Argumentative Reasoning in ASPIC$^+$ under Incomplete Information

DAPHNE ODEKERKEN*, Department of Information and Computing Sciences, Utrecht University, and National Police Lab AI, Netherlands Police, The Netherlands
TUOMO LEHTONEN, Department of Computer Science, University of Helsinki, and Department of Computer Science, Aalto University, Finland
JOHANNES P. WALLNER, Institute of Software Engineering and Artificial Intelligence, Graz University of Technology, Austria
MATTI JÄRVISALO, Department of Computer Science, University of Helsinki, Finland

Reasoning under incomplete information is an important research direction in the study of computational argumentation. Most advances in this direction so far have focused on abstract argumentation frameworks. In particular, development of computational approaches to reasoning under incomplete information in structured formalisms remains to a large extent a challenge. We address this challenge by studying the problems of determining stability and relevance—with the aim of analyzing aspects of resilience of acceptance statuses in light of new information—in the central structured formalism of ASPIC$^+$. The specific ASPIC$^+$ instantiation and grounded argumentation semantics we focus on are motivated by current applications in criminal investigation at the Netherlands Police. Our contributions consist of a theoretical analysis of the complexity of deciding stability and relevance as well as first exact algorithms for reasoning about stability and relevance in incomplete ASPIC$^+$ theories. In terms of complexity results, we show that deciding stability is coNP-complete for incomplete ASPIC$^+$ when assuming a preference ordering on defeasible rules via the last-link ordering, while deciding relevance is significantly more complex, namely Σ_2^{P}-complete. Complementing the complexity results, we develop practical algorithms for deciding stability and relevance based on the declarative paradigm of answer set programming (ASP). Furthermore, we provide an open-source implementation of the algorithms, and show empirically that the implementation exhibits promising scalability on both real-world and synthetic data. Our exact approach to stability is competitive with a previously proposed inexact approach, and the run times of our algorithms for both stability and relevance are sufficiently low on real-world data to be used in online settings.

JAIR Associate Editor: Laura Giordano

JAIR Reference Format:
Daphne Odekerken, Tuomo Lehtonen, Johannes P. Wallner, and Matti Järvisalo. 2025. Argumentative Reasoning in ASPIC$^+$ under Incomplete Information. *Journal of Artificial Intelligence Research* 83, Article 28 (August 2025), 52 pages. DOI: 10.1613/jair.1.18404

*Corresponding Author.

Authors' Contact Information: Daphne Odekerken, ORCID: 0000-0003-0285-0706, d.odekerken@uu.nl, Department of Information and Computing Sciences, Utrecht University, and National Police Lab AI, Netherlands Police, Utrecht, The Netherlands; Tuomo Lehtonen, ORCID: 0000-0001-6117-4854, tuomo.lehtonen@aalto.fi, Department of Computer Science, University of Helsinki, and Department of Computer Science, Aalto University, Helsinki, Finland; Johannes P. Wallner, ORCID: 0000-0002-3051-1966, johannes.p.wallner@tugraz.at, Institute of Software Engineering and Artificial Intelligence, Graz University of Technology, Graz, Austria; Matti Järvisalo, ORCID: 0000-0003-2572-063X, matti.jarvisalo@helsinki.fi, Department of Computer Science, University of Helsinki, Helsinki, Finland.

DOI: 10.1613/jair.1.18404

1 Introduction

The study of computational aspects of argumentation is a vibrant research direction in artificial intelligence and in particular in knowledge representation and reasoning [5, 7]. Argumentation is intrinsically a dynamic process: when arguing about a particular claim, the acquisition of new information can lead to new arguments related to the topic, which may lead to a change in whether the claim at hand is considered acceptable. Indeed, in recent years the study of different forms and notions of dynamics in formal models of argumentation has gained significant traction from both representational and computational points of view. This line of work encompasses aspects of dynamics in both abstract argumentation formalisms, see for example [16, 8, 20, 1, 54, 11, 36, 9, 3, 10], and structured formalisms, see for example [2, 52, 13, 47, 48, 45, 12].

1.1 Stability and Relevance in Structured Argumentation

We focus on *stability* and *relevance* in the context of structured argumentation as suitable notions for capturing various argumentative settings that require reasoning under incomplete information. In particular, when reasoning under incomplete information, it is sensible (if not necessary) to take into account not only the information that is currently present, but also relevant information that may be obtained in the future. This is particularly important in inquiry dialogues, where the goal is to find out whether a claim should be accepted regardless of additional future information [55, 43]. The participants of an inquiry dialogue can be human, artificial, or a combination of humans and machines, typically assisting a human user in the inquiry process. A specific example of such a system is an inquiry agent that supports the investigation of online trade fraud in the law enforcement domain [38]. This system is equipped with a set of rules that describe online trade fraud. The rules are defeasible in the sense that inference is not absolutely certain. For instance, a simplified version of one of these rules is: "If a citizen paid for a product but did not receive anything after waiting for a sufficiently long time and was not dealing with a trusted web shop, then *usually* they are victim of online trade fraud." In addition, the system is aware of which information can generally be obtained from a citizen, such as "Did you pay?" or "Did you receive the item you ordered?". When a new complaint of potential online trade fraud is submitted by a citizen, the system uses this domain knowledge in combination with knowledge extracted from the complaint to deduce whether a fraud has likely happened. For instance, suppose that the citizen states that they have not received an item they ordered from a web shop even after waiting for three months for the item to arrive. Then no argument for fraud can be constructed due to missing information: it is not known whether the citizen paid for the item. Given current information, the claim that this citizen is a victim of fraud should not be accepted. However, more information about the payment could change this. Therefore, the system will ask the citizen whether they paid. If the citizen answers that they did pay, then they might be a victim of fraud. The system will hence proceed by asking additional questions about the web shop. In case the payment was not made, the system can conclude that the citizen is certainly not a victim of fraud. Additional information cannot change this conclusion.

Apart from the law enforcement setting just described, systems for argument-based inquiry can in general be useful for supporting individuals in complex decision-making. This is especially true when decisions are made based on rule-based reasoning in settings where information is not complete and acquiring additional information comes at a cost [49, 40]. In such inquiry settings it is important that a system can recognize whether additional information can change the acceptance of the claim at hand. In addition, such a system must be able to determine which thus-far unknown information may need to be investigated in order to reach a point where additional information cannot change the acceptance of the claim. These practical requirements correspond to the formal notions of *stability* and *relevance*. The two notions provide perspectives to reasoning about acceptance of conclusions under incomplete information. Specifically, stability refers to the impossibility of the acceptance status of a conclusion to change with new information. Stability provides a key point of view 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. On the other hand, relevance provides a point of view to reasoning about not (yet) stable conclusions. The choice of which additional information is gathered can play an important role in the efficiency of an inquiry application: only information that can change the stability status of a conclusion is relevant for determining which yet unknown information should be investigated to ensure stability of a conclusion. The notion of stability was originally defined for ASPIC⁺ [52, 40, 38] and subsequently studied in the realm of abstract argumentation [32, 41]. In contrast, relevance has to date only been studied in the representationally less complex abstract setting [41, 37]. However, the aforementioned fraud inquiry system, as well as envisioned future argument-based inquiry systems, should be able to reason with defeasible rules as these provide a natural way to model practical and legal rules concerning crime [40, 38].

1.2 Focus of this Work

We study stability and relevance from a computational perspective in the context of the central structured argumentation formalism of ASPIC⁺ [33] under incomplete information. ASPIC⁺ is a very general formalism, meant to capture diverse settings. To this end we provide definitions for stability and relevance over *incomplete* ASPIC⁺ theories. In analogy with incomplete abstract argumentation frameworks (IAFs) [15, 11, 21, 10, 4], incomplete ASPIC⁺ theories allow for modeling a set of possible "future theories", thereby also enabling argumentative reasoning under incomplete information. The specific ASPIC⁺ instantiation we focus on is motivated by applications in criminal investigation, as an extension of the aforementioned instantiation used for inquiry dialogue at the Netherlands Police [38]. In such inquiry dialogues it is essential that reasoning is based on observations that are considered certain, corresponding to axioms and unattackable premises in the knowledge base. In terms of types of rules, we consider defeasible rules. In particular, 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 rationality postulates [14] are satisfied. Finally, all notions of conflict in the police use-case [38] can be modelled using rebuttal attacks, that is, attacks on conclusions of defeasible rules. It should be noted that, generalizing the ASPIC⁺ instantiation used by Odekerken et al. [38], we also allow preferences on rules. We focus on grounded semantics, 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 characterized. Towards extending our analysis to more general variants of ASPIC⁺, we additionally show some general complexity results, applying to other argumentation semantics and the inclusion of strict rules and ordinary premises.

1.3 Contributions

Our main contributions consist of theoretical analysis of the complexity of deciding stability and relevance in ASPIC⁺ under incomplete information, as well as first exact algorithms for reasoning about stability and relevance in ASPIC⁺.

In terms of complexity results, we pinpoint the computational complexity of deciding stability and relevance in the considered variant of ASPIC⁺. Whereas deciding stability in incomplete ASPIC⁺ has been shown to be coNP-complete in earlier work [38], this earlier result was established for a restricted ASPIC⁺ version in which all defeasible rules are equally preferred. Extending the analysis, we show that deciding stability is coNP-complete for incomplete ASPIC⁺ also when assuming the last-link preference ordering on defeasible rules. Along with our coNP-membership proof, we provide a polynomial-time algorithm for the *justification* problem of deciding the status of a given claim in the "complete" ASPIC⁺ setting. Moving from stability to the intuitively harder problem of relevance, we establish that deciding relevance is indeed significantly more complex, namely Σ_2^{P}-complete. We also identify some complexity upper and lower bounds of deciding stability and relevance for more general fragments of ASPIC⁺ and semantics other than grounded.

Complementing the theoretical analysis, we develop the first exact algorithms for deciding stability and relevance in incomplete ASPIC$^+$. Despite the need for practical algorithms in applications, the only practical algorithmic approach available for stability in ASPIC$^+$ has been an approximative approach [38]. Although the approximative approach is sound in the sense that all conclusions it identifies a stable are indeed stable, it is not complete as it may incorrectly classify stable conclusions as not being stable. Furthermore, to the best of our knowledge, no practical algorithmic approaches to reasoning about relevance have been proposed to date. Our algorithms are based on the declarative paradigm of answer set programming (ASP) [26, 35], motivated by recent successful ASP-based approaches to reasoning about acceptance in structured argumentation formalisms [28, 30, 31, 29]. Going beyond deciding relevance of a single piece of information, we also extend our algorithm for deciding relevance to allow for finding all relevant information within an ASPIC$^+$ theory. This is useful for applications in inquiry, as the relevant information within an ASPIC$^+$ theory intuitively maps to all relevant questions that one could ask a user. We provide an open-source implementation of the algorithms and present an extensive empirical evaluation of our implementation on both real-world and synthetic data. The empirical results show promising scalability: our exact approach to stability is competitive with the previously-proposed inexact approach in terms of run times, and the run times of our algorithms for both stability and relevance are sufficiently low on real-world data to enable their use in online settings, and also scale well on synthetic instances.

As an additional contribution, we relate the notions of stability and relevance in incomplete ASPIC$^+$ to stability and relevance studied earlier in the context of incomplete abstract argumentation frameworks. In particular, we show that—although the notions are related—stability and relevance in incomplete ASPIC$^+$ cannot be directly reduced to stability and relevance in IAFs, highlighting the need to study stability and relevance in ASPIC$^+$.

This article noticeably extends and revises a preliminary version of this work presented at the KR 2023 conference [42]. New contributions include general complexity results for incomplete ASPIC$^+$ (Section 4.5); an extension of the algorithmic approach to allow for computing all relevant pieces of information (Section 5.4); a revised implementation of our algorithms together with a more extensive empirical evaluation with results for the problem of computing all relevant information as well as for computing relevance under several different justification statuses (Section 6); and finally a detailed analysis of the relation between stability and relevance in incomplete ASPIC$^+$ to the analogous concepts in incomplete abstract argumentation frameworks (Section 7) which further motivates studying stability and relevance natively on the structured level. Furthermore, the presentation has been revised throughout and made self-contained. In particular, all complexity results are accompanied with full formal proofs, and all ASP encodings employed in the presented algorithms are described in full detail.

1.4 Outline

The rest of this article is organized as follows. After providing the necessary background on ASPIC$^+$ (Section 2), we introduce the notions of stability and relevance for the incomplete version of ASPIC$^+$ considered in this work (Section 3). We then detail complexity results for deciding stability and relevance (Section 4). Subsequently, we describe our ASP-based algorithmic approach to deciding stability and relevance (Section 5), and overview empirical results on the scalability of an implementation of the algorithms (Section 6). Finally, before concluding, we compare in detail the notions of stability and relevance in ASPIC$^+$ to the corresponding problems in incomplete abstract argumentation frameworks as previously proposed, and show that the latter do not capture the former (Section 7).

2 ASPIC$^+$

We recall ASPIC$^+$ to the extent relevant for our work, following the definitions by Modgil and Prakken [33]. A basic notion in ASPIC$^+$ is that of an argumentation system.

Definition 1 (Argumentation system). *An argumentation system is a quadruple $AS = (\mathcal{L}, \overline{\ }, \mathcal{R}, \leq)$, where*

- *$\mathcal{L}$ is a finite set of literals,*
- *$\overline{\ }$ is a contradiction function from $\mathcal{L}$ to $2^{\mathcal{L}}$ such that*
  - *for each ϕ, ψ: if $\phi \in \overline{\psi}$ then $\psi \in \overline{\phi}$ and*
  - *there is no $\phi \in \mathcal{L}$ such that $\phi \in \overline{\phi}$,*
- *$\mathcal{R}$ is 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*
- *$\leq$ is a partial preorder (i.e., a reflexive and transitive binary relation) on $\mathcal{R}$.*

While the requirement that literal and rule sets are finite is not present in the ASPIC⁺ definitions by Modgil and Prakken [33], this requirement is not restrictive for applications in, for example, criminal investigation, where it is natural that only a limited number of rules and literals are used for capturing domain-specific information.

We say that l is a *contradictory* of m if and only if $m \in \overline{l}$ and $l \in \overline{m}$. In examples we will use classical negation as the 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.

An argumentation system gives rise to arguments with respect to a knowledge base. We assume that knowledge bases are consistent sets of axioms.

Definition 2 (Knowledge base). *A knowledge base $\mathcal{K} \subseteq \mathcal{L}$ over an argumentation system $AS = (\mathcal{L}, \overline{\ }, \mathcal{R}, \leq)$ is a consistent set of literals, that is, 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 of an AT T is inductively defined as follows.*

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

The set Arg_T is the smallest set containing these arguments.

We use several functions and shorthands for components of arguments in Arg_T for a given AT T. For an observation-based argument c,

- the set of premises is $\texttt{prem}(c) = \{c\}$
- the set of defeasible rules is $\texttt{defrules}(c) = \emptyset$,
- the conclusion is $\texttt{conc}(c) = c$,
- the set of subarguments is $\texttt{sub}(c) = \{c\}$, and
- the top rule, denoted by $\texttt{top-rule}(A)$, is undefined.

For a rule-based argument $A = (A_1, \ldots, A_m \Rightarrow c) \in Arg_T$ with defeasible rule $r = \texttt{conc}(A_1), \ldots, \texttt{conc}(A_m) \Rightarrow c$,

- $\texttt{prem}(A) = \texttt{prem}(A_1) \cup \cdots \cup \texttt{prem}(A_m)$,
- $\texttt{defrules}(A) = \{r\} \cup \texttt{defrules}(A_1) \cup \cdots \cup \texttt{defrules}(A_m)$,
- $\texttt{conc}(A) = c$,
- $\texttt{sub}(A) = \texttt{sub}(A_1) \cup \cdots \cup \texttt{sub}(A_m) \cup \{A\}$, and
- $\texttt{top-rule}(A) = r$.

We refer to an argument with conclusion c as "an argument for c" and an argument A with $\texttt{defrules}(A) \subseteq R \subseteq \mathcal{R}$ as "an argument based on R". Given $R \subseteq \mathcal{R}$, we define $Arg_T(R) = \{A \in Arg_T \mid \texttt{defrules}(A) \subseteq R\}$, that is, $Arg_T(R)$ is the set of arguments that are based on R.

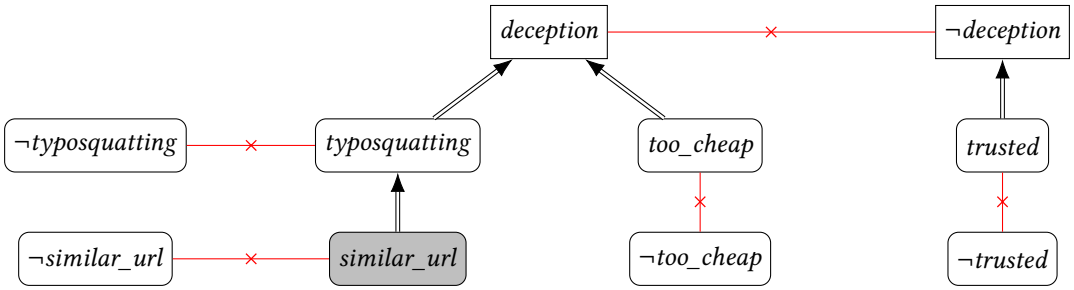

Fig. 1. AT $T = (AS, \mathcal{K})$ from Example 1. Each square is a literal in $\mathcal{L}$. Rounded squares are so-called queryable literals (defined in Section 3). Literals in $\mathcal{K}$ are shaded, double-lined arrows are defeasible rules and red single lines with a cross correspond to contradictory literals.

Example 1. *Consider the argumentation theory visualized in Figure 1, an excerpt from the domain of online trade fraud. Define* $T = (AS, \mathcal{K})$ *over* $AS = (\mathcal{L}, \bar{}, \mathcal{R}, \leq)$ *with* $\mathcal{L}$ *consisting of the following literals and their negations.*

- *deception: the web shop used some deceptive tricks.*
- *typosquatting: the url is a misspelling of a popular web shop (rather than an original company name).*
- *similar_url: the url is similar to the url of a popular web shop.*
- *too_cheap: the prices displayed are unrealistically low.*
- *trusted: the url is registered at a trustmark company.*

The set $\mathcal{R}$ *consists of the following defeasible rules:* $(similar\_url \Rightarrow typosquatting)$, $(typosquatting \Rightarrow deception)$, $(too\_cheap \Rightarrow deception)$, *and* $(trusted \Rightarrow \neg deception)$. *For now, assume that rules are equally preferred, that is,* $\leq = \emptyset$. *Suppose then that we are faced with a url that is similar to that of a popular web shop:* $\mathcal{K} = \{similar\_url\}$. *Then the set of arguments* Arg_T *consists of the observation-based argument similar_url and the rule-based arguments* $similar\_url \Rightarrow typosquatting$ *and* $[similar\_url \Rightarrow typosquatting] \Rightarrow deception$.

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′) if* $\mathsf{conc}(A) \in \overline{\mathsf{conc}(B')}$ *for some rule-based argument* $B' \in \mathsf{sub}(B)$.

Example 2. *Continuing Example 1, none of the arguments in* Arg_T *attack any other argument in* Arg_T. *However, in the AT* $T' = (AS, \mathcal{K} \cup \{\neg typosquatting\})$, *there is an argument for* $\neg typosquatting$ *that attacks the argument for deception (on the argument for typosquatting). Being observation-based, this argument for* $\neg typosquatting$ *is not attacked by any argument in* $Arg_{T'}$. *As an alternative example, the AT* $T'' = (AS, \mathcal{K} \cup \{trusted\})$ *contains two additional arguments compared to* Arg_T: *trusted and* $trusted \Rightarrow \neg deception$. *The argument for* $\neg deception$ *attacks the argument for deception and vice versa.*

Not all rebuttals succeed as defeats. In ASPIC$^+$ this depends on a preference relation, denoted by $\preceq$, between arguments. As usual, the strict variant is defined by $A \prec B$ if and only if $A \preceq B$ and $B \not\preceq A$.

