilkina, B.; Milano, M.; Barro, S.; Bugarín, A.; and Lang, J., eds., *Proc. ECAI*, volume 325 of *Frontiers in Artificial Intelligence and Applications*, 849–856. IOS Press.

Odekerken, D.; Bex, F.; Borg, A.; and Testerink, B. 2022a. Approximating stability for applied argument-based inquiry. *Intelligent Systems with Applications* 16:200110.

Odekerken, D.; Bex, F.; Borg, A.; and Testerink, B. 2022b. Computing the justification status of literals in polynomial time. Technical appendix, avail-

able at https://www.uu.nl/sites/default/files/Odekerken_etal-JustificationLabelAlgorithm.pdf.

Odekerken, D.; Borg, A.; and Bex, F. 2020. Estimating stability for efficient argument-based inquiry. In Prakken, H.; Bistarelli, S.; Santini, F.; and Taticchi, C., eds., *Proc. COMMA*, volume 326 of *Frontiers in Artificial Intelligence and Applications*, 307–318. IOS Press.

Odekerken, D.; Borg, A.; and Bex, F. 2022. Stability and relevance in incomplete argumentation frameworks. In Toni, F.; Polberg, S.; Booth, R.; Caminada, M.; and Kido, H., eds., *Proc. COMMA*, volume 353 of *Frontiers in Artificial Intelligence and Applications*, 272–283. IOS Press.

Parsons, S.; Wooldridge, M. J.; and Amgoud, L. 2002. An analysis of formal inter-agent dialogues. In *Proc. AAMAS*, 394–401. ACM.

Prakken, H. 2010. An abstract framework for argumentation with structured arguments. *Argument & Computation* 1(2):93–124.

Rapberger, A., and Ulbricht, M. 2022. On dynamics in structured argumentation formalisms. In Kern-Isberner, G.; Lakemeyer, G.; and Meyer, T., eds., *Proc. KR*, 288–298. IJCAI.

Strass, H.; Wyner, A.; and Diller, M. 2019. *EMIL*: Extracting meaning from inconsistent language: Towards argumentation using a controlled natural language interface. *International Journal of Approximate Reasoning* 112:55–84.

Testerink, B.; Odekerken, D.; and Bex, F. 2019. A method for efficient argument-based inquiry. In Cuzzocrea, A.; Greco, S.; Larsen, H. L.; Saccà, D.; Andreasen, T.; and Christiansen, H., eds., *Proc. FQAS*, volume 11529 of *Lecture Notes in Computer Science*, 114–125. Springer.

Wallner, J. P.; Niskanen, A.; and Järvisalo, M. 2017. Complexity results and algorithms for extension enforcement in abstract argumentation. *Journal of Artificial Intelligence Research* 60:1–40.