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

In other words, an argument A might rebut B but not succeed (because $A \prec B'$). The idea is to obtain $\prec$ from the given partial order $\leq$ on defeasible rules $\mathcal{R}$. The "lifting" of $\leq$ to $\prec$ is defined using two steps. First the ordering on defeasible rules is lifted to sets of defeasible rules and, subsequently, this order on sets is lifted to $\prec$, comparing arguments. Modgil and Prakken [33] define four orderings, based on combinations of the so-called elitist and

democratic comparisons on sets (of defeasible rules) and the weakest and last-link principles. Here we focus on the last-link ordering, which considers the top-most defeasible rule(s). We assume that there is at most one defeasible top rule per argument (none for observation-based arguments). Democratic and elitist set comparisons coincide when the sets do not have multiple elements, and thus we have a singular ordering.

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

In words, an observation-based argument cannot be strictly less preferred to another argument. For rule-based arguments, the top rules of the arguments are compared.

Example 3. *Suppose that the rule for ¬deception given that the website is trusted is considered stronger than the rule for deception given typosquatting. Here we slightly adapt the AT $T'' = (AS, \mathcal{K} \cup \{trusted\})$ with argumentation system $AS = (\mathcal{L}, \bar{\ }, \mathcal{R}, \leq)$ from Example 2. Let $T^* = (AS^*, \mathcal{K} \cup \{trusted\})$, $AS^* = (\mathcal{L}, \bar{\ }, \mathcal{R}, \leq^*)$ and $\leq^* = \{((typosquatting \Rightarrow deception), (trusted \Rightarrow \neg deception))\}$. This AT contains an argument for deception ($A = [similar\_url \Rightarrow typosquatting] \Rightarrow deception$) and an argument for ¬deception ($B = trusted \Rightarrow \neg deception$). Since A and B are rule-based and* $\texttt{top-rule}(A) \leq \texttt{top-rule}(B)$, *it holds that $A \preceq B$. As* $\texttt{top-rule}(B) \not\leq \texttt{top-rule}(A)$, *we have $B \not\preceq A$. Hence argument B for ¬deception is strictly stronger than argument A for deception.*

Each argumentation theory gives rise to an abstract argument framework (AF) [19]. Semantics for argumentation theories are defined through semantics for AFs. Baroni et al. [6] give an in-depth introduction to semantics on AFs.

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

We focus on grounded semantics, motivated by practical applications (as outlined in the introduction): in many situations, such as criminal investigation, it is convenient to adopt a single-status semantics with a strong sceptical flavour, namely the grounded semantics.

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

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

Every AF F has exactly one grounded extension.

Example 4. *In the AT $T = (AS, \mathcal{K})$ of Example 1, all arguments in Arg_T are undefeated and therefore part of the grounded extension. Adding ¬typosquatting to the knowledge base results in the AT $T' = (AS, \mathcal{K} \cup \{\neg typosquatting\})$ in which the arguments for similar_url and ¬typosquatting are undefeated and therefore in the grounded extension. The arguments for typosquatting and deception are defeated by an argument in the grounded extension. As for $T'' = (AS, \mathcal{K} \cup \{trusted\})$, neither the argument for deception, nor the argument for ¬deception is in the grounded extension.*

In ASPIC⁺, a statement $x \in \mathcal{L}$ is justified under grounded semantics in an AT T if and only if there is an argument with conclusion x in the grounded extension $G(T)$ [33, Definition 15]. Applications may require a more fine-grained distinction between different types of justifications [38]. 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 $\langle \mathcal{A}, C \rangle$ be the AF defined by T. The justification status of $l \in \mathcal{L}$ in T is*

- *unsatisfiable if there is no argument for l in $\mathcal{A}$,*
- *defended if there is an argument for l in $\mathcal{A}$ that is in the grounded extension $G(T)$,*
- *out if there is an argument for l in $\mathcal{A}$ and each argument for l in $\mathcal{A}$ is defeated by an argument in $G(T)$, and*
- *blocked if 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 defined previously [33, Definition 15]. Conclusions of arguments that are not *defended* (i.e., whose justification status is not *defended*) can be either *out* or *blocked*. For 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. The justification statuses are mutually exclusive and complementary [38].

Example 5 (Justification statuses). *For the argumentation system AS from Example 1, we have that*

- *$\neg$deception is unsatisfiable w.r.t. $T = (AS, \{similar\_url\})$,*
- *deception is defended w.r.t. $T = (AS, \{similar\_url\})$,*
- *deception is out w.r.t. $T' = (AS, \{similar\_url, \neg typosquatting\})$, and*
- *deception is blocked w.r.t. $T'' = (AS, \{similar\_url, trusted\})$.*

3 Stability and Relevance

We turn to the main focus of this work: stability and relevance. Stability can be seen as considering a dynamic view on justification statuses. A justification status dictates 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 the justification status of l. If additional information cannot influence the justification status of l, we say that l is *stable*.

In this work we focus on modifications of the current information represented by an expansion of the knowledge base $\mathcal{K}$ through adding axioms. We impose restrictions on the allowed additions on the knowledge base by distinguishing between queryable literals—ones that can be added—and non-queryable literals.

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 (i) $\mathcal{K} \subseteq \mathcal{Q} \subseteq \mathcal{L}$ and (ii) if $q \in \mathcal{Q}$, then $\bar{q} \subseteq \mathcal{Q}$.*

All axioms are queryables by definition. 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 $\bar{q} \cap \mathcal{K} = \emptyset$, results in the AT $T' = (AS, \mathcal{K} \cup \{q\})$. The set of argumentation theories that can be obtained by adding queryables to the knowledge base of an AT 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 a 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, the axioms in their knowledge bases must be consistent.

We distinguish between 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 w.r.t. $\mathcal{Q}$ if l is j in T' for each T' with $T \sqsubseteq_Q T'$.*

Example 6 (Stability statuses). *Consider the argumentation system AS from Example 1. The rounded squares in Figure 1 represent queryables:* $\mathcal{Q} = \{typosquatting, similar\_url, too\_cheap, trusted, \neg typosquatting, \neg similar\_url, \neg too\_cheap, \neg trusted\}$. *We make the following observations.*

- *Let* $T_1 = (AS, \{similar\_url, \neg trusted\})$. *There is no future AT of* T_1 *having trusted in its knowledge base, as* $\neg trusted$ *is already in the knowledge base of* T_1 *and knowledge bases must be consistent. Thus* $\neg deception$ *is stable-unsatisfiable w.r.t.* T_1 *and* $\mathcal{Q}$.
- *Let* $T_2 = (AS, \{similar\_url, typosquatting, \neg trusted\})$. *There are arguments for deception in* T_2, *having subarguments for similar_url and typosquatting. These arguments are also in every future AT of* T_2. *Furthermore, there is no future AT containing an argument defeating any of these arguments due to the presence of* $\neg trusted$ *and typosquatting in the knowledge base. Hence every future AT of* T_2 *contains at least one undefeated argument for deception which must be in the grounded extension. This implies that deception is stable-defended w.r.t.* T_2 *and* $\mathcal{Q}$.
- *Let* $T_3 = (AS, \{similar\_url, \neg too\_cheap, \neg typosquatting\})$. *There is an argument for deception in* T_3. *The argument is defeated by the observation-based argument for* $\neg typosquatting$ *which is in the grounded extension of* T_3 *(and every future AT). In addition, there is no alternative argument for deception in any future AT of* T_3. *Consequently, deception must be stable-out w.r.t.* T_3 *and* $\mathcal{Q}$.
- *Let* $T_4 = (AS, \{similar\_url, typosquatting, trusted\})$. *Just like in* T_2, *there are arguments for deception in* T_4 *having subarguments for similar_url and typosquatting. These arguments are also in every future AT of* T_4. *Due to the presence of typosquatting in the knowledge base of* T_4, *there is no future AT containing an observation-based argument for* $\neg typosquatting$. *As a result, the arguments for deception cannot be defeated by an argument in the grounded extension. However, they are defeated by the argument for* $\neg$*defeated with the subargument for trusted in every future AT of* T_4. *To conclude, deception is stable-blocked w.r.t.* T_4.

When a literal does not have a stable status, that is, 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, that is, which queryables should be added to the knowledge base in order to obtain an AT where this literal is stable. This question is captured via the (novel) notion of relevance in ASPIC$^+$ that we define based on the notion of minimal stable future ATs. Minimal stable future ATs are future ATs in which 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 a literal* $l \in \mathcal{L}$, *a minimal stable-*j *future theory for* l *w.r.t.* T *and* $\mathcal{Q}$ *is an AT* T' *with* $T \sqsubseteq_{\mathcal{Q}} T'$ *such that*

(1) l *is stable-*j *in* T', *and*
(2) *there is no* T'' *such that* l *is stable-*j *in* T'' *and* $T \sqsubseteq_{\mathcal{Q}} T'' \sqsubset_{\mathcal{Q}} T'$.

Example 7 (Minimal stable-j future theory). *Consider again the AT* $T = (AS, \mathcal{K})$ *from Example 1 with* $\mathcal{Q}$ *and* $\mathcal{K} = \{similar\_url\}$ *as specified in Example 6. The AT* $T = (AS, \mathcal{K})$ *and some of its future ATs are illustrated in Figure 2. We find that* $\neg deception$ *is stable-unsatisfiable w.r.t.* $T' = (AS, \{similar\_url, too\_cheap, \neg trusted\})$ *and* $\mathcal{Q}$. *The AT* T' *is not a minimal stable-unsatisfiable future theory for* $\neg deception$ *w.r.t.* T *and* $\mathcal{Q}$, *since* $\neg deception$ *would also be stable-unsatisfiable without too_cheap. In contrast, it holds that the future AT* $(AS, \{similar\_url, \neg trusted\})$ *is minimal stable-unsatisfiable for* $\neg deception$. *This is because removing* $\neg trusted$ *from the knowledge base would result in an AT where* $\neg deception$ *is not stable-unsatisfiable.*

There are two minimal stable-defended future theories for deception w.r.t. T *and* $\mathcal{Q}$: $(AS, \{similar\_url, typosquatting, \neg trusted\})$ *and* $(AS, \{similar\_url, too\_cheap, \neg trusted\})$. *The future AT* $(AS, \{similar\_url, \neg trusted\})$ *is not minimal stable-defended for deception because deception is not stable-defended w.r.t. this AT.*

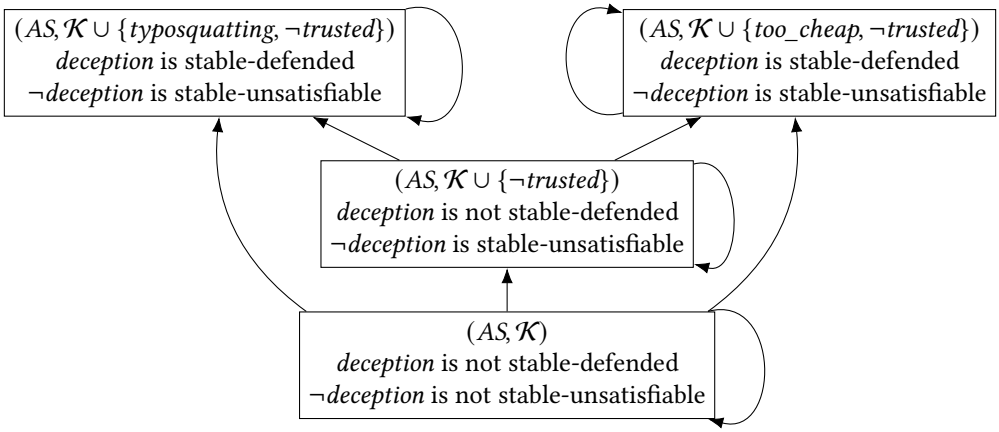

Fig. 2. Illustration of the future argumentation theories of T from Example 1 with Q as specified in Example 6. Argumentation theories are depicted as rectangular nodes. If there is an arrow from the node corresponding to an AT T_i to the node corresponding to an AT T_j, then $T_i \sqsubseteq_Q T_j$. Note that not all future ATs w.r.t. T and Q are illustrated in this figure.

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 $AS = (\mathcal{L}, \overline{\ }, \mathcal{R}, \leq)$, Q a set of queryables, and j 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 w.r.t. T and Q if there is a minimal stable-j future theory $T' = (AS, \mathcal{K}')$ for l w.r.t. T and Q such that $q \in \mathcal{K}'$.*

Example 8 (j-relevance). *Continuing Example 7 illustrated in Figure 2, for the AT $T = (AS, \mathcal{K})$ and queryables Q there is a single minimal stable-unsatisfiable future theory for $\neg$deception w.r.t. T and Q. The only queryable in the knowledge base of this theory that is not in $\mathcal{K}$ is $\neg$trusted. Consequently $\neg$trusted is the only literal that is unsatisfiable-relevant for $\neg$deception w.r.t. T and Q. For deception, there are two minimal stable-defended future theories w.r.t. T and Q. Their knowledge bases contain (combinations of) the queryables too_cheap, typosquatting and $\neg$trusted. Hence the defended-relevant literals for deception w.r.t. T and Q are too_cheap, typosquatting and $\neg$trusted.*

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

Example 9. *Consider the AT $T = (AS, \mathcal{K})$ where $AS = (\mathcal{L}, \overline{\ }, \mathcal{R}, \leq)$, $\mathcal{L} = \{q, \neg q, l, \neg l\}$, the contradiction function corresponds to classical negation, $\mathcal{R} = \{(q \Rightarrow l), (\neg q \Rightarrow l)\}$, $\leq = \emptyset$ and $\mathcal{K} = \emptyset$. Suppose that $Q = \{q, \neg q\}$. Then both q and $\neg q$ are defended-relevant for l w.r.t. T, since l is unsatisfiable in T and defended in both $(AS, \{q\})$ and $(AS, \{\neg q\})$.*

4 Complexity Results

In this section we present the main complexity-theoretic contributions of this work. Specifically, we pinpoint for the considered ASPIC$^+$ fragment the complexity of deciding for a given literal

(1) the justification status of the literal,
(2) the stability status of the literal, and
(3) whether a given queryable is relevant for the literal

for each of the four different justification statuses. It is important to note that complexity results previously established for the problems of stability and relevance on the abstract level do not directly translate to determining the complexity of the problems for ASPIC⁺ because, as we will explicate in Section 7, the notions of stability and relevance in abstract argumentation do not capture the notions in the structured setting.

We begin by rephrasing the grounded semantics in terms of sets of rules rather than sets of arguments in order to avoid explicitly considering the potentially exponential number of arguments that a given AT gives rise to. The rephrasing forms the basis for both the complexity results we show in this section, and the declarative algorithms we develop later on in this article. In terms of our algorithms, the rephrasing allows for avoiding the explicit (and unnecessary) construction of exponentially many arguments.

After rephrasing the grounded semantics in Section 4.1, we show in Section 4.2 that the justification status of a literal is decidable in polynomial time. We then show in Section 4.3 that the stability problems for the four justification statuses are coNP-complete, and furthermore establish Σ_2^{P}-completeness for deciding relevance in Section 4.4. Finally, in Section 4.5 we identify general complexity results beyond the specific ASPIC⁺ fragment considered in this work and the grounded semantics.

4.1 Rephrasing Grounded Semantics

As the first step, we prove P-membership for the justification problem. Note that this is not immediately clear from the definitions. In particular, Definition 9 specifies the grounded extension in terms of arguments, while the number of arguments an AT gives rise to is not in general polynomially bounded in the size of the AT. An example that exhibits an exponential number of arguments is provided by Strass et al. [51]. To establish polynomial-time decidability, we reformulate the grounded extension in terms of sets of *rules* rather than sets of *arguments*. Towards this, we introduce the notions of applicability, defeats and defence with respect to sets of rules.

Firstly, a rule r is applicable by D if there is an argument that is based on $D \cup \{r\}$ and uses r.

Definition 16 (Applicable by rule set). *Given an AT $T = (AS, \mathcal{K})$, and a set of defeasible rules $D \subseteq \mathcal{R}$, a rule $r \in \mathcal{R}$ is applicable by D if there is an argument A based on $D \cup \{r\}$ with $r \in \mathtt{defrules}(A)$.*

Example 10. *Recall the AT $T = (AS, \mathcal{K})$ in the web shop domain from Example 1 with the argumentation system $AS = (\mathcal{L}, \bar{}, \mathcal{R}, \leq)$. The set $\mathcal{R}$ contains four defeasible rules: (similar_url $\Rightarrow$ typosquatting), (typosquatting $\Rightarrow$ deception), (too_cheap $\Rightarrow$ deception) and (trusted $\Rightarrow$ $\neg$deception). Given that $\mathcal{K}$ contains only similar_url, $Arg_T(\mathcal{R})$ consists of three arguments (for similar_url, for typosquatting and for deception). Hence only the rules similar_url $\Rightarrow$ typosquatting and typosquatting $\Rightarrow$ deception are applicable by $\mathcal{R}$. Consider $D = \{$(similar_url $\Rightarrow$ typosquatting)$\} \subseteq \mathcal{R}$. Then $Arg_T(D)$ consists of the arguments for similar_url and for typosquatting. The rules applicable by D are similar_url $\Rightarrow$ typosquatting and typosquatting $\Rightarrow$ deception. For $D' = \{$(typosquatting $\Rightarrow$ deception), (too_cheap $\Rightarrow$ deception)$\} \subseteq \mathcal{R}$, $Arg_T(D')$ consists of the observation-based argument for similar_url. The only rule that is applicable by D' is similar_url $\Rightarrow$ typosquatting.*

Turning to the general case with preferences, we define defeat in terms of rule sets (in analogy with 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 if*

- *there is an l in $\mathcal{K}$ such that $l \in \overline{\mathsf{cons}(r)}$, or*
- *there is an r' in D such that $\mathsf{cons}(r') \in \overline{\mathsf{cons}(r)}$, r' is applicable by D, and $r' \not\prec r$.*

In words, a set of defeasible rules D defeats a single rule r when (i) the knowledge base contains an axiom contradictory to the consequent of r, or (ii) there is a rule $r' \in D$ applicable by D that has as consequent a

contradictory of the consequent of r and that is not strictly less preferred to r. Intuitively, in the former case, an observation-based argument directly defeats any argument with r as its top rule. In the latter 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.

Example 11. *Continuing Example 4, consider again the AT $T'' = (AS, \mathcal{K} \cup \{trusted\})$. Then $D = \{(similar\_url \Rightarrow typosquatting), (typosquatting \Rightarrow deception)\}$ defeats the rule $trusted \Rightarrow \neg deception$, since the consequence of one of the applicable rules in D and the latter rule are contradictory to each other. The set $D' = \{(trusted \Rightarrow \neg deception)\}$ defeats $typosquatting \Rightarrow deception$. On the other hand, if $typosquatting \Rightarrow deception$ is strictly preferred to $trusted \Rightarrow \neg deception$, then D' does not defeat $typosquatting \Rightarrow deception$.*

The preceding example suggests a correspondence between defeats by rule sets and defeats by arguments: the rule set $D = \{(similar\_url \Rightarrow typosquatting), (typosquatting \Rightarrow deception)\}$ defeats the rule $trusted \Rightarrow \neg deception$ and, correspondingly, the argument $[similar\_url \Rightarrow typosquatting] \Rightarrow deception$ in $Arg_T(D)$ defeats the argument $trusted \Rightarrow \neg deception$ which has the rule $trusted \Rightarrow \neg deception$ in its defeasible rules. We now show that this correspondence holds in general.

Proposition 1. *Given an AT $T = (AS, \mathcal{K})$, the corresponding AF $\langle \mathcal{A}, \mathcal{C} \rangle$, 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 $r \in \mathtt{defrules}(A)$.*

Proof. We show both directions of the equivalence individually.

- *From left to right.* Assume that there is some argument B in $Arg_T(D)$ that defeats A on $A' \in \mathtt{sub}(A)$. Since A' is defeated, it must be rule-based. Let r be the top rule of A'. Then B is either observation-based or rule-based, with some $r' \in D$ as its top rule. If B is observation-based, then $\mathtt{conc}(B) \in \mathcal{K}$ and $\mathtt{conc}(B) \in \overline{\mathtt{cons}(r)}$. This implies that D defeats r. Otherwise B has some rule $r' \in D$ as its top rule. In that case, there must be an argument with top rule r' in $Arg_T(D) = Arg_T(D \cup \{r'\})$, and hence r' is applicable by D. Since B defeats A', we have $\mathtt{cons}(r') \in \overline{\mathtt{cons}(r)}$ and $r' \not< r$, and hence D defeats r. Finally, since r is the top rule of A' and A' is a subargument of A, we have $r \in \mathtt{defrules}(A)$.
- *From right to left.* Assume that $D \subseteq \mathcal{R}$ defeats some $r \in \mathtt{defrules}(A)$. Since $r \in \mathtt{defrules}(A)$, there must be a subargument of A with top rule r. Let $A' \in \mathtt{sub}(A)$ be such a subargument. Since D defeats r, either (i) there is some $l \in \overline{\mathtt{cons}(r)}$ in $\mathcal{K}$ or (ii) there is an $r' \in D$ applicable by D with $r' \not< r$ and whose consequent is a contradictory of r. In case (i), there is an observation-based argument in $Arg_T(D)$ that defeats A on A'. In case (ii), since r' is applicable by D, there is an argument B in $Arg_T(D)$ with top rule r'. The argument B is not less preferred than A under the last-link principle. Hence B defeats A' and thus also A. □

Analogously to defeat, we 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 not defeated by any argument in S must not defeat A. We rephrase this notion into defence by rule sets.

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

Example 12. *Consider the AT $T = (AS, \mathcal{K})$ with $AS = (\mathcal{L}, \overline{\ }, \mathcal{R}, \leq)$ and*

- *$\mathcal{L} = \{q_1, \neg q_1, q_2, \neg q_2, q_3, \neg q_3, a, b, c\}$;*
- *$q \in \overline{\neg q}$ and $\neg q \in \overline{q}$ for each $q \in \{q_1, q_2, q_3\}$, $\overline{a} = \{b\}$, $\overline{b} = \{a, c\}$ and $\overline{c} = \{b\}$;*
- *$\mathcal{R} = \{r_1 : (q_1 \Rightarrow a), r_2 : (q_2 \Rightarrow b), r_3 : (q_3 \Rightarrow c)\}$;*
- *$q_1 \Rightarrow a > q_2 \Rightarrow b$; and*

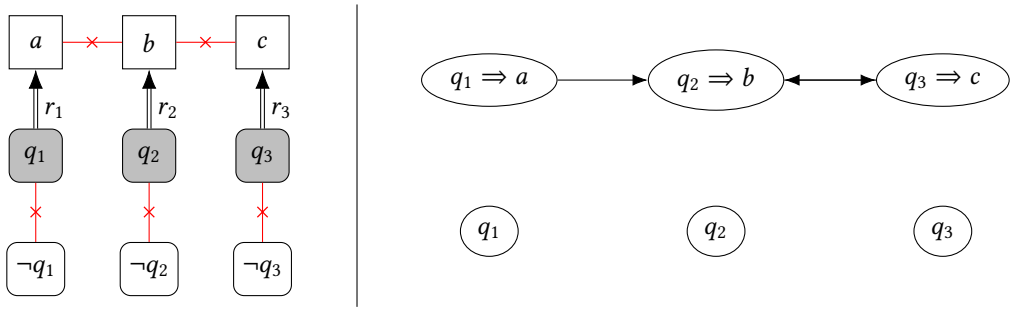

Fig. 3. Illustration of the argumentation theory in Example 12 and its corresponding argumentation framework. The preference over the rules is $r_1 > r_2$.

- $\mathcal{K} = \{q_1, q_2, q_3\}$.

This AT and its corresponding argumentation framework are illustrated in Figure 3.

- *Consider* $D = \emptyset$. *The set* U *of rules not defeated by* D *is* $\{r_1, r_2, r_3\}$. *Since* $\{r_1, r_2, r_3\}$ *does not defeat* r_1, *the rule* r_1 *is defended by* D. *No other rule is defended by* D.
- *Now consider* $D = \{r_1\}$. *The set of rules not defeated by* D *is* $U = \{r_1, r_3\}$. U *does not defeat* r_1 *or* r_3. *Thus* r_1 *and* r_3 *are defended by* D.
- *Similarly, for* $D = \{r_3\}$, *the set of rules undefeated by* D *is* $U = \{r_1, r_3\}$. *As* U *does not defeat* r_1 *and* r_3, *both* r_1 *and* r_3 *are defended by* D.

Note that there is a correspondence between defence by arguments and defence by rule sets. For example, D' *defends* r_1 *and* r_3. *Also the set of arguments* $Arg_T(D') = \{q_1 \Rightarrow a\}$ *defends all arguments that can be constructed from* r_1 *and* r_3, *that is,* $q_1 \Rightarrow a$ *and* $q_3 \Rightarrow c$.

We formally establish that this correspondence between defence by arguments (Definition 9) and defence by rule sets (Definition 18) holds in general.

Proposition 2. *Given an AT* $T = (AS, \mathcal{K})$, *its corresponding AF* $\langle \mathcal{A}, \mathcal{C} \rangle$, *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* $r \in \texttt{defrules}(A)$.

Proof. We consider each of the direction of the if-and-only-if separately.

- *From left to right.* Assume that $Arg_T(D)$ defends A and let r be an arbitrary rule in $\texttt{defrules}(A)$. Towards a contradiction, assume that r is not defended by D. Then by Definition 18 the set U, consisting of all rules in $\mathcal{R}$ that are not defeated by D, defeats r. By Proposition 1 there is some $B \in Arg_T(U)$ that defeats A. As $B \in Arg_T(U)$, the argument B can be constructed using $\mathcal{K}$ and U. Due to the way U is constructed (consisting only of rules not defeated by D), by Proposition 1 there is no argument $C \in Arg_T(D)$ that defeats B. Then by Definition 9, $Arg_T(D)$ does not defend A. From this contradiction it follows that r is defended by D.
- *From right to left.* Assume that D defends all $r \in \texttt{defrules}(A)$. Towards a contradiction, assume that $Arg_T(D)$ does not defend A. Then by Definition 9 there is some $B \in \mathcal{A}$ that defeats A, while no argument in $Arg_T(D)$ defeats B. By Proposition 1 this implies that each rule $r' \in \texttt{defrules}(B)$ is not defeated by D. Hence $\texttt{defrules}(B) \subseteq U$, where U is the set of all rules in $\mathcal{R}$ not defeated by D. This implies that $B \in Arg_T(U)$. As $B \in Arg_T(U)$ defeats A, by Proposition 1 U defeats a rule $r \in \texttt{defrules}(A)$. By Definition 18 r is not defended by D, which contradicts our assumption. Hence $Arg_T(D)$ must defend A. □

Based on the notion of defence for rule sets, we show a counterpart of Dung's fundamental lemma [19, Lemma 10] for rule sets (instead of argument sets) in Proposition 3. We first show that the defence relation is monotonous.

Lemma 1 (Monotonicity of defence). *Let $T = (AS, \mathcal{K})$ be an AT with $AS = (\mathcal{L}, \bar{}, \mathcal{R}, \leq)$. For each $R \subseteq \mathcal{R}$ and $r \in \mathcal{R}$, it holds that if R defends r, then each R' such that $R \subseteq R' \subseteq \mathcal{R}$ defends r.*

Proof. If R defends r, then by Definition 18 the set of rules U in $\mathcal{R}$ not defeated by R does not defeat r. Consider a set of rules $R' \subseteq \mathcal{R}$ such that $R \subseteq R'$ and let U' be the set of all rules in $\mathcal{R}$ that is not defeated by R'. Then $U' \subseteq U$, since U does not defeat r it follows that U' cannot defeat r either. □

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

Proof. Assume that that $Arg_T(R)$ is admissible and that r and r' are rules in $\mathcal{R}$ defended by R.

(i) We show that $Arg_T(R \cup \{r\})$ is admissible, that is, it defends itself and is conflict-free.
- By Definition 9 of admissibility, $Arg_T(R)$ defends each argument in $Arg_T(R)$. By Proposition 2 R defends each rule in R. As r is defended by R, R defends each rule in $R \cup \{r\}$. By monotonicity of defence (Lemma 1), $R \cup \{r\}$ defends each rule in $R \cup \{r\}$. Then for each argument A in $Arg_T(R \cup \{r\})$ it holds that each rule in $\texttt{defrules}(A)$ is defended by $R \cup \{r\}$. By Proposition 2 this implies that $Arg_T(R \cup \{r\})$ defends itself.
- To establish admissibility, what remains to be shown is conflict-freeness. Towards a contradiction, assume that $Arg_T(R \cup \{r\})$ is not conflict-free. Then there are A, B in $Arg_T(R \cup \{r\})$ such that A defeats B. As R defends each rule in $R \cup \{r\}$, by Proposition 2 it holds that $Arg_T(R)$ defends $Arg_T(R \cup \{r\})$. As $B \in Arg_T(R \cup \{r\})$ is defeated by A, there is an argument C in $Arg_T(R)$ that defeats A. However, $Arg_T(R)$ defends $Arg_T(R \cup \{r\})$, and hence there is an argument D in $Arg_T(R)$ defeating C. That implies that C and D in $Arg_T(R)$ defeat each other, which contradicts conflict-freeness of the admissible set $Arg_T(R)$. It follows that $Arg_T(R \cup \{r\})$ is conflict-free.

  Therefore $Arg_T(R \cup \{r\})$ is admissible.

(ii) From the assumption that R defends r' and Lemma 1 it directly follows that $R \cup \{r\}$ defends r'. □

Towards defining the grounded extension without constructing arguments, we define a characteristic function for sets of rules, in analogy with the characteristic function for AFs [19, Definition 16].

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

For $i > 0$, we denote i applications of def_T on $\emptyset$ by $def_T^i(\emptyset)$ 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 $T = (AS, \mathcal{K})$ with $AS = (\mathcal{L}, \bar{}, \mathcal{R}, \leq)$, let C be the least fixed point of def_T. Then $G(T) = Arg_T(C)$.*

Proof. First we show that def_T has a unique least fixed point. The def_T function is based on rule applicability and defence by Definition 19. Since defense is $\subseteq$-monotonic (recall Lemma 1) and the notion of applicability (i.e., for any $D \subseteq D' \subseteq \mathcal{R}$, if a rule is applicable by D, then the rule is also applicable by D'), we have that def_T is $\subseteq$-monotonic. Hence def_T has a unique least fixed point. The least fixed point is obtained by iterating def_T, starting with the empty set.

Next, we show by induction that $Arg_T(def_T^i(\emptyset))$ is admissible for any $i \in \mathbb{N}$ and thus $Arg_T(C)$ is admissible. For the base case $i = 0$, we have $def_T^0(\emptyset) = \emptyset$. Then $Arg_T(\emptyset)$ consists of knowledge-based arguments which cannot

be defeated. Hence $Arg_T(def_T^0(\emptyset))$ is admissible by Definition 9. Now, we assume as the induction hypothesis that $Arg_T(def_T^k(\emptyset))$ is admissible for some $k \in \mathbb{N}$. By Definition 19, $def_T^{k+1}(\emptyset)$ consists of all rules in $\mathcal{R}$ that are applicable and defended by $def_T^k(\emptyset)$. By Proposition 3, $def_T^{k+1}(\emptyset)$ is admissible. Since $\mathcal{R}$ is finite, we reach a fixed point C at some $i \in \mathbb{N}$. This implies that $Arg_T(C)$ is admissible.

Next, we show that $Arg_T(C)$ is complete. We will prove this by induction on the height of defended arguments, where the height $h(A)$ of any argument $A \in \mathcal{A}$ is defined as follows. If A is observation-based, then $h(A) = 0$. Otherwise, A has the form $A_1, \ldots, A_m \Rightarrow \texttt{conc}(A)$ and $h(A) = 1 + \max(h(A_1), \ldots, h(A_m))$.

For the base case, each argument $A \in \mathcal{A}$ with $h(A) = 0$ defended by $Arg_T(C)$ must be in $Arg_T(C)$; as $\texttt{defrules}(A) = \emptyset$ and A cannot be defeated, we have $A \in Arg_T(\emptyset)$ and hence $A \in Arg_T(C)$. Now assume as the induction hypothesis that, for each $k \leq n$, every argument $A \in \mathcal{A}$ with $h(A) = k$ defended by $Arg_T(C)$ is in $Arg_T(C)$. Let $A = A_1, \ldots, A_m \Rightarrow \texttt{conc}(A)$ be an arbitrary argument in $\mathcal{A}$ with $h(A) = n + 1$ defended by $Arg_T(C)$. As $Arg_T(C)$ defends A, by Proposition 2 C defends every rule in $\texttt{defrules}(A)$. Furthermore, each rule in $\texttt{defrules}(A)$ is applicable by C, since for each $A' \in \{A_1, \ldots, A_m\}$, A' is in $Arg_T(C)$ by the induction hypothesis. By Definition 19 of def_T, each rule in $\texttt{defrules}(A')$ is applicable by C. By Definition 16 of applicability, $\texttt{top-rule}(A)$ is also applicable by C. As all rules in $\texttt{defrules}(A)$ are applicable and defended by C, $def_T(C) = C$ contains all rules in $\texttt{defrules}(A)$, and hence $A \in Arg_T(C)$. Since every argument in $\mathcal{A}$ has finite height (by Definition 4), $Arg_T(C)$ is complete.

Finally, suppose towards a contradiction that $Arg_T(C)$ is not grounded. As $Arg_T(C)$ is complete, we have $G(T) \subset Arg_T(C)$ and hence there is an argument in $Arg_T(C)$ that is not in $G(T)$. Let j be the first iteration of def_T for which there is an argument $B \in Arg_T(def_T^j(\emptyset))$, $B \notin G(T)$, and each argument in $Arg_T(def_T^{j-1}(\emptyset))$ is in $G(T)$. By definition of def_T, every rule in $def_T^j(\emptyset)$ is applicable and defended by $def_T^{j-1}(\emptyset)$. Hence every rule in $\texttt{defrules}(B)$ is applicable and defended by $def_T^{j-1}(\emptyset)$. Then, by Proposition 2, $Arg_T(def_T^{j-1}(\emptyset))$ defends B. As each argument in $Arg_T(def_T^{j-1}(\emptyset))$ is in $G(T)$, we have that $G(T)$ defends B, which is a contradiction. We conclude that $Arg_T(C) = G(T)$. □

Example 13. *Consider the argumentation system AS from Example 1, AT* $T = (AS, \{similar\_url, trusted, too\_cheap\})$*, and the preference relation* $(trusted \Rightarrow \neg deception) < (similar\_url \Rightarrow deception)$*. The least fixed point of* def_T *is* $\{(similar\_url \Rightarrow typosquatting), (too\_cheap \Rightarrow deception), (typosquatting \Rightarrow deception)\}$.

(1) $i = 1$*: no rules are defeated by* $\emptyset$*, and hence the set of undefeated rules is* $U = \mathcal{R}$*. Firstly,* $(similar\_url \Rightarrow typosquatting)$ *is applicable by* U *and, as* $\neg typosquatting$ *is not derivable by* U*, defended by* $\emptyset$*. Secondly,* $\neg deception$ *is derivable from* U *only by* $(trusted \Rightarrow \neg deception)$*, which is less preferred than* $(too\_cheap \Rightarrow deception)$*, the latter of which is thus defended (and applicable) by* $\emptyset$*. Thus we have* $def_T^1(\emptyset) = def_T(\emptyset) = \{(similar\_url \Rightarrow typosquatting), (too\_cheap \Rightarrow deception)\}$.

(2) $i = 2$*:* $U = \mathcal{R} \setminus \{(trusted \Rightarrow \neg deception)\}$ *and thus* $\neg deception$ *is not derivable from* U *and we find that* $(typosquatting \Rightarrow deception)$ *is not defeated by* U*. Thus we have* $def_T^2(\emptyset) = def_T^1(\emptyset) \cup \{(typosquatting \Rightarrow deception)\}$.

(3) $def_T^3(\emptyset) = def_T^2(\emptyset)$*; terminate.*

The set of arguments that can be constructed based on these rules (deriving deception, typosquatting, too_cheap, and similar_url) constitute the grounded extension.

4.2 Complexity of Justification

We are now ready to establish that justification is polynomial-time computable. We show that one can compute the least fixed point of def_T in $|\mathcal{R}|/2$ iterations, starting with the empty set of rules. At the fixed point we conclude a rule to be defended or defeated.

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

Proof. Consider for each $i \in \mathbb{N}$ the sets $S^i = def_T^i(\emptyset)$ and the set of rules $D^i = \{r \in \mathcal{R} \mid r \text{ is defeated by } S^i\}$ defeated by S^i. Note that $S^i \subseteq S^{i+1}$ and $D^i \subseteq D^{i+1}$. Furthermore, S^i and D^j are disjoint for all i, j, since $Arg(S^i)$ is conflict-free by Proposition 3. Moreover, S^i defending a rule that S^{i-1} does not defend requires that S^i defeats some rule that S^{i-1} does not defeat. This implies that if $D^i = D^{i-1}$, then S^i is the least fixed point of $def_T^i(\emptyset)$. Thus on every iteration either at least one element is added to both S^i and D^i or a least fixed point is reached. Therefore, when $i = |\mathcal{R}|/2$ is reached, every $r \in \mathcal{R}$ is in either S^i or D^i, and hence a least fixed point has been reached. □

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

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

(i) *unsatisfiable if and only if there is no argument A for l in Arg_T,*
(ii) *defended if and only if there is an argument for l in $Arg_T(C)$,*
(iii) *out if and only if there is an argument for l in Arg_T and no argument for l in $Arg_T(U)$, where U is the set of rules not defeated by C, and*
(iv) *blocked otherwise.*

Proof. (i) The *unsatisfiable* case corresponds directly to Definition 10 of justification.
(ii) A literal l is *defended* if and only if there is an argument for l in the grounded extension $G(T)$ (Definition 10). By Proposition 4, $Arg_T(C) = G(T)$, so l is defended if and only if there is an argument for l based on C. This is the case if and only if there is an argument for l in $Arg_T(C)$.
(iii) A literal l is *out* if and only if (a) there is an argument for l in Arg_T and (b) each argument for l in Arg_T is defeated by some argument in $G(T)$ (Definition 10), where $G(T) = Arg_T(C)$ (Proposition 4). For any argument A for l in Arg_T, A is defeated by $Arg_T(C)$ if and only if C defeats some rule r in $\texttt{defrules}(A)$ (Proposition 1). Hence all arguments for l in Arg_T are defeated by $Arg_T(C)$ if and only if it is not possible to construct an argument for l based on only rules not defeated by C (i.e., based on U), which holds if and only if there is no argument for l in $Arg_T(U)$.
(iv) The justification statuses in Definition 10 are mutually exclusive and complementary, so a literal is *blocked* if and only if it is not unsatisfiable, defended or out. By cases (i)–(iii), this is the case if and only if there is some argument for l in $Arg_T(U)$ and there is no argument for l in $Arg_T(C)$. □

In summary, 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 $T = (AS, \mathcal{K})$ be an AT with $AS = (\mathcal{L}, \bar{}, \mathcal{R}, \leq)$ and let j be the unsatisfiable, defended, out, or blocked justification status. The problem of deciding whether a literal $l \in \mathcal{L}$ has justification status j w.r.t. T is in P.*

Proof. The following operations can be done in polynomial time.

(1) Given $R \subseteq \mathcal{R}$ and $l \in \mathcal{L}$, decide if there is some argument for l in $Arg_T(R)$. This is achieved by starting from the set $S = \mathcal{K}$ and adding conclusions of rules for which all antecedents are in S, until a fixed point is reached.
(2) Compute the least fixed point of def_T. This procedure takes polynomial time as the def_T operator is applied no more than $|\mathcal{R}|/2$ times before reaching a fixed point (Proposition 5).
(3) Given a set of rules $R' \subseteq \mathcal{R}$, compute the set of rules that are not defeated by R'.

These observations and Proposition 6 together imply that deciding whether a given $l \in \mathcal{L}$ has justification status j is achieved as follows.

- j = unsatisfiable: Decide in polynomial time (by Item 1) whether there is some argument for l in $Arg_T(\mathcal{R})$. If this is not the case, then l is unsatisfiable w.r.t. T.
- j = defended: Compute the least fixed point of def_T in polynomial time (by Item 2) and let this be C. Then decide in polynomial time (by Item 1) whether there is some argument for l in $Arg_T(C)$. It holds that l is defended w.r.t. T if only if such an argument exists (Proposition 6 Item 2).
- j = out: Compute the least fixed point of def_T in polynomial time (by Item 2) and let this be C. Compute $U = \{r \in \mathcal{R} \mid r \text{ is not defeated by } C\}$, which can be done in polynomial time (by Item 3). Then decide in polynomial time (by Item 1) whether there is an argument for l in $Arg_T(\mathcal{R})$ and whether there is no argument for l in $Arg_T(U)$. By Item 3 of Proposition 6, this holds if and only if l is out w.r.t. T.
- If j = blocked: By the above, check in polynomial time if j cannot be unsatisfiable, defended or out. This is the case if and only if j is blocked w.r.t. T (Item 4 of Proposition 6). □

4.3 Complexity of Stability

Polynomial-time decidability of justification, proven in Theorem 1, has immediate implications for the complexity of stability.

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

Proof. To decide whether a literal is stable w.r.t. 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, that is, a literal is not stable w.r.t. a justification status, is in NP. In addition to membership in coNP, we can infer coNP-hardness from earlier results on hardness for the case without preferences [38] which imply hardness for the case with 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 $j \in \{\text{unsatisfiable}, \text{defended}, \text{out}, \text{blocked}\}$. This problem turns out to be Σ_2^{P}-complete for each of the four justification statuses.

We first establish an auxiliary result that characterizes relevance of literals in terms of checking (non-)stability. Intuitively, one can verify that a queryable q is j-relevant for a literal l if one is able to find an AT for which the literal is not stable-j, but by adding q to the axioms, the literal becomes stable.

Lemma 2. *Let $T = (AS, \mathcal{K})$ be an AT, $\mathcal{Q}$ a set of queryables and j a justification status in $\{$unsatisfiable, defended, out, blocked$\}$. Given a literal $l \in \mathcal{L}$ and a queryable literal $q \in Q \setminus \mathcal{K}$ and $\overline{q} \cap \mathcal{K} = \emptyset$, q is j-relevant for l w.r.t. T and $\mathcal{Q}$ if and only if*

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

Proof. *From left to right.* If q is j-relevant for l w.r.t. T and $\mathcal{Q}$, then there is a minimal stable-j future theory $(AS, \mathcal{K}^*)$ such that $q \in \mathcal{K}^*$; by minimality, l cannot be stable-j in $(AS, \mathcal{K}^* \setminus \{q\})$. *From right to left.* Assume that there is an $(AS, \mathcal{K}' \cup \{q\})$ such that l is stable-j and l is not stable-j w.r.t. $(AS, \mathcal{K}')$. If $(AS, \mathcal{K}' \cup \{q\})$ is minimal stable-j, the claim follows. Otherwise there is some $\mathcal{K}'' \subset \mathcal{K}' \cup \{q\}$ such that l is stable-j w.r.t. $(AS, \mathcal{K}'')$. If $q \notin \mathcal{K}''$, then $(AS, \mathcal{K}'') \sqsubseteq_Q (AS, \mathcal{K}')$, contradicting that $(AS, \mathcal{K}'')$ is minimally stable-j for l. Thus $q \in \mathcal{K}''$ and the claim follows. □

Lemma 2 provides the basis for establishing Σ_2^{P}-membership for relevance for all four considered justification statuses. We establish hardness via a reduction from quantified Boolean formulas with one quantified alternation (2-QBFs). Here we detail the hardness proof for the defended status; the hardness proofs for the justification statuses unsatisfiable, out, and blocked are provided in Appendix A.

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

Proof of Theorem 2 membership and hardness for the defended status. For a given AT $T = (AS, \mathcal{K})$, justification status j, set of queryables Q, a specific queryable $q \in Q$, and a literal l, we show that deciding whether q is j-relevant for l is in Σ_2^{P}. Consider a non-deterministic construction of a future AT $T' = (AS, \mathcal{K}')$ of T. Compared to T, the guessed AT T' is the same except for the knowledge base. Each item in the knowledge base must be in the language of AS which is part of T. In other words, the non-deterministic construction of T' involves constructing $\mathcal{K}' \subseteq \mathcal{L}$. It directly follows that T' is of polynomial size with respect to T since at most $|\mathcal{L}|$ many elements are added. Check that l is not stable-j for T' and l is stable-j w.r.t. $(AS, \mathcal{K}' \cup \{q\})$. Checking whether stability holds is in coNP by Proposition 7. If the conditions hold for some non-deterministically constructed future AT, then q is j-relevant for l in T by Lemma 2. If there is no such future AT, then q is not j-relevant for l in T. Thus the decision problem is in Σ_2^{P}.

For establishing Σ_2^{P}-hardness, we give a reduction from deciding whether a given 2-QBF formula $\Phi = \exists X \forall Y \neg\phi$ is True, where ϕ is a propositional formula in conjunctive normal form (CNF), and $X = \{x_1, \ldots, x_n\}$ and $Y = \{y_1, \ldots, y_m\}$ are pairwise disjoint sets of variables. This problem is Σ_2^{P}-complete [50, 56].

For a given Φ, construct an AT T and queryables Q defined via

$$
\begin{aligned}
Q &= X \cup \overline{X} \cup Y \cup \overline{Y} \cup \{d, \overline{d}\},\\
\mathcal{L} &= Q \cup C \cup \overline{C} \cup V \cup \overline{V} \cup \{t, \overline{t}\},\\
\overline{} &= \{(x, \overline{x}), (\overline{x}, x) \mid x \in X \cup Y \cup V \cup C \cup \{d, t\}\},\\
\mathcal{R} &= \{(d, v_1, \ldots, v_n \Rightarrow t)\} \cup\\
&\quad \{(x \Rightarrow c) \mid x \in c\} \cup \{(\overline{x} \Rightarrow c) \mid \neg x \in c\} \cup\\
&\quad \{(y \Rightarrow c) \mid y \in c\} \cup \{(\overline{y} \Rightarrow c) \mid \neg y \in c\} \cup\\
&\quad \{(c_1, \ldots, c_p \Rightarrow \overline{t})\} \cup\\
&\quad \{(x_i \Rightarrow v_i), (\overline{x_i} \Rightarrow v_i) \mid x_i \in X\}\\
&\quad \{(y \Rightarrow t), (\overline{y} \Rightarrow t) \mid y \in Y\},\\
\mathcal{K} &= \emptyset,\\
\leq &= \emptyset,
\end{aligned}
$$

with $C = \{c_1, \ldots, c_p\}$ the set of clauses in ϕ, $\overline{X} = \{\overline{x} \mid x \in X\}$, $\overline{Y} = \{\overline{y} \mid y \in Y\}$, $\overline{C} = \{\overline{c} \mid c \in C\}$, and $V = \{v_i \mid x_i \in X\}$ and $\overline{V} = \{\overline{v_i} \mid x_i \in X\}$. The reduction is illustrated by an example in Figure 4.

We assume without loss of generality that d, $\overline{d}$, t, and $\overline{t}$ are fresh variables not occurring in Φ. Thereby $T = (AS, \mathcal{K})$ with $AS = (\mathcal{L}, \overline{}, \mathcal{R}, \leq)$ and Q can be constructed in polynomial time w.r.t. Φ.

Next, we argue that Φ is True if and only if d is defended-relevant for t w.r.t. T.

- *From left to right.* Assume that Φ is True. Then there is an assignment τ'_X to variables of X such that for each assignment τ'_Y to variables of Y, $\phi[\tau'_X, \tau'_Y]$ is False. Let τ_X be such an assignment. Construct a knowledge base $\mathcal{K}' = \{x \in X \mid \tau_X[x] = \text{True}\} \cup \{\overline{x} \in X \mid \tau_X[x] = \text{False}\}$. Note that $\mathcal{K}'$ must be consistent as no $x \in X$ can be assigned both True and False by τ_X. Hence $T \sqsubseteq_Q (AS, \mathcal{K}')$. We make the following observations.

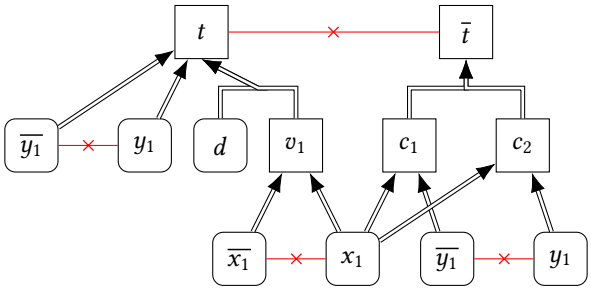

Fig. 4. Illustration of the reduction used in Theorem 2 for the defended status for the formula $\phi = (x_1 \vee y_1) \wedge (x_i \vee \neg y_1)$. The queryables $\overline{y_1}$ and y_1 are displayed twice for readability.

- t is not stable-defended w.r.t. $(AS, \mathcal{K}')$ and $\mathcal{Q}$, because t is not defended w.r.t. $(AS, \mathcal{K}')$: given that $d \notin \mathcal{K}'$ and for each $y \in Y$ both $y \notin \mathcal{K}'$ and $\overline{y} \notin \mathcal{K}'$, there is no argument for t in $Arg_{(AS,\mathcal{K}')}$.
- t is stable-defended w.r.t. $(AS, \mathcal{K}' \cup \{d\})$ and $\mathcal{Q}$. To see this, let $T'' = (AS, \mathcal{K}' \cup \{d\} \cup \mathcal{K}'')$ be an arbitrary AT such that $(AS, \mathcal{K}' \cup \{d\}) \sqsubseteq_{\mathcal{Q}} T''$. Note that $\mathcal{K}'' \subseteq Y \cup \overline{Y}$. As there is no assignment τ_Y to variables in Y such that $\phi[\tau_X, \tau_Y]$ is True, there is no argument for $\overline{t}$ in $Arg_{T''}$. On the other hand, there is at least one argument for t with top rule $(d, v_1, \ldots, v_n \Rightarrow t)$ in $Arg_{T''}$. Since the argument for t is undefeated, t is defended w.r.t. T''. As T'' was chosen arbitrarily from all T''' such that $(AS, \mathcal{K}' \cup \{d\}) \sqsubseteq_{\mathcal{Q}} T'''$, we can conclude that t is stable-defended w.r.t. $(AS, \mathcal{K}' \cup \{d\})$ and $\mathcal{Q}$.

Hence by Lemma 2, d is defended-relevant for t w.r.t. T.

- *From right to left.* Assume that d is defended-relevant for t w.r.t. T. By Definition 15, there is a minimal stable-defended future theory $T' = (AS, \mathcal{K}' \cup \{d\})$ w.r.t. T and $\mathcal{Q}$. Since t is stable-defended w.r.t. $(AS, \mathcal{K}' \cup \{d\})$ and $\mathcal{Q}$, t is defended w.r.t. $(AS, \mathcal{K}' \cup \{d\})$. Then there must be an argument for t and no argument for $\overline{t}$. Further, by minimality of $(AS, \mathcal{K}' \cup \{d\})$, t cannot be stable-defended w.r.t. $(AS, \mathcal{K}')$ and $\mathcal{Q}$. Hence there is an argument for t having the observation-based argument for d as a subargument; in particular, this is the argument with top rule $(d, v_1, \ldots, v_n \Rightarrow t)$. Since such an argument exists, for each $x \in X$, either $x \in \mathcal{K}'$ or $\overline{x} \in \mathcal{K}'$. In addition, for each $y \in Y$ we have $y \notin \mathcal{K}'$ and $\overline{y} \notin \mathcal{K}'$. This is because if there would be any $y \in Y$ such that $y \in \mathcal{K}'$ or $\overline{y} \in \mathcal{K}'$, then there would be an argument for t with top rule $y \Rightarrow t$ or $\overline{y} \Rightarrow t$, and hence the argument with top rule $(d, v_1, \ldots, v_n \Rightarrow t)$ would not have been required and $(AS, \mathcal{K}' \cup \{d\})$ would not have been minimal. Now let τ_X be the assignment to variables in X corresponding to $\mathcal{K}'$, that is, for each $x \in X$, $\tau_X[x] = \text{True}$ if and only if $x \in \mathcal{K}'$ and $\tau_X[x] = \text{False}$ if and only if $\overline{x} \in \mathcal{K}'$.
  Next, we argue that $\phi[\tau_X, \tau_Y]$ is False for each assignment τ_Y to variables in Y. Towards a contradiction, assume that there is an assignment τ_Y such that $\phi[\tau_X, \tau_Y]$ is True. Let $\mathcal{K}' \cup \{d\} \cup \mathcal{K}''$ be the corresponding knowledge base, that is, $\mathcal{K}'' = \{y \in Y \mid \tau_Y[y] = \text{True}\} \cup \{\overline{y} \in Y \mid \tau_Y[y] = \text{False}\}$. Since $\phi[\tau_X, \tau_Y]$ is True, there is an argument for $\overline{t}$ in $Arg_{(AS,\mathcal{K}' \cup \{d\} \cup \mathcal{K}'')}$, which implies that t is not defended w.r.t. $(AS, \mathcal{K}' \cup \{d\} \cup \mathcal{K}'')$. However, then t cannot be stable-defended w.r.t. $(AS, \mathcal{K}' \cup \{d\})$ and $\mathcal{Q}$; a contradiction. Hence $\phi[\tau_X, \tau_Y]$ is False for each assignment τ_Y to variables in Y, that is, Φ is True. □

In summary, the justification, stability, and relevance problems are clearly separated in terms of standard complexity classes, ranging from polynomial-time decidability (justification) to coNP-completeness (stability) and further to Σ_2^{p}-completeness (relevance). The results hold regardless of preferences over defeasible rules.

Table 1. Complexity results. First row: the fragment we focus on in this work. Second row: hardness results for justification statuses under grounded semantics for a more general setting of ASPIC$^+$ possibly including strict rules and ordinary premises. Third row: complexity upper bounds for general ASPIC$^+$ under the assumption that the justification problem is in Σ_i^{P} (i.e., beyond the grounded semantics).

	Reasoning mode		
Setting	**Justification**	**Stability**	**Relevance**
Grounded, considered fragment	in P	coNP-complete	Σ_2^{P}-complete
Grounded, general ASPIC$^+$		coNP-hard	Σ_2^{P}-hard
Beyond grounded, general ASPIC$^+$	in Σ_i^{P}	in Π_{i+1}^{P}	in $\Sigma_{i+2}^{\mathrm{P}}$

4.5 General Complexity Results for Incomplete ASPIC$^+$

The complexity results presented so far in this section hold for a specific ASPIC$^+$ fragment and under grounded semantics. Going beyond this, we also establish complexity upper and lower bounds more generally for ASPIC$^+$, as summarized in Table 1. First, we show lower bounds for more general forms of ASPIC$^+$ under grounded semantics. In this setting, we showed in Section 4.2 that justification can be decided in polynomial time and the bounds on stability and relevance are tight, that is, deciding stability is coNP-complete and deciding relevance is Σ_2^{P}-complete (Sections 4.3 and 4.4). For any fragment of ASPIC$^+$ that subsumes the fragment in this article, it holds that stability and relevance under grounded semantics must be at least as hard, that is, coNP-hard and Σ_2^{P}-hard, respectively. Note that these results, presented on the second line of Table 1, also extend to the case of including strict rules and ordinary premises.

Next, we establish general complexity upper bounds for deciding stability and relevance. Informally, independent of the choice of semantics, preference ordering, and justification status, deciding stability is at most one level higher and relevance at most two levels higher in the polynomial hierarchy than the complexity of deciding justification. In order to formalize and prove these general complexity upper bounds, let us consider an abstract function $\mathit{accepted}(T) \subseteq \mathcal{L}$ that, for a given AT T, returns a set of literals that are *accepted*. This gives a new justification status for literals. For instance, one could define that all literals that are defended in T are accepted, or that all literals for which an argument for that literal exists in an admissible set in the AT are accepted (the last condition is generally referred to as credulous acceptance under admissibility). The idea is that if deciding whether a given literal is accepted in a given AT is a decision problem in C, then we can derive upper bounds on the stability and relevance problems, rephrased accordingly w.r.t. the *accepted* function. We let C be a complexity class in the polynomial hierarchy [50, 56], such as P, NP, coNP, or Σ_2^{P}.

The notions of *stability* and *relevance* are directly adapted to the notion being accepted by replacing justification status j with being in the set $\mathit{accepted}(T)$ for the AT T for the notions of stable-j and j-relevance. In short, given an AT and a set of queryables, a literal is stable-*accepted* if the literal is accepted in all future ATs. Likewise, a queryable is *accepted*-relevant for a literal if there is a minimal future AT where the literal is stable-*accepted* and the queryable is part of this future AT.

Proposition 8. *Let C be a complexity class in the polynomial hierarchy. If deciding whether a literal is accepted in a given AT is a decision problem in C, given additionally a set of queryables, the following holds:*

- *deciding whether a literal is stable-accepted in a given AT is in* coNP^C*, and*
- *deciding whether a queryable is accepted-relevant for a given literal in a given AT is in* $\mathrm{NP}^{\mathrm{coNP}^C}$.

Proof. For showing the first item, consider the complementary problem. That is, given an AT T, a literal l, and Q a set of queryables, non-deterministically construct an AT $T' \supseteq_Q T$. Check whether $l \in \mathit{accepted}(T')$. The

latter problem is in C by assumption. This non-deterministic algorithm decides the problem of whether l is not stable-*accepted* and shows that this problem is in NP^C. Thus, the problem in the statement is in coNP^C.

For the second item, let again an AT T, a literal l, and a set of queryables Q be given. Non-deterministically construct a future AT $T' \sqsupseteq_Q T$ with $T' = (AS, \mathcal{K}')$ and $q \notin \mathcal{K}'$. Check whether l is stable-*accepted* in $(AS, \mathcal{K}' \cup \{q\})$ and not stable-*accepted* in $(AS, \mathcal{K}')$. By the same argument as in Lemma 2, we show that there is a minimal future theory of T with q such that l stable-*accepted* in this future theory. Suppose that there is a minimal $\mathcal{K}'' \subset \mathcal{K}' \cup \{q\}$ such that l is stable-*accepted* in $(AS, \mathcal{K}'')$. If $q \in \mathcal{K}''$, the claim follows. If $q \notin \mathcal{K}''$, then a future theory of $(AS, \mathcal{K}')$ witnessing that l is not stable-*accepted* in this AT is also a future theory of $(AS, \mathcal{K}'')$, contradicting that $(AS, \mathcal{K}'')$ is stable-*accepted*. Thus, either there is a contradiction, or there is a minimal future theory containing q in its knowledge base showing that l is stable-*accepted*. It holds that checking stable-*accepted* for a given AT and literal is in coNP^C. Thus the non-deterministic algorithm shown here decides the problem of *accepted*-relevance, and shows membership in $\text{NP}^{\text{coNP}^C}$. □

We remark that Proposition 8 still holds if one additionally considers strict rules and ordinary premises in ATs. In particular, as long as ATs are based on pairs with an argumentation system and a knowledge base, with the latter defined as a set, and future theories are defined based on such sets, the same proof can be applied (the part for stability makes use of the *accepted* function and future theories, and the relevance part additionally requires that Lemma 2 can be applied).

The upper bounds for stability and relevance thus can be derived directly from the complexity of the justification problem. For the justification statuses considered in this paper, these upper bounds match the tight bounds, since deciding whether a literal is $j \in \{\text{unsatisfiable}, \text{defended}, \text{out}, \text{blocked}\}$ is in P, $\text{coNP}^{\text{P}} = \text{coNP}$, and $\text{NP}^{\text{coNP}^{\text{P}}} = \Sigma_2^{\text{P}}$. Inspecting Table 1, if $C = \Sigma_i^{\text{P}}$, then $\text{coNP}^{\Sigma_i^{\text{P}}} = \Pi_{i+1}^{\text{P}}$ and $\text{NP}^{\text{coNP}^{\Sigma_i^{\text{P}}}} = \Sigma_{i+2}^{\text{P}}$. For other settings (that is, more general ASPIC$^+$ instantiations and/or other semantics), the complexity of justification is not yet known, with the following exceptions. Polynomial-time decidability for a specific fragment under grounded semantics has been shown by Odekerken et al. [39]. Results under other semantics without preferences and under stable semantics and a specific preference handling mechanism have been shown by Lehtonen et al., Lehtonen et al. [28, 29][1]. The challenge in determining the complexity of justification for ASPIC$^+$ is that, in general, the complexity of deciding justification cannot be directly inferred from existing results for AFs. This is due to the fact that AFs resulting from ASPIC$^+$ might have an exponential number of arguments.

5 Algorithms for Stability and Relevance

Complementing the complexity results, we develop declarative algorithms for deciding stability and relevance based on the declarative paradigm of answer set programming (ASP) [26, 35]. After giving background on ASP, we detailed ASP encodings for deciding justification and stability statuses, as well as iterative algorithms for deciding relevance, making use of the ASP encodings.

5.1 ASP in Brief

A normal ASP program π consists of rules r of the form $b_0 \leftarrow b_1, \ldots, b_k, \mathit{not}\ b_{k+1}, \ldots, \mathit{not}\ b_m$, where each b_i is an atom. A rule is positive if $k = m$ and a fact if $m = 0$. A literal is an atom b_i or $\mathit{not}\ b_i$. A rule without head b_0 is a constraint and a shorthand for $a \leftarrow b_1, \ldots, b_k, \mathit{not}\ b_{k+1}, \ldots, \mathit{not}\ b_m, \mathit{not}\ a$ for a fresh a. An atom b_i is $p(t_1, \ldots, t_n)$ with each t_j either a constant or a variable. An answer set program is ground if it is free of variables. For a non-ground program, GP is the set of rules obtained by applying all possible substitutions from the variables to the set of constants appearing in the program. An interpretation I, that is, a subset of all the ground atoms, satisfies a positive

[1] In addition, after the writing of this paper, the complexity of justification under multiple semantics was established for general ASPIC$^+$ with preferences under the last-link principle [27].

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(R) ← rule(R), derivable(L) : body(R,L).
8 unsat(L) ← not derivable(L), literal(L).
```

rule $r = h \leftarrow b_1, \ldots, b_k$ if and only if all positive body elements $b_1, \ldots, b_k$ being in I implies that the head atom is in I. For a program π consisting only of positive rules, let $Cl(\pi)$ be the uniquely determined interpretation I that satisfies all rules in π and no subset of I satisfies all rules in π. Interpretation I is an answer set of a ground program π if $I = Cl(\pi^I)$ where $\pi^I = \{(h \leftarrow b_1, \ldots, b_k) \mid (h \leftarrow b_1, \ldots, b_k, not\ b_{k+1}, \ldots, not\ b_m) \in \pi, \{b_{k+1}, \ldots, b_m\} \cap I = \emptyset\}$ is the reduct; and of a non-ground program π if I is an answer set of GP of π.

5.2 Encoding Justification Status

In this section we detail ASP encodings for deciding the justification status of literals for a given AT $T = (AS, \mathcal{K})$ with $AS = (\mathcal{L}, \overline{\ }, \mathcal{R}, \leq)$. These encodings will also be used as parts of our approaches to deciding stability and relevance. We represent a given AT and queryable set Q in ASP as the following set of facts, denoted by $\mathtt{AT}(T)$.

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

In words, the fact $\mathbf{axiom}(a)$ indicates that a is an axiom and $\mathbf{queryable}(a)$ indicates that a is a queryable. The heads (consequents) and bodies (antecedents) of rules are expressed via predicates **head** and **body**. A fact $\mathbf{head}(r, b)$ indicates that b is the head/consequent of the rule r, and $\mathbf{head}(r, b)$ indicates b is contained in the body/antecedents of rule r. A fact $\mathbf{ctr}(a, b)$ indicates that a and b are contradictory to each other, and $\mathbf{preferred}(a, b)$ indicates that a is (not necessarily strictly) preferred to b.

Towards encoding the grounded semantics in ASP, we introduce Listing 1 to compute what is derivable from a given knowledge base. 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 containing $\mathbf{derivable}(L)$ for all L for which $\mathbf{body}(R, L)$ holds [22]. Line 8 assigns unsatisfiable literals based on derivability.

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 $\pi_{<\text{-}just}$ is more generally applicable, π_{just} is conceptually simpler and possibly practically more efficient, and can be used in a comparison to the approximative stability algorithm by Odekerken et al. [38].

5.2.1 No preferences. In the context of ATs without preferences, arguments can only be in the grounded extension if they are defended by observation-based arguments.

Lemma 3. *[39, Lemmas 4–5] Given an AT $T = (AS, \mathcal{K})$ with $AS = (\mathcal{L}, \bar{}, \mathcal{R}, \leq)$, where $\leq = \emptyset$, an argument $A \in Arg_T$ is in $G(T)$ if and only if each argument defeating A is defeated by an observation-based argument, and defeated by an argument in $G(T)$ if and only if A is defeated by an observation-based argument.*

This property is employed in the following proposition, in which we collect rules that are not defeated by axioms U, and defended rules D (that is, rules that are not defeated by arguments based on U).

Proposition 9. *Given an AT $T = (AS, \mathcal{K})$ with $\leq = \emptyset$, let $U = \{r \in \mathcal{R} \mid \overline{\mathtt{cons}(r)} \cap \mathcal{K} = \emptyset\}$. The justification status of a literal l is*

- *defended if and only if there is an argument A for l such that $\mathtt{defrules}(A) \subseteq D$, where $D = \{r \in \mathcal{R} \mid$ an argument for $\overline{\mathtt{cons}(r)}$ based on U does not exist$\}$, and*
- *out if and only if l is not unsatisfiable and an argument A for l with $\mathtt{defrules}(A) \subseteq U$ does not exist.*

Proof. First note that U contains exactly the rules that do not have an axiom as the contrary of their head, implying that an argument B is not defeated by an observation-based argument if and only if B is based on U.

First item, left to right. Assume that l is defended. Then by Definition 10 there is an argument A' for l in $G(T)$. Suppose, towards a contradiction, that some $r_A \in \mathtt{defrules}(A')$ is not in D. Then by definition of D there is some argument B for $\overline{\mathtt{cons}(r_A)}$ based on U. By construction of U, B is not defeated by any observation-based argument, while B defeats A by Definition 6. This contradicts Lemma 3, so $\mathtt{defrules}(A') \subseteq D$.

First item, right to left. Assume that an argument A' for l exists with $\mathtt{defrules}(A') \subseteq D$. Then by construction of D and U, for each $r \in \mathtt{defrules}(A')$, there is no argument for $\overline{\mathtt{cons}(r)}$ that is not defeated by an observation-based argument. Thus l is defended by Lemma 3.

Second item, left to right. If l is out then l is not unsatisfiable and each argument for l is defeated by an argument in $G(T)$. By Lemma 3, each argument for l is defeated by an observation-based argument. Then there is no argument A' for l with $\mathtt{defrules}(A') \subseteq U$: if A' would exist, then it would not be defeated by an observation-based argument.

Second item, right to left. Assume that l is not unsatisfiable and there is no argument A for l such that $\mathtt{defrules}(A) \subseteq U$. As l is not unsatisfiable, there is an argument A' for l (with $\mathtt{defrules}(A') \not\subseteq U$). This argument is defeated by an observation-based argument on a rule $r \in \mathtt{defrules}(A') \setminus U$. Thus, A' is defeated by $G(T)$ due to Lemma 3, hence the justification status of l is out. □

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).
```

Based on Proposition 9, our ASP encoding implements a procedure for assigning the justification status of each literal as the program $\pi_{just} = \pi_{common} \cup \Delta_{just}$. The module Δ_{just} (Listing 2) assigns the justification statuses (other than unsatisfiable). 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 ensures 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, the justification status of literals whose status is not unsatisfiable, defended, or out is inferred to be blocked on Line 10.

The correctness of the ASP encoding is formalized by stating the correspondence between the justification status of literals in an AT and answers sets of our proposed encoding, based on the previous discussion and Proposition 9.

Proposition 10. *Given an AT $T = (AS, \mathcal{K})$ where $\leq = \emptyset$ and a literal $l \in \mathcal{L}$, it holds that l is j in T, where $j \in \{$unsatisfiable, defended, out, blocked$\}$, if and only if there is an answer set to the program $\mathsf{AT}(T) \cup \Delta_{just}$ that contains $\mathbf{j}(l)$.*

5.2.2 Capturing Preferences. For ATs with preferences, we present an encoding $\pi_{<\text{-}just} = \pi_{common} \cup \Delta_{<\text{-}just}$ based on the least fixed point of the defence operator (recall Definition 19), from which one can further infer the justification statuses 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 enforces the transitivity of preferences and Lines 2–3 detect when a rule is not strictly less preferred than another. Lines 4–6 set the iteration upper bound. For clarity of presentation, in the following we denote the set of defended rules on each iteration i 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 there is an argument based on D^i whose top rule is not less preferred than r and that concludes $\overline{\mathsf{cons}(r)}$. 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.

We state the correspondence between the justification status of literals in an AT that includes preferences between rules, and answers sets of our proposed encoding, based on the previous discussion and Proposition 6.

Proposition 11. *Given an AT $T = (AS, \mathcal{K})$, a literal $l \in \mathcal{L}$ and a justification status $j \in \{$unsatisfiable, defended, out, blocked$\}$, it holds that l is j in T, where $j \in \{$unsatisfiable, defended, out, blocked$\}$, if and only if there is an answer set to the program $\mathsf{AT}(T) \cup \Delta_{<\text{-}just}$ that contains $\mathbf{j}(l)$.*

5.3 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 check by conjoining our encodings for justification with a non-deterministic guess of a future AT. Concretely, via the following rules a set of queryables is non-deterministically chosen to be made axioms, and their consistency is enforced:

$$\begin{aligned}\Delta_{stab} = \{&\{\mathbf{axiom}(L)\} \leftarrow \mathbf{queryable}(L),\\ &\leftarrow \mathbf{axiom}(L), \mathbf{axiom}(C), \mathbf{ctr}(L, C)\}.\end{aligned}$$

The encoding for checking stability without preferences is $\pi_{stab} = \Delta_{stab} \cup \pi_{just}$, and for the case with preferences it 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 [25, 24] which can be used for directly computing the intersection of all

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) ← not strictly_less_preferred(X,Y), rule(X), rule(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), not defeated_by_undefeated(R,J), J+1=I, rule(R).
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), not derived_from_undefeated(L,N), max_iterations(N).
21 blocked(L) ← literal(L), not unsat(L), not out(L), not defended(L).
```

answer sets to a given program: literals that are **j** in the cautious solution are the stable-j literals. Note also that the stability status of a single literal can be decided with one ASP solver call by checking that there is no future theory where the literal is not **j**. Concretely this is achieved by including the constraint $\leftarrow \mathbf{j}(l)$ for literal $l \in \mathcal{L}$ and justification status $j \in \{\textit{unsatisfiable}, \textit{defended}, \textit{out}, \textit{blocked}\}$. The resulting answer set program does not have an answer set if and only if l is stable-j.

We state the correspondence between the j-stability of literals in an AT and answers sets of our proposed encodings, based on the previous discussion.

Proposition 12. *Given an AT $T = (AS, \mathcal{K})$, a literal $l \in \mathcal{L}$ and a justification status $j \in \{$unsatisfiable, defended, out, blocked$\}$, it holds that l is stable-j in T if and only if there is no answer sets to the program $\mathsf{AT}(T) \cup \pi_{<\text{-}stab} \cup \{\leftarrow \mathbf{j}(l)\}$. Further, if $\leq = \emptyset$ in T, then l is stable-j if and only if there is no answer sets to the program $\mathsf{AT}(T) \cup \pi_{stab} \cup \{\leftarrow \mathbf{j}(l)\}$.*

Example 14. *To illustrate our encodings, consider AS from Example 1, the AT $T' = (AS, \{trusted\})$, and the problem of deciding whether $\neg deception$ is stable-defended w.r.t. T'. By invoking an ASP solver on the answer set program $\pi_{<\text{-}just}$, we obtain that $\neg deception$ is defended w.r.t. T' (in a single iteration, as $\overline{\neg deception} = deception$ is not derivable). However, as $\pi_{<\text{-}stab}$ (conjoined with the additional constraint "$\leftarrow$ **defended**$(\neg deception)$") considers all consistent subsets of the queryables, the program has an answer set corresponding to the AT T illustrated in Example 13 in which $\neg deception$ is not defended. This is a counterexample to the claim that $\neg deception$ is stable-defended w.r.t. T'.*

5.4 ASP-Based Algorithm for Relevance

Deciding j-relevance is Σ_2^{p}-complete, as shown in Section 4.4, while deciding if a normal ASP program has an answer set is NP-complete. In order to overcome the higher complexity of deciding relevance, we propose algorithms that make iterative ASP calls, harnessing efficiency from state-of-the-art multi-shot solving in which the solver is not restarted between solver calls, retaining learned information and avoiding repeated grounding [24]. In particular, we propose ASP-based counterexample-guided abstraction refinement (CEGAR) [18, 17] 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, that is, solutions to the abstraction, are computed with a solver. If there is a candidate solution, we check with another solver if there is a counterexample to the claim that the candidate is a solution. If there is no counterexample, the candidate is guaranteed to be an actual solution. Otherwise a counterexample is obtained, leading to refining the abstraction based on an analysis of the counterexample, after which the search is continued.

The basic problem we are interested in is deciding, given an AT $T = (AS, \mathcal{K})$, a queryable q and a justification status j, whether q is j-relevant for l. We specify that candidates are consistent sets of queryables containing the given queryable q. Checking whether the candidate (say $\mathcal{K}'$) is an actual solution amounts to checking if the candidate satisfies the conditions of Lemma 2. Namely, we check if the candidate l is stable-j w.r.t. $(AS, \mathcal{K}')$ but not w.r.t. $(AS, \mathcal{K}' \setminus \{q\})$. If so, q is j-relevant for l by Lemma 2.

In addition to detailing a CEGAR algorithm for the "basic" problem of deciding whether a given queryable q is j-relevant for l, we will also detail an extension of the algorithm for the more generic problem of finding *all* queryables in T that are j-relevant for l.

Before detailing the algorithms further, we establish properties which we will employ in the algorithms to narrow down the search space upon finding counterexamples.

Proposition 13. *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 $\overline{q} \cap \mathcal{K} = \emptyset$, the following observations hold.*

(1) *If $T' = (AS, \mathcal{K}') \sqsupseteq_{\mathcal{Q}} T$ such that l is not stable-j w.r.t. T', then l is not stable-j w.r.t. $(AS, \mathcal{K}'')$ for any $\mathcal{K}'' \subseteq \mathcal{K}'$.*
(2) *Given a $T' = (AS, \mathcal{K}') \sqsupseteq_{\mathcal{Q}} T$ such that l is stable-j w.r.t. T' and $q \notin \mathcal{K}'$, l is stable-j w.r.t. $(AS, \mathcal{K}'' \setminus \{q\})$ for each consistent $\mathcal{K}'' \supseteq \mathcal{K}'$.*

Proof. (1) Assume that $T' = (AS, \mathcal{K}')$ is a future AT with $T \sqsubseteq_{\mathcal{Q}} T'$ such that l is not stable-j w.r.t. T'. Then there is a T^* with $(AS, \mathcal{K}') \sqsubseteq_{\mathcal{Q}} T^*$ such that l is not j w.r.t. T*. For each $\mathcal{K}'' \subseteq \mathcal{K}'$ we have $(AS, \mathcal{K}'') \sqsubseteq_{\mathcal{Q}} T^*$. Hence l is not stable-j w.r.t. $(AS, \mathcal{K}'')$ and $\mathcal{Q}$.

(2) Assume that $T' = (AS, \mathcal{K}')$ is a future AT with $T \sqsubseteq_{\mathcal{Q}} T'$ such that l is stable-j w.r.t. T' and $q \notin \mathcal{K}'$. Then l is stable-j w.r.t. $(AS, \mathcal{K}'')$ for each $\mathcal{K}'' \supseteq \mathcal{K}'$. Since $q \notin \mathcal{K}'$, we have $\mathcal{K}'' \setminus \{q\} \supseteq \mathcal{K}'$. Thus l is stable-j w.r.t. $(AS, \mathcal{K}'' \setminus \{q\})$. □

Our algorithmic approach to deciding whether a given queryable is j-relevant for a given literal is presented as Algorithm 1. The algorithm employs two answer set programs, π_c and π_v. The program π_c is initialized as a guess of a consistent set of queryables (Line 1) and the program π_v as π_{stab} (if no preferences need to be taken into account) or $\pi_{<\text{-}stab}$ (if preferences must be taken into account), together with a constraint enforcing that the literal of interest is not stable-j (Line 2). Concretely, $\pi_{candidate}$ is

$$\pi_{candidate} = \{\{\mathbf{axiom}(L)\} \leftarrow \mathbf{queryable}(L), \\ \leftarrow \mathbf{axiom}(L), \mathbf{axiom}(A), \mathbf{ctr}(L, A)\}.$$

Algorithm 1 ASP-based CEGAR algorithm for deciding 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 l, NO otherwise

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

A candidate is also required to include q, the queryable whose relevance is being decided. This is handled on Line 3 by an ASP solver call that tries to find a consistent set of queryables $\mathcal{K}'$ such that $\mathcal{K} \subseteq \mathcal{Q}$ and $q \in \mathcal{K}'$. If such a candidate $\mathcal{K}'$ exists, then it is extracted from the answer set in Line 4. Then Line 5 checks whether l is stable-j with respect to $(AS, \mathcal{K}')$ using the program π_v. In more detail, if there is no answer sets to π_v (with elements of $\mathcal{K}'$ as axioms), then it is not possible to find a future AT (of $(AS, \mathcal{K}')$) in which l is *not* j, and thus l is j-stable. If l is stable-j (Line 7), we check whether l is stable-j w.r.t. $\mathcal{K}' \setminus \{q\}$ (Line 8). If not, then $\mathcal{K}'$ is a witness for q being j-relevant for l (Line 10) by Lemma 2.

In other cases, we refine the abstraction. There are two types of counterexamples, corresponding to the two items of Proposition 13.

(1) If the first check fails, then the ASP solver call provides a $C \supset \mathcal{K}'$ such that l is not stable-j w.r.t. (AS, C). By the first item of Proposition 13, no subset of C can be a witness to q being j-relevant for l. Therefore all subsets of the counterexample C can be excluded as candidates in Line 6.
(2) If the second condition (in Line 8) holds, then l is stable-j w.r.t. $(AS, \mathcal{K}' \setminus \{q\})$. By the second item of Proposition 13, l is also stable-j w.r.t. any AT with a superset of $\mathcal{K}' \setminus \{q\}$ as the knowledge base. Therefore all supersets of $\mathcal{K}'$ can be ruled out in Line 9.

The refinements on Lines 6 and 9 are accomplished by the ASP constraints

$$no\_subsets(C) = \{\leftarrow not\ \mathbf{axiom}(q_1), \ldots, not\ \mathbf{axiom}(q_n)\}$$
$$no\_supersets(\mathcal{K}') = \{\leftarrow \mathbf{axiom}(q_1), \ldots, \mathbf{axiom}(q_m)\}$$

where $q_1, \ldots, q_n$ (for *no_subsets*) are the queryables not contained in the counterexample C and $q_1, \ldots, q_m$ (for *no_supersets*) are the queryables contained in the candidate $\mathcal{K}'$.

We state the correctness of Algorithm 1.

Proposition 14. *Given an AT $T = (AS, \mathcal{K})$, a set of queryables $\mathcal{Q}$, a literal $l \in \mathcal{L}$, a justification status $j \in \{unsatisfiable, defended, out, blocked\}$, and a queryable $q \in \mathcal{Q}$, it holds that q is j-relevant for l in T if and only if Algorithm 1 reports YES.*

Proof. First assume that q is j-relevant for l. By Lemma 2 there is a set of queryables $\mathcal{K}' \subseteq \mathcal{Q}$ such that $q \in \mathcal{K}'$, l is stable-j w.r.t. $\mathcal{K}'$ and l is not stable-j w.r.t. $\mathcal{K}' \setminus q$. Observe that such a $\mathcal{K}'$ is in the initial search space of the candidate generation of Algorithm 1 (Line 3). The algorithm terminates when either the search space is exhausted or a solution is found. The latter case takes place exactly when the conditions of Lemma 2 are met, as

checked on Lines 5 and 8; if the algorithm considers $\mathcal{K}'$ as a candidate, it correctly identifies q as j-relevant for l. Finally, $\mathcal{K}'$ is never wrongly removed from the search space, as the refinements of Lines 6 and 9 correspond to the two items of Proposition 13.

For the other direction, assume that q is not j-relevant for l. The algorithm does not terminate by returning YES, since otherwise a witness $\mathcal{K}'$ for q being j-relevant for l would have been found. For each iteration of the outer loop, at least one candidate is ruled out from the search (*no_subsets* and *no_supersets* both rule out the candidate itself), so the algorithm eventually terminates, returning NO. □

Example 15. *Consider the AT* $T = (AS, \{similar\_url\})$ *from Example 1. To decide if* $\neg trusted$ *is defended-relevant for* $deception$*, the abstraction at first requires candidates to be consistent and include both* $similar\_url$ *(as it is an axiom) and* $\neg trusted$ *(as it is the queryable for which relevance is of interest). We describe one possible execution of Algorithm 1 on this input.*

(1) *Suppose that* $\mathcal{K}'_1 = \{similar\_url, \neg trusted, \neg typo\_squatting, \neg too\_cheap\}$ *is considered as a candidate. Then* $deception$ *is not stable-defended w.r.t.* $\mathcal{K}'_1$ *and thus subsets of* $\mathcal{K}'_1$ *are not considered as candidates in subsequent iterations. This refinement excludes* $\{similar\_url, \neg trusted\}$, $\{similar\_url, \neg trusted, \neg typosquatting\}$, $\{similar\_url, \neg trusted, \neg too\_cheap\}$, *and* $\mathcal{K}'_1$ *itself as candidates.*

(2) *Now consider* $\mathcal{K}'_2 = \{similar\_url, \neg trusted, typosquatting\}$. *Then* $deception$ *is stable-defended w.r.t.* $\mathcal{K}'_2$ *and not stable-defended w.r.t.* $\mathcal{K}'_2 \setminus \{\neg trusted\}$. *Thus* $\neg trusted$ *is detected to be defended-relevant for* $deception$.

In practice, it is not only interesting to decide if a given queryable is relevant for some topic literal l, but also to list all relevant queryables for l. One approach to listing *all* j-relevant queryables for a given literal l is to call Algorithm 1 repeatedly for each queryable. However, this approach is suboptimal, because it is possible to potentially detect multiple relevant queryables at a time. We therefore propose an alternative approach for finding all j-relevant queryables for a given literal, by extending Algorithm 1. Specifically, we modify the algorithm to run until all subsets of queryables are implicitly considered as candidates, while collecting in each iteration all queryables that are shown to be relevant.

This approach is detailed in Algorithm 2. Line 4 tries to find a candidate $\mathcal{K}'$. Note that, in contrast to Algorithm 1, no queryable is enforced to be in $\mathcal{K}'$. The check and refinement in Line 6–7 is the same as in Algorithm 1: if l is not stable-j w.r.t. $\mathcal{K}'$, the candidate and its subsets can be discarded due to the first item of Proposition 13. Otherwise (Line 8), l is stable-j w.r.t. $\mathcal{K}'$. Then we check for each queryable in the candidate whether l is stable-j w.r.t. the candidate without this queryable (Lines 9–10). If this check fails, then the queryable is j-relevant for l due to Lemma 2. After checking this for all queryables in the candidate, supersets of the candidate can be ruled out from further consideration (Line 12). To see this, note that by the second item of Proposition 13 l is stable-j w.r.t. $Q' \setminus \{q\}$ for any consistent $Q' \supseteq \mathcal{K}'$ and any $q \notin \mathcal{K}'$. Thus no superset of the candidate Q' can satisfy the first condition of Lemma 2 and thus cannot be a witness for the relevance of any further queryable.

Proposition 15. *Given an AT* $T = (AS, \mathcal{K})$*, a set of queryables* $\mathcal{Q}$*, a literal* $l \in \mathcal{L}$ *and a justification status* $j \in \{$*unsatisfiable, defended, out, blocked*$\}$*, Algorithm 2 returns the set of queryables that are* j*-relevant for* l *w.r.t.* T.

Proof. First assume that an arbitrary $q \in \mathcal{Q}$ is j-relevant for l. By Lemma 2 there is a set of queryables $Q' \subseteq \mathcal{Q}$ such that $q \in Q'$, l is stable-j w.r.t. Q' and l is not stable-j w.r.t. $Q' \setminus \{q\}$. Such a Q' is in the initial search space of the candidate generation of Algorithm 2 (Line 4). If the algorithm considers Q' as a candidate, it correctly identifies q as being j-relevant for l and adds it to R, since Lines 6, 9 and 10 check the conditions of Lemma 2 for each queryable in the candidate. Moreover, Q' is never wrongly removed from the search space: if Line 8 is reached, then l is not stable-j w.r.t. the candidate or (by Proposition 13) any subset of the candidate. This implies that no subset of the candidate is a witness to q being relevant. On the other hand, supersets of the candidate are ruled out of the search space whenever a candidate $\mathcal{K}'$ is found such that l is stable-j w.r.t. $\mathcal{K}'$. Note that this refinement does not play a role w.r.t. any q that has already been detected to be j-relevant. Otherwise, l is

Algorithm 2 ASP-based CEGAR algorithm for finding all relevant queryables

Require: AT $T = (AS, \mathcal{K})$, $l \in \mathcal{L}$ and $j \in \{defended, out, blocked, unsatisfiable\}$
Ensure: return the set of queryables that are j-relevant for l

1: $\pi_c := \mathsf{AT}(T) \cup \pi_{candidate}$
2: $\pi_v := \mathsf{AT}(T) \cup \pi_{(<\text{-})stab} \cup \{\leftarrow \mathbf{j}(l)\}$
3: $R = \emptyset$
4: **while** π_c is satisfiable; let I be the found answer set **do**
5: $\quad \mathcal{K}' := \{q' \in \mathcal{Q} \mid \mathbf{axiom}(q') \in I\}$
6: $\quad$ **if** $\pi_v \cup \{\mathbf{axiom}(q') \mid q' \in \mathcal{K}'\}$ is satisfiable; let C be the found answer set
7: $\quad\quad$ **then** $\pi_c := \pi_c \cup no\_subsets(C)$
8: $\quad$ **else**
9: $\quad\quad$ **for** $q \in \mathcal{K}' \setminus R$ **do**
10: $\quad\quad\quad$ **if** $\pi_v \cup \{\mathbf{axiom}(q') \mid q' \in \mathcal{K}' \setminus \{q\}\}$ is satisfiable
11: $\quad\quad\quad\quad$ **then** $R := R \cup \{q\}$
12: $\quad\quad$ $\pi_c := \pi_c \cup no\_supersets(\mathcal{K}')$
13: **return** R

stable-j w.r.t. $\mathcal{K}' \setminus \{q\}$. By the second item of Proposition 13 l is also stable-j w.r.t. every consistent superset of $\mathcal{K}'$. This implies that any supersets of the candidate $\mathcal{K}'$ cannot be a witness for q being j-relevant for l. Thus q is contained in R when the algorithm terminates.

For the other direction, assume that an arbitrary q is not j-relevant for l. The algorithm only adds q to R if a set Q' witnessing the fact that q is j-relevant for l is found, as the checks on Line 6 and 10 correspond to the conditions of Lemma 2. Thus q being added to R would contradict the assumption of q not being j-relevant. Finally, on each iteration of the outer loop at least one candidate is ruled out from the search (*no_subsets* and *no_supersets* both rule out the candidate itself), so the algorithm eventually terminates, returning R which does not contain q. □

Example 16. *Consider AS from Example 1 and the AT $T = (AS, \{similar\_url\})$. To find all queryables that are defended-relevant for deception, the abstraction enforces that candidates are consistent and include the axiom similar_url. Note that, in contrast to Example 15, no other queryable is enforced to be included. We describe one possible execution of Algorithm 2 on this input.*

(1) *Candidate $\mathcal{K}'_1 = \{similar\_url, too\_cheap, typosquatting\}$. Now deception is not stable-defended w.r.t. $\mathcal{K}'_1$. Hence, subsets of $\mathcal{K}'_1$ are ruled out as candidates in the future, and the search continues.*
(2) *Candidate $\mathcal{K}'_2 = \{similar\_url, too\_cheap, typosquatting, \neg trusted\}$. Now deception is stable-defended w.r.t. $\mathcal{K}'_2$. It is then checked for each queryable $q' \in \mathcal{K}'_2 \setminus \mathcal{K}$ whether deception is stable-defended w.r.t. $\mathcal{K}'_2 \setminus \{q'\}$. Now deception is also stable-defended w.r.t. $\mathcal{K}'_2 \setminus \{too\_cheap\}$ and $\mathcal{K}'_2 \setminus \{typosquatting\}$. On the other hand, deception is not stable-defended w.r.t. $\mathcal{K}'_2 \setminus \{\neg trusted\}$, and thus $\neg trusted$ is added to R. Finally, supersets of $\mathcal{K}'_2$ are refined out for further consideration.*
(3) *Candidate $\mathcal{K}'_3 = \{similar\_url, too\_cheap, \neg trusted\}$. Now deception is stable-defended w.r.t. $\mathcal{K}'_3$, and is not stable-defended w.r.t. $\mathcal{K}'_3 \setminus \{too\_cheap\}$. Thus too_cheap is added to R and supersets of $\mathcal{K}'_3$ are not considered as candidates in subsequent iterations.*
(4) *Candidate $\mathcal{K}'_4 = \{similar\_url, typosquatting, \neg trusted\}$. Here we find that deception is stable-defended w.r.t. $\mathcal{K}'_4$, and is not stable-defended w.r.t. $\mathcal{K}'_4 \setminus \{typosquatting\}$. Moreover, typosquatting is added to R and supersets of $\mathcal{K}'_4$ are not considered as candidates in subsequent iterations.*

(5) *Candidate* $\mathcal{K}'_5 = \{similar\_url, \neg typosquatting, \neg too\_cheap, trusted\}$. *Now deception is not stable-defended w.r.t.* $\mathcal{K}'_5$, *so subsets of* $\mathcal{K}'_5$ *are not considered as candidates in subsequent iterations.*
(6) *Some further iterations are required to exhaust the search space. No further queryables are found to be defended-relevant for deception. The algorithm returns* $R = \{\neg trusted, typosquatting, too\_cheap\}$.

6 Empirical Evaluation

We empirically evaluate the ASP-based approaches to deciding stability and relevance proposed in this work. We use Clingo [25, 23, 24] (version 5.5.1) as the ASP solver and its incremental (multi-shot) features [24] for implementing the CEGAR algorithms for relevance. The implementation is available in open source at https://bitbucket.org/coreo-group/raspic[2].

Our approach constitutes the first algorithm for relevance in ASPIC$^+$ and for stability in ASPIC$^+$ with preferences, and the first exact algorithm for stability without preferences. Hence an empirical comparison to competing approaches is to a large extent not applicable here. However, for stability, we compare our ASP-based approach to a polynomial-time inexact algorithm [38, 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 non-stability for stable literals).

All experiments were run on 2.50 GHz Intel Xeon Gold 6248 machines under a per-instance time limit of 600 seconds and memory limit of 32 GB.

6.1 Benchmarks

As benchmarks, we consider both real-world and synthetic data.

For real-world benchmarks, we generated instances for the stability and relevance problems based on the argumentation system $AS = (\mathcal{L}, \bar{}, \mathcal{R}, \leq)$ and set of queryables $\mathcal{Q}$ used in an inquiry system for the intake of online trade fraud at the Netherlands Police [38]. In this setting, $|\mathcal{L}| = 60$, $|\mathcal{Q}| = 30$ and $|\mathcal{R}| = 43$. All literals in $\mathcal{L} \setminus \mathcal{Q}$ have a single contradictory. Considering the queryables in $\mathcal{Q}$, 19 queryables have a single contradictory; three literals have two contradictories; seven literals have three contradictories and one literal has four contradictories. All rules are equally preferred: $\leq = \emptyset$. Most rules have one (13) or two (14) antecedents; four rules have three antecedents; eight rules have four and the remaining four rules have five antecedents. The rules are defined in such a way that they form a tree-like structure, without (support) cycles. Each literal is assigned a layer, which informally is the largest number of rule applications to reach a queryable. Out of the 60 literals, 40 have layer 0 (this includes all 30 queryables); 6 have layer 1; 5 have layer 2; 6 have layer 3 and 3 have layer 4. We consider one literal as a "topic", for which the stability status is needed. To generate stability instances, we obtained knowledge bases by randomly sampling 25 consistent subsets of each size between 1 and 14 from $\mathcal{Q}$, as well as the empty knowledge base. Similarly, instances for relevance were created for each combination of stability instances and a queryable in $\mathcal{Q}$, randomly sampled from the set of queryables that are not axioms and whose contradictory is not an axiom.

To further study the scalability of our implementations, we also consider *synthetic data*. For this, we generated argumentation theories and queryable sets that are parametrised by the size of the language $|\mathcal{L}|$ and rule set size $|\mathcal{R}|$. We generated a set with larger instances to test our stability algorithm, and a set with smaller ones to test our relevance algorithms. We generated ATs $T = (AS, \mathcal{K})$, queryables $\mathcal{Q}$, with the following parameters.

- For the language size ($|\mathcal{L}|$), we generated instances for the stability instances with $|\mathcal{L}| \in \{50, 100, 150, 200, 250, 500, 1000, 2500, 5000\}$ and for the relevance instances with $|\mathcal{L}| \in \{50, 60, 70, 80, 90, 100, 110, 120, 130, 140, 150\}$.

[2]The implementation has been revised with bug-fixes made after the publication of the preliminary conference version of this article [42]. All empirical results reported here have been obtained by running the revised implementation.

- The number of rules was chosen to be $|\mathcal{R}| \in \{\frac{1}{2} \cdot |\mathcal{L}|, |\mathcal{L}|, \frac{3}{2} \cdot |\mathcal{L}|\}$.
- The body size of rules was chosen to be between 1 and 5, with one third of the rules having one rule antecedent, another third having two antecedents, and the remaining third was split equally to have three, four, or five antecedents.
- The literal layer distribution was selected by having $\frac{2}{3} \cdot |\mathcal{L}|$ literals with layer 0, each one-tenth of the literals for layers 1, 2, and 3, and the remaining ones with layer 4.
- The ratio between queryables and literal ($|\mathcal{Q}|/|\mathcal{L}|$) is 0.5.
- The ratio between axioms and queryables ($|\mathcal{K}|/|\mathcal{Q}|$) is 0.5.

Similarly to the fraud data set, all queryables are on layer 0, that is, there are no rules for queryable literals. We obtained a partial ordering for the rule preferences by considering all rules with contradictory consequents and sampling half of them into $\leq$. We randomly selected one topic per instance from layers 3 and 4.

We generated for each combination of $|\mathcal{L}|$ and $|\mathcal{R}|$ five different argumentation theories and for each theory, five random sets of queryables. This gives a total of 675 instances (75 per $|\mathcal{L}|$) for stability, and a total of 825 instances (75 per $|\mathcal{L}|$) for relevance. For each synthesis relevance benchmark instance, we randomly selected one literal as topic, and one queryable whose relevance for the topic is to be decided.

6.2 Results

We continue with an overview of the results of the empirical evaluation.

6.2.1 Stability. Table 2 provides an overview of the results for the task of computing the stability of *each* literal in a given AT. The table provides the number of solved instances (as the primary metric), with mean run times over solved instances and maximum run time (as secondary metrics) for each number of literals $|\mathcal{L}|$ ranging from 50 to 5000. For instances without preferences, the table provides a comparison of the performance of our ASP approach to that of the state-of-the-art inexact solver on instances without preferences. Furthermore, we also report the performance of our approach for instances with preferences.

Focusing first on the case without preferences, we observe that on the real-world instances, both our exact approach and the inexact algorithm terminate on all instances in a fraction of a second on average. On the synthetic instances, the inexact algorithm takes 82 seconds on average for instances with $|\mathcal{L}| = 5000$, while our approach takes one second on average on the same instances. Note also that while our approach is exact—always assigning correct stability statuses—the inexact algorithm mislabelled in total 69 out of 1689 topic literals in the real-world instances and 109 out of 714431 topic literals in the synthetic instances. Overall, our exact ASP-based approach thus outperforms the inexact approach in both run times and in terms of accuracy of providing (provably) correct answers. Turning to instances with preferences, we observe that deciding stability is empirically harder than without preferences. All instances up to $|\mathcal{L}| = 2500$ are solved also with preferences. The mean run time of our approach on instances with preferences at $|\mathcal{L}| = 2500$ is 3 minutes, compared to one second without preferences. At $|\mathcal{L}| = 5000$, with preferences, resource limits start becoming a bottleneck, with 25 of 75 instances solved. The solved instances at $|\mathcal{L}| = 5000$ are the ones with $|\mathcal{R}| = 2500$, while instances with $|\mathcal{R}| = 5000, 7500$ surpass the memory limit, as the size of the stability encoding with preferences ($\pi_{<\text{-}stab}$) grows with the number of rules. On the other hand, even with preferences, the mean run times remain within 5 seconds up to $|\mathcal{L}| = 500$ on the synthetic instances, and the real-world instances are solved particularly fast, with a mean run time of a fraction of a second.

6.2.2 Relevance. Turning to relevance, here we focus solely on our approach, as there are no immediate competitors to the best our knowledge. We first consider the task of deciding defended-relevance. Table 3 shows results for our approach for this task both with and without preferences. Similar tables for the task of deciding j-relevance for the other justification statuses $j \in \{\textit{unsatisfiable}, \textit{out}, \textit{blocked}\}$ are provided in Appendix B. For

Table 2. Number of solved instances, mean run times over solved instances, and maximum run times for detecting stability of all literals. Maximum number of instances per each value of $|\mathcal{L}|$ is 351 for the real-world data and 75 for synthetic.

		#solved (mean/maximum run time (s))					
Dataset	$\lvert\mathcal{L}\rvert$	In-exact		ASP		ASP under prefs	
Real	60	351	(0.1/0.1)	351	(0.1/0.1)	351	(0.1/0.1)
Synthetic	50	75	(0.1/0.1)	75	(0.1/0.1)	75	(0.1/0.1)
	100	75	(0.1/0.2)	75	(0.1/0.1)	75	(0.2/0.4)
	150	75	(0.1/0.2)	75	(0.1/0.1)	75	(0.4/0.8)
	200	75	(0.2/0.3)	75	(0.1/0.1)	75	(0.7/1.5)
	250	75	(0.3/0.5)	75	(0.1/0.1)	75	(1.2/2.6)
	500	75	(0.8/1.5)	75	(0.1/0.1)	75	(5.3/11.1)
	1000	75	(3.3/6.2)	75	(0.1/0.2)	75	(23.7/54.0)
	2500	75	(20.2/40.0)	75	(0.4/0.5)	75	(180.1/375.1)
	5000	75	(82.3/160.4)	75	(1.0/1.4)	25	(149.5/-)

the real-world instances without preferences, our approach decides defended-relevance of a query in mean run time of 0.17 seconds, with a maximum run time of 4.8 seconds. With preferences, the mean run time is 0.35 seconds and the maximum run time is 13.6 seconds. Compared to the results for stability, the run times for relevance on the synthetic instances reflect the fact that deciding relevance is harder in terms of computational complexity. On synthetic instances without preferences, our algorithm solves all instances with up to 90 literals and a majority of the instances with up to 130 literals within the run time limit. Instances with preferences appear somewhat, but not significantly, more difficult to solve: with preferences, all instances with up to 80 literals and a majority of the instances with up to 130 literals are solved.

Table 3. Number of solved instances, mean run times over solved instances, and maximum run times for deciding defended-relevance of a single queryable. Total number of instances per each value of $|\mathcal{L}|$ is 351 for the real-world instances and 75 for the synthetic ones.

		#solved (mean/maximum run time (s))			
Dataset	$\lvert\mathcal{L}\rvert$	ASP no prefs		ASP under prefs	
Real	60	351	(0.2/4.8)	351	(0.4/13.6)
Synthetic	50	75	(0.1/0.2)	75	(0.3/0.7)
	60	75	(0.4/1.8)	75	(1.2/8.7)
	70	75	(1.3/4.5)	75	(5.3/32.9)
	80	75	(3.2/28.0)	75	(20.6/190.6)
	90	75	(21.8/167.3)	73	(73.5/-)
	100	46	(17.2/-)	46	(38.0/-)
	110	54	(15.6/-)	53	(42.9/-)
	120	47	(41.4/-)	48	(94.0/-)
	130	46	(157.5/-)	38	(241.1/-)
	140	16	(0.1/-)	16	(0.8/-)
	150	13	(0.1/-)	13	(0.4/-)

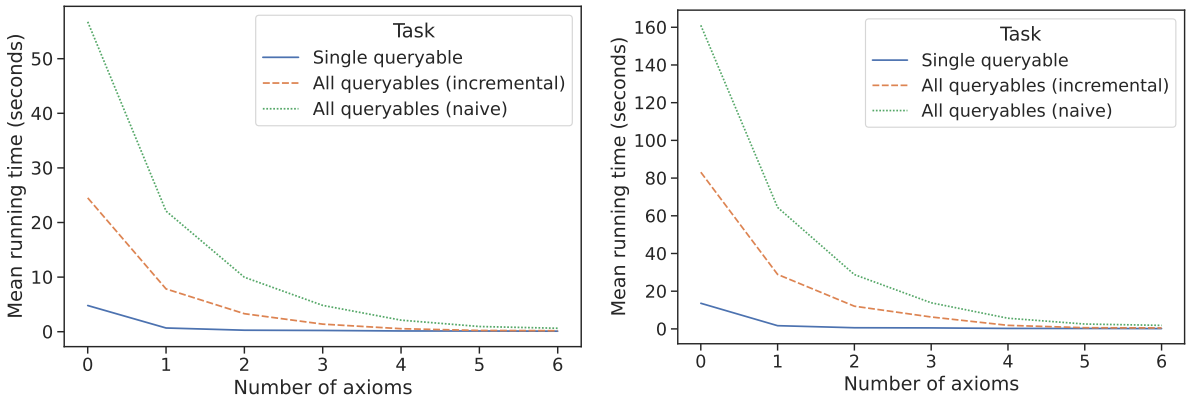

Fig. 5. Mean run times time for the tasks of (ii) deciding defended-relevance of one queryables, and finding all defended-relevant queryables (ii) incrementally or (iii) non-incrementally ("naively") for different number of axioms in the real-world data set. Left: without preferences, right: with preferences.

Next, we compare the mean run times of our approach for defended-relevance with and without preferences on the real-world instances; see Figure 5 (left: without preferences, right: with preferences) as a function of the number of axioms in the instances. In each of the two plots, we provide a comparison of the mean run times on the tasks of (i) deciding whether a single queryable is defended-relevant for a literal with Algorithm 1 ("Single queryable"), and finding all defended-relevant queryables for a literal either (ii) with Algorithm 2 ("All queryables (incremental)") or (iii) by sequentially running Algorithm 1 for each queryable ("All queryables (naive)"). First, we observe that the run times depend noticeably on the number of axioms; the fewer axioms, the longer the run times. This follows the intuition that the underlying search space over the queryables is the largest when fewer queryables are fixed as axioms. Finding all defended-relevant queryables for a literal takes noticeably more time than deciding defended-relevance of a single queryable, as expected. However, especially as the number of axioms increases, our approach also scales well to finding all defended-relevant queryables. The incremental approach (ii) to finding all defended-relevant queryables performs significantly better than the non-incremental version (iii), the incremental approach exhibiting around twice as fast mean run times.

Finally, we compare the run times of our approach for deciding j-relevance under the four different justification statuses $j \in \{\textit{unsatisfiable}, \textit{defended}, \textit{out}, \textit{blocked}\}$ on the synthetic instances. Figure 6 provides an overview of the comparison on the task of deciding whether a single queryable is relevant for a given literal without (left) and with (right) preferences, showing the number of instances (out of 825) solved (y-axis) as a function of per-instance run time (x-axis). Figure 7 provides an analogous overview for the task of finding all relevant queryables for a given literal using the incremental approach. Overall, we observe that a significant portion of the instances are solved in each case in essentially negligible time. On the task of deciding whether a single queryable is relevant (Figure 6), the choice of the justification status considered does not appear to have a significant impact on the run time distribution. In contrast, on the task of finding all relevant queryables (Figure 7), the justification status considered does have an effect on the run time distribution. The approach exhibits significantly faster run times for the statuses blocked and out compared to defended and unsatisfiable.

All in all, the results suggest that the ASP-based approach developed in this work is efficient enough to be applicable in a real-world setting.

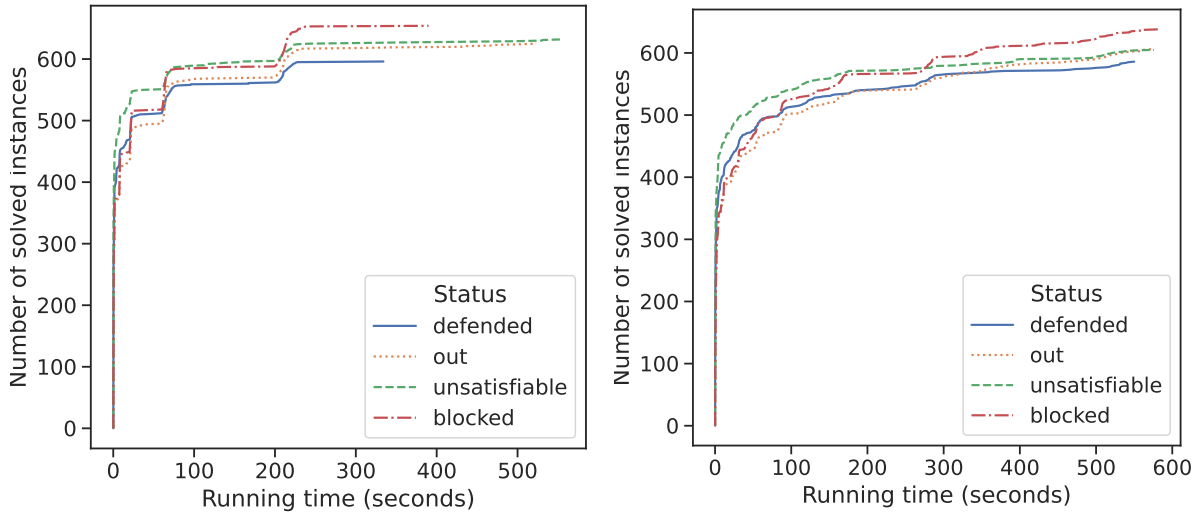

Fig. 6. Run times on synthetic instances for the task of deciding relevance of one queryable for a given literal under the different relevance status. The y-axis gives the number of instances (out of 825) solved as a function of per-instance run time (x-axis). Left: without preferences, right: with preferences.

7 Relation to Stability and Relevance in Abstract Argumentation

Finally, beyond our main contributions concerning stability and relevance in ASPIC$^+$, we discuss the relation between these problems in the structured and abstract settings. In the abstract setting, namely for incomplete

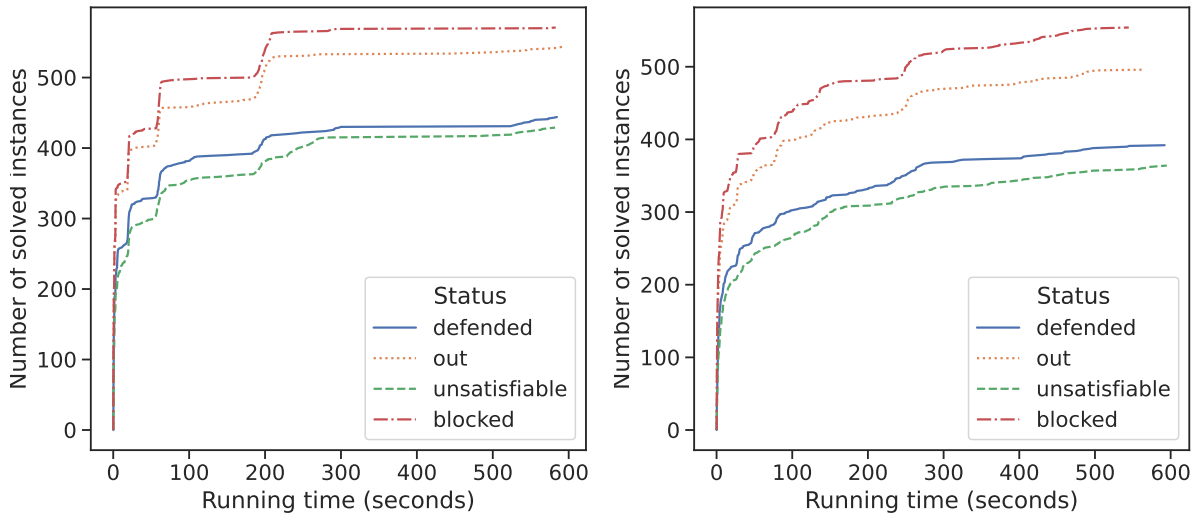

Fig. 7. Run times on synthetic instances for the task of deciding relevance of each queryable for a given literal under the differ relevance statuses. The y-axis gives the number of instances (out of 825) solved as a function of per-instance run time (x-axis). Left: without preferences, right: with preferences.

argumentation frameworks (IAFs) [11, 10, 32], stability and relevance were previously proposed by Mailly and Rossit [32] and Odekerken et al. [41], respectively. A natural question is how these notions are related to stability and relevance in the structured setting: can stability in incomplete ASPIC$^+$ be expressed in terms of the existing notion of IAF-stability, and, analogously, can ASPIC$^+$-relevance be expressed in terms of IAF-relevance? We show in this section that stability and relevance on IAFs do not capture their counterparts on the structured level. This further corroborates the need for studying computational aspects of stability and relevance on the structured level, as done in this article.

7.1 Stability and Relevance in Incomplete Abstract Frameworks

We start by recalling the definition of IAFs and the definitions of stability and relevance in the context of IAFs. IAFs are an extension to AFs, initially proposed as partial AFs by Cayrol et al. [15]. In an IAF the set of arguments is divided into a certain part ($\mathcal{A}$) and an uncertain part ($\mathcal{A}^?$). For the uncertain arguments, it is not known whether they are part of the argumentation framework or not. Although, in general, IAFs may have uncertain defeats as well, these are not required for our purposes. Therefore, we only consider so-called argument-incomplete argumentation frameworks.

Definition 20 (Incomplete argumentation framework, IAF). *An incomplete argumentation framework is a tuple $\mathcal{I} = \langle \mathcal{A}, \mathcal{A}^?, C \rangle$, where*

- $\mathcal{A} \cap \mathcal{A}^? = \emptyset$,
- *$\mathcal{A}$ is the set of certain arguments,*
- *$\mathcal{A}^?$ is the set of uncertain arguments, and*
- *$C \subseteq (\mathcal{A} \cup \mathcal{A}^?) \times (\mathcal{A} \cup \mathcal{A}^?)$ is the defeat relation.*

An IAF can be *completed* by deciding for all uncertain arguments and defeats whether or not the arguments are present [10].

Definition 21 (Completions). *A completion of an IAF $\mathcal{I} = \langle \mathcal{A}, \mathcal{A}^?, C \rangle$ is an AF $\langle \mathcal{A}', C' \rangle$ that satisfies (i) $\mathcal{A} \subseteq \mathcal{A}' \subseteq \mathcal{A} \cup \mathcal{A}^?$ and (ii) $C' = C|_{\mathcal{A}'}$, where the restriction $C|_{\mathcal{A}'}$ of a defeat C to a set of arguments $\mathcal{A}'$ is defined as $C|_{\mathcal{A}'} = \{(A, B) \in C \mid A \in \mathcal{A}' \text{ and } B \in \mathcal{A}'\}$.*

Next, we recall the definitions of (argument-centered) justification and stability status on IAFs as proposed by Odekerken et al. [41]. In order to keep the definitions succinct, we consider only the IN status under grounded semantics here.

Definition 22 (Justification and stability on IAFs). *Let $\mathcal{I} = \langle \mathcal{A}, \mathcal{A}^?, C \rangle$ be an IAF and let A be an argument in $\mathcal{A}$. For a completion F of $\mathcal{I}$,*

- *A is GR-IN w.r.t. F if and only if A is in the grounded extension of F, and*
- *A is stable-GR-IN w.r.t. $\mathcal{I}$ if and only if for each completion F of $\mathcal{I}$, A is GR-IN w.r.t. F.*

An IAF can also be *partially completed* by obtaining more information about the uncertain part of the IAF, that is, by adding uncertain arguments to the certain arguments or by removing uncertain arguments from the IAF [41].

Definition 23 (Partial completion). *A partial completion of an IAF $\mathcal{I} = \langle \mathcal{A}, \mathcal{A}^?, C \rangle$ is an IAF $\mathcal{I}' = \langle \mathcal{A}', \mathcal{A}^{?\prime}, C' \rangle$ with*

- *$\mathcal{A} \subseteq \mathcal{A}' \subseteq \mathcal{A} \cup \mathcal{A}^?$,*
- *$\mathcal{A}^{?\prime} \subseteq \mathcal{A}^?$, and*
- *$C' = C|_{\mathcal{A}' \cup \mathcal{A}^{?\prime}}$.*

Note that $\mathcal{A}' \cap \mathcal{A}^{?\prime} = \emptyset$ since $\mathcal{I}'$ is an IAF. We denote all possible partial completions for $\mathcal{I}$ by $\texttt{part}(\mathcal{I})$.

In analogy to relevance in the structured formalism of ASPIC$^+$, relevance in IAFs is defined by Odekerken et al. [41] in terms of a minimal stable partial completion.

Definition 24 (Minimal stable partial completion on IAFs). *Given an IAF $\mathcal{I} = \langle \mathcal{A}, \mathcal{A}^?, C \rangle$ and a certain argument $A \in \mathcal{A}$, a minimal stable-GR-IN partial completion for A w.r.t. $\mathcal{I}$ is a partial completion $\mathcal{I}'$ in $\texttt{part}(\mathcal{I})$ such that (i) A is stable-GR-IN in $\mathcal{I}'$ and (ii) there is no partial completion $\mathcal{I}''$ in $\texttt{part}(\mathcal{I})$ such that A is stable-GR-IN in $\mathcal{I}''$, $\mathcal{I}'' \neq \mathcal{I}'$ and $\mathcal{I}' \in \texttt{part}(\mathcal{I}'')$.*

Definition 25 (Relevance on IAFs). *Consider an IAF $\mathcal{I} = \langle \mathcal{A}, \mathcal{A}^?, C \rangle$, a certain argument $A \in \mathcal{A}$ and an uncertain argument $U \in \mathcal{A}^?$.*

- *Addition of U is GR-IN-relevant for A w.r.t. $\mathcal{I}$ if there is a minimal stable-GR-IN partial completion $\mathcal{I}' = \langle \mathcal{A}', \mathcal{A}^{?\prime}, C \rangle$ for A w.r.t. $\mathcal{I}$ such that $U \in \mathcal{A}'$; and*
- *Removal of U is GR-IN-relevant for A w.r.t. $\mathcal{I}$ if there is a minimal stable-GR-IN partial completion $\mathcal{I}' = \langle \mathcal{A}', \mathcal{A}^{?\prime}, C \rangle$ for A w.r.t. $\mathcal{I}$ such that $U \notin \mathcal{A}' \cup \mathcal{A}^{?\prime}$.*

7.2 IAFs Corresponding to Argument Theories and Queryables

We next specify a natural instantiation of an IAF for a given argumentation theory and set of queryables. All arguments that can be inferred from a future argumentation theory, but not from the current one, are included in the set of uncertain arguments in the IAF. For the defeat relation, we include all defeats that occur in some future argumentation theory.

Definition 26 (IAF corresponding to AT and Q). *Let $T = (AS, \mathcal{K})$ be an argumentation theory with $AS = (\mathcal{L}, \bar{}, \mathcal{R}, \leq)$ and let Q be a set of queryables. The corresponding IAF is the tuple $IAF(T, Q) = \langle \mathcal{A}, \mathcal{A}^?, C \rangle$ with*

- $\mathcal{A} = Arg_T$,
- $\mathcal{A}^? = \{A \mid \exists T' \text{ s.t. } T \sqsubseteq_Q T' \text{ and } A \in Arg_{T'}\} \setminus \mathcal{A}$, and
- $C = \{(A, B) \mid \exists T' \text{ s.t. } T \sqsubseteq_Q T', A \in Arg_{T'}, B \in Arg_{T'} \text{ and } A \text{ defeats } B \text{ in } T'\}$.

Example 17. *In Figure 8 we illustrate the IAF $\mathcal{I} = \langle \mathcal{A}, \mathcal{A}^?, C \rangle = IAF(T, Q)$ corresponding to the AT T and set of queryables Q from our running example on online trade fraud. Recall from Example 1 that the set of arguments Arg_T consists of an observation-based argument for similar_url and rule-based arguments for typosquatting and deception. In addition, all queryables (as identified in Example 6) that are not contradictory to literals in the knowledge base can become an observation-based argument in some future argumentation theory. These queryables in question correspond to six of the eight uncertain arguments, depicted with dashed borders in Figure 8. Note that there is no uncertain argument for ¬similar_url in $IAF(T, Q)$: as similar_url is in the knowledge base, there is no future argumentation theory of T that gives rise to an (observation-based) argument for ¬similar_url. In addition, $\mathcal{A}^?$ contains uncertain arguments for deception and ¬deception. Between these certain and uncertain arguments, we identify six defeats, represented by arrows. Note that there are no defeats between, for example, ¬too_cheap and too_cheap ⇒ deception. The reason for this is that there is no future argumentation theory giving rise to both of these arguments.*

The argument similar_url is stable-GR-IN w.r.t. $\mathcal{I}$ as it is in the grounded extension of every completion of $\mathcal{I}$. The other arguments are not stable-GR-IN as they can be defeated by undefeated arguments in completions of $\mathcal{I}$.

For the argument similar_url ⇒ typosquatting ⇒ deception, removal of ¬typosquatting is GR-IN-relevant w.r.t. $\mathcal{I}$. Similarly, the removal of trusted ⇒ ¬deception is GR-IN-relevant w.r.t. $\mathcal{I}$.

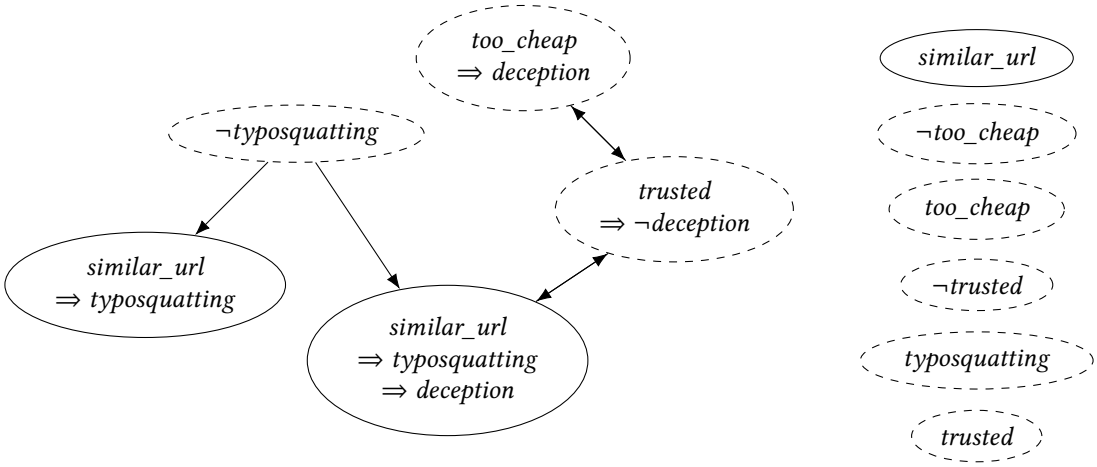

Fig. 8. The IAF corresponding to the argumentation theory and set of queryables from our running example in the domain of online trade fraud. Arguments are presented as ellipses, where those corresponding to certain arguments have solid borders and those corresponding to uncertain arguments have dashed borders. Defeats are represented by arrows.

7.3 Structured vs Abstract Stability

Having recalled the notions of stability and relevance on IAFs, we now compare them to the corresponding notions on incomplete ASPIC⁺, as considered in this paper. In particular, we consider the question of whether stable-GR-IN arguments have stable-defended conclusions and vice versa. If this is the case, then stability and relevance on incomplete ASPIC⁺ could be computed via stability and relevance on IAFs (using, for example, the algorithms proposed by Odekerken [37]).

We first give an example of an AT T and set of queryables $\mathcal{Q}$ where an argument that is stable-GR-IN w.r.t. $IAF(T, \mathcal{Q})$ (on the abstract level) has as its conclusion a literal that is stable-defended w.r.t. T and $\mathcal{Q}$ (on the structured level).

Example 18. *Recall the IAF $\mathcal{I} = \langle \mathcal{A}, \mathcal{A}^?, C \rangle = IAF(T, \mathcal{Q})$ corresponding to the AT T and set of queryables $\mathcal{Q}$ from our running example on online trade fraud, illustrated in Figure 8. By Example 17, the argument similar_url is stable-GR-IN w.r.t. $\mathcal{I}$ since similar_url in the grounded extension of every completion of $\mathcal{I}$. Furthermore, the conclusion of this argument (the literal similar_url) is stable-defended w.r.t. T and $\mathcal{Q}$.*

We next show that, in general, GR-IN-stability on the abstract level implies defended-stability on the structured level.

Proposition 16. *Let $T = (AS, \mathcal{K})$ be an argumentation theory, $\mathcal{Q}$ a set of queryables, and $\mathcal{I} = \langle \mathcal{A}, \mathcal{A}^?, C \rangle = IAF(T, \mathcal{Q})$. If $A \in \mathcal{A}$ is stable-GR-IN w.r.t. $\mathcal{I}$, then $\mathsf{conc}(A)$ is stable-defended w.r.t. T and $\mathcal{Q}$.*

Proof. Assume that $A \in \mathcal{A}$ is stable-GR-IN w.r.t. $\mathcal{I}$. Let $T' = (AS, \mathcal{K}')$ be an arbitrary future argumentation theory of T. We consider the AF defined by T' (Definition 8): let $\mathcal{A}' = Arg_{T'}$ and let C' be all defeats between arguments from $\mathcal{A}'$: $C' = \{(A, B) \mid A \in \mathcal{A}', B \in \mathcal{A}' \text{ and } A \text{ defeats } B \text{ in } T'\}$. By Definition 8, $\mathcal{A} \cup \mathcal{A}^? = \{A \mid$ there is a T'' s.t. $T \sqsubseteq_{\mathcal{Q}} T''$ and $A \in Arg_{T''}\}$. Consequently $\mathcal{A}' \subseteq \mathcal{A} \cup \mathcal{A}^?$. Since $\mathcal{K} \subseteq \mathcal{K}'$, each argument in Arg_T must also be in $Arg_{T'}$, so $\mathcal{A} \subseteq \mathcal{A}'$. Further note that $C' = C|_{\mathcal{A}'}$. By Definition 21 $\langle \mathcal{A}', C' \rangle$ is a completion of $\mathcal{I}$. Since $A \in \mathcal{A}$ is stable-GR-IN w.r.t. $\mathcal{I}$, by Definition 22 A must be in the grounded extension of $\langle \mathcal{A}', C' \rangle$ as well: $A \in G(T')$. Then by Definition 10 of justification, $\mathsf{conc}(A)$ is defended w.r.t. $\langle \mathcal{A}', C' \rangle$. Since T' was chosen

arbitrarily from the future argumentation theories of T, by Definition 13 $\mathsf{conc}(A)$ is stable-defended w.r.t. T and $\mathcal{Q}$. □

Having established that GR-IN-stability w.r.t. the IAF implies defended-stability w.r.t. the AT, we consider whether the implication also holds in the other direction, that is, does it hold that if there is a literal l that is stable-defended w.r.t. some argumentation theory T and set of queryables $\mathcal{Q}$, then there is an argument for l that is stable-GR-IN w.r.t. $IAF(T, \mathcal{Q})$? We provide a negative answer to this question via a counterexample.

COUNTEREXAMPLE 1. *Consider the argumentation theory $T = (AS, \mathcal{K})$ and set of queryables $\mathcal{Q}$ illustrated in Figure 9. This is another (simplified) example in the domain of online trade fraud, which models possible reasons for a web shop being suspect. In this example, a web shop is suspect if (i) it pretends to be old (pretends_old) although it has been recently registered (new_registration) or (ii) if it does not have many positive reviews ($\neg$many_reviews) against expectations (reviews_expected)—one would expect reviews if the web shop has a page on trustpilot, unless the web shop has been registered recently.*

In a situation where we have a web shop that (i) pretends to be old, (ii) has a page on trustpilot and (iii) does not have many reviews, as shown in Figure 9 there are three future argumentation theories: $T_1 = T$, $T_2 = T \cup \{$new_registration$\}$ and $T_3 = T \cup \{\neg$new_registration$\}$. In each of these future ATs, there is an undefeated argument for suspect: for T_1 and T_3 the argument $[$trustpilot $\Rightarrow$ reviews_expected$]$, $\neg$many_reviews $\Rightarrow$ suspect is undefeated (as the argument for $\neg$reviews_expected does not exist); for T_2 the argument pretends_old, new_registration $\Rightarrow$ suspect is undefeated. Consequently, there is an argument for suspect in the grounded extension of every future AT w.r.t. T. Hence suspect is stable-defended w.r.t. T and $\mathcal{Q}$ by Definition 13 of stability.

Now consider the $\mathcal{I} = IAF(T, Q) = \langle \mathcal{A}, \mathcal{A}^?, \mathcal{C} \rangle$ that corresponds to T and $\mathcal{Q}$, as illustrated in Figure 10. The only argument for suspect in $\mathcal{A}$ is $[$trustpilot $\Rightarrow$ reviews_expected$]$, $\neg$many_reviews $\Rightarrow$ suspect. This argument can still be defeated by the uncertain argument new_registration $\Rightarrow$ $\neg$reviews_expected. For a completion containing this argument, the argument $[$trustpilot $\Rightarrow$ reviews_expected$]$, $\neg$many_reviews $\Rightarrow$ suspect is not in the grounded extension. Since $\mathcal{I}$ has a completion in which the argument for suspect is not in the grounded extension, by Definition 22 this argument cannot be stable-GR-IN.

To conclude, if there is a literal l that is stable-defended w.r.t. some argumentation theory T and set of queryables $\mathcal{Q}$, it is not guaranteed that there is an argument for l that is stable-GR-IN w.r.t. $IAF(T, \mathcal{Q})$. The underlying issue is that IAFs do not model the phenomenon occurring in incomplete ASPIC$^+$ where the addition of an uncertain argument A may require the addition of supplementary arguments, in particular those arguments that can be constructed from the premises of A and the premises of arguments that were already in $\mathcal{A}$.

7.4 Structured vs Abstract Relevance

Finally, we establish that GR-IN-relevance on the abstract level does not imply defended-relevance on the structured level.

COUNTEREXAMPLE 2. *Consider the argumentation theory $T = (AS, \mathcal{K})$ and set of queryables $\mathcal{Q}$ illustrated in Figure 11. This AT models a dilemma related to traveling. Suppose that you are planning some journey to a location far away, on a weekday in April. You can go by train or by bike, but not at the same time: by_train $\in \overline{\text{by\_bike}}$ and by_bike $\in \overline{\text{by\_train}}$. Usually, you take the train for traveling far, provided that trains are running. On the other hand, you generally take the bike as long as there is no snow. You do not know yet whether there will be snow during your journey. But you do know that if it snows, usually all trains are canceled. At this point, there is an argument for traveling by train and there is an equally strong argument for traveling by bike. Hence the claim by_train is blocked w.r.t. T. You wish to investigate whether by_train can become stable-defended (so you can buy a train ticket) and wonder whether it makes sense to wait for the weather forecast.*

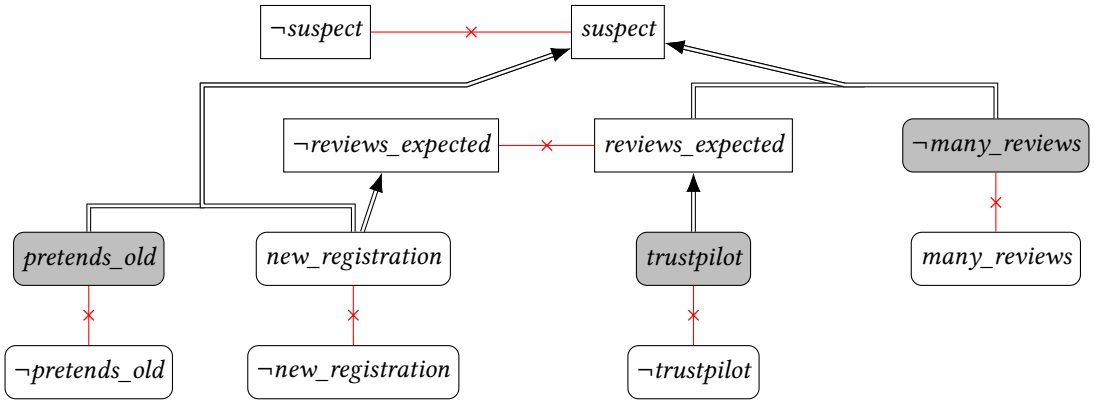

Fig. 9. The literal *suspect* is stable-defended, so no queryable is defended-relevant for this literal.

The AT T has three future argumentation theories: $T_1 = T$, $T_2 = (AS, \mathcal{K} \cup \{snow\})$ and $T_3 = (AS, \mathcal{K} \cup \{\neg snow\})$. It holds that by_train is not defended in any of these future ATs. In T_1 and T_3, the only argument for by_train (i.e., $[weekday \Rightarrow trains\_running], far \Rightarrow by\_train$) is defeated by an argument for by_bike ($[april \Rightarrow \neg snow] \Rightarrow by\_bike$, and in case of T_3 also $\neg snow \Rightarrow by\_bike$). In T_2, the only argument for by_train is defeated by the argument $snow \Rightarrow \neg trains\_running$. This implies that none of the queryables are defended-relevant for by_train w.r.t. T and $\mathcal{Q}$.

Now let $\mathcal{I} = IAF(T, \mathcal{Q}) = \langle \mathcal{A}, \mathcal{A}^?, C \rangle$, as illustrated in Figure 12. Note that there is no defeat depicted between observation-based uncertain arguments such as snow and ¬snow, as there is no future AT T' of T such that these arguments are present in T'. It holds that $\mathcal{I}$ has many different partial completions, including $\mathcal{I}' = \langle \mathcal{A} \cup \{snow\}, \emptyset, C' \rangle$.

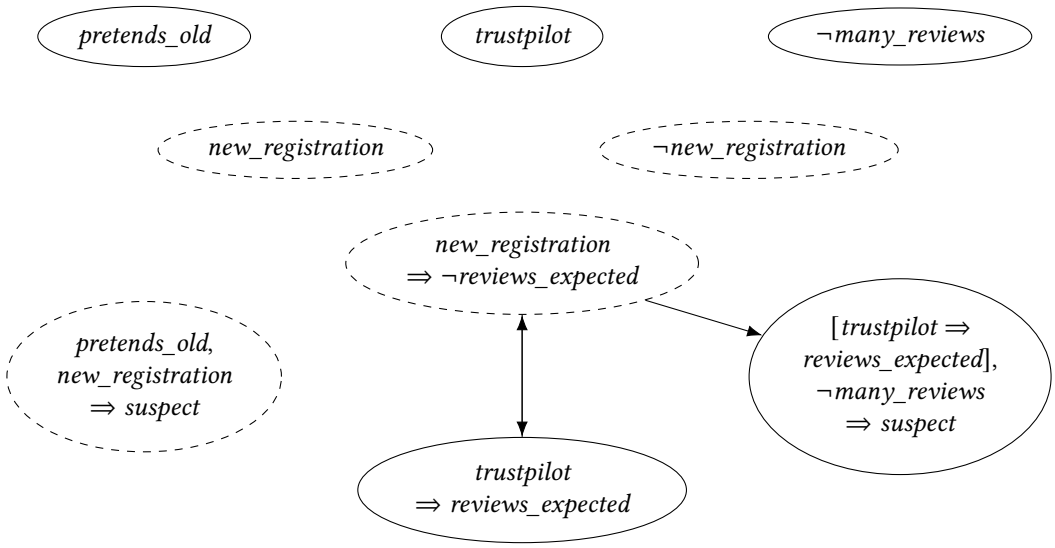

Fig. 10. An incomplete argumentation framework, where *suspect* is not stable-GR-IN. The removal of *new_registration* ⇒ ¬*reviews_expected* is GR-IN-relevant.

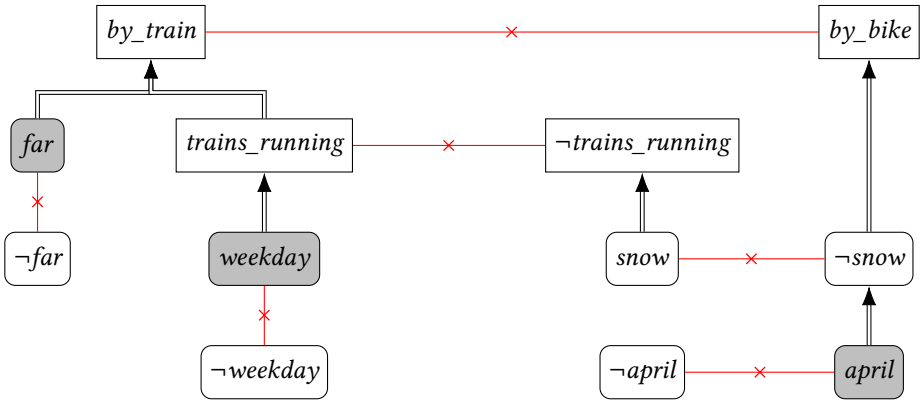

Fig. 11. The claim *by_train* is stable-blocked, so no queryable is defended-relevant for *by_train*.

In fact, $\mathcal{I}'$ is a minimal stable-GR-IN partial completion for [*weekday* ⇒ *trains_running*], *far* ⇒ *by_train w.r.t.* $\mathcal{I}$. *By Definition 25, the addition of snow is GR-IN-relevant for* [*weekday* ⇒ *trains_running*], *far* ⇒ *by_train w.r.t.* $\mathcal{I}$. *This implies that that the addition of snow* ∈ $\mathcal{A}^?$ *is GR-IN-relevant for* [*weekday* ⇒ *trains_running*], *far* ⇒ *by_train, while none of the premises of snow are defended-relevant for the conclusion literal by_train w.r.t.* T *and* Q. *This is because there is no* T' *such that* $T \sqsubseteq_Q T'$ *and IAF*$(T', Q) = \mathcal{I}'$*: snow can be an argument only in argumentation frameworks that also include snow* ⇒ ¬*trains_running. The reason for this is that both arguments snow and snow* ⇒ ¬*trains_running have exactly the same premises (i.e.,* {*snow*}*).*

In summary, we showed in this section that stable-defended literals do not necessarily have a stable-GR-IN conclusion (Counterexample 1) and that GR-IN-relevance on the abstract level does not imply defended-relevance

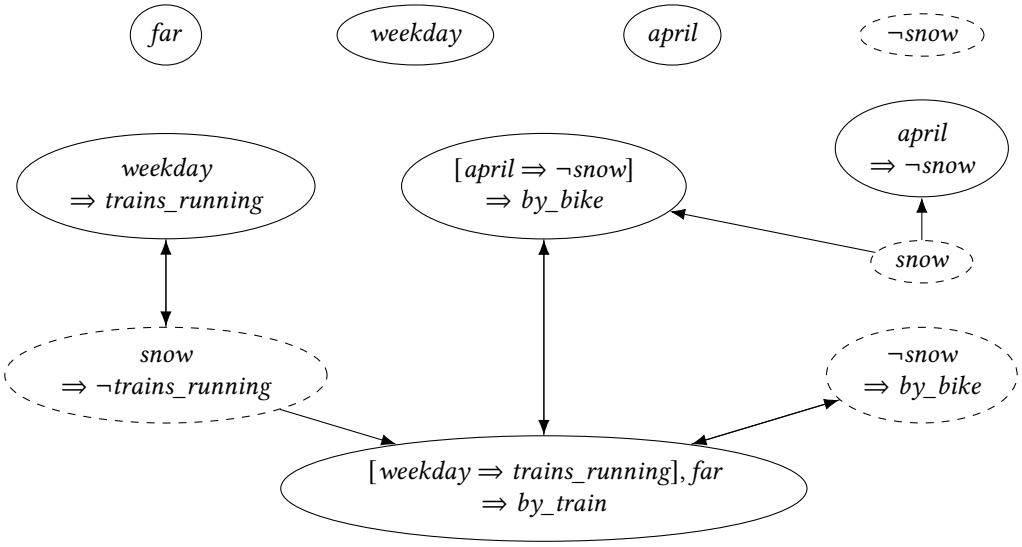

Fig. 12. Adding *snow* is defended-relevant for *by_train* in the IAF.

on the structured level (Counterexample 2), when considering the natural notion of the IAF corresponding to an AT and queryable set (recall Definition 8). The reason for this is that the notions of stability and relevance in the structured setting require that future argumentation theories can only be induced by updates on the structured level—in contrast, this is not by definition required for completions of IAFs.

Our results show that questions of stability and relevance cannot be delegated to the abstract level (at least given the notions currently proposed in the literature) and must instead be considered on the structured level. This observation is in line with earlier studies on the relation between abstract and structured formalisms [46] which have focused in particular in the resolution of attacks [34], strength of arguments [44], enforcement and strong equivalence [53, 47, 48], specification of expansions [45], forgetting arguments from a knowledge base [12] and comparing completions in structured and abstract settings [57].

8 Conclusions

We addressed the challenge of developing computational approaches to reasoning under incomplete information in the central structured argumentation formalism of ASPIC⁺. In particular, we studied the stability and relevance problems both from a complexity theoretic and an algorithmic perspective in an instantiation of ASPIC⁺ for which the two problems are motivated through applications in inquiry dialogue applied at the Netherlands Police. In terms of complexity results, we pinpointed the complexity of deciding stability and relevance under grounded semantics, establishing coNP-completeness of stability and Σ_2^{P}-completeness of relevance. Complementing the complexity results, we developed and implemented first practical exact algorithms for deciding stability and relevance, as well as for the task of finding all relevant queryables, by harnessing ASP solving. We showed through an extensive empirical evaluation on both real-world and synthetic data that our algorithmic approach exhibits promising scalability. Our approach is competitive with an earlier-proposed inexact approach with run times for both stability and relevance sufficiently low on real-world data to enable their use in online settings.

There are various interesting directions to extend our study of stability and relevance in ASPIC⁺. The main directions are to extend our complexity analysis and algorithms beyond the ASPIC⁺ instantiation we focused on to a more general fragment, as well as to consider different argumentation semantics and different preference handling mechanisms. Given that the computational aspects of ASPIC⁺ are somewhat underexplored in current literature, there are interesting research challenges to overcome in order to extend our work in these directions. Firstly, it would be interesting to study deciding stability and relevance for other semantics. While upper bounds are given in Section 4.5, lower bounds and algorithms for these semantics are not known. It is known that deciding acceptance for admissible, complete, stable and preferred semantics is typically (co)NP-hard [28]. Therefore it is likely that computing stability and relevance for these semantics is also more computationally expensive in practice compared to grounded semantics. Another direction for future work is to take into account a more general fragment of ASPIC⁺. This opens various design choices in generalizing the concept of future ATs, and thereby stability and relevance. For instance: should queryables only be able to become axioms, or can queryables also become ordinary premises in future ATs? And can queryables switch from an ordinary premise to an axiom? Should we consider all ATs that can be obtained by adding queryables to the knowledge base, or only those for which the rationality postulates are satisfied [14]? Yet another extension would be to allow for rules to be "added" in future theories. Such an extension would open up further design choices such as whether to fix concrete rules that can be added, or whether to allow for example for adding instantiations of given rule schemes. A further potential extension of our approach would be to consider preferences under weakest-link rather than last-link ordering. It has been observed that the complexity of justification under stable semantics rises to the second level of the polynomial hierarchy under weakest-link principle [29], in contrast to the (co)NP-completeness when no preferences are included [28]. Thus a similar increase in the complexity of deciding stability and relevance is expected, and along with it the need for more elaborate algorithms.

Acknowledgments

This work has been financially supported in part by University of Helsinki Doctoral Programme in Computer Science DoCS, Helsinki Institute for Information Technology HIIT, Austrian Science Fund (FWF) P35632, and Research Council of Finland grants 322869 and 356046. The authors wish to thank AnneMarie Borg for her input on the KR 2023 conference version of this article, and the Finnish Computing Competence Infrastructure (FCCI) for supporting this project with computational and data storage resources.

References

[1] Gianvincenzo Alfano, Sergio Greco, and Francesco Parisi. "Efficient Computation of Extensions for Dynamic Abstract Argumentation Frameworks: An Incremental Approach". In: *Proceedings of the 26th International Joint Conference on Artificial Intelligence (IJCAI)*. Ed. by Carles Sierra. ijcai.org, 2017, pp. 49–55.

[2] Gianvincenzo Alfano, Sergio Greco, Francesco Parisi, Gerardo I. Simari, and Guillermo R. Simari. "An Incremental Approach to Structured Argumentation over Dynamic Knowledge Bases". In: *Proceedings of the 16th International Conference on Principles of Knowledge Representation and Reasoning (KR)*. Ed. by Michael Thielscher, Francesca Toni, and Frank Wolter. AAAI Press, 2018, pp. 78–87.

[3] Gianvincenzo Alfano, Sergio Greco, Francesco Parisi, Gerardo I. Simari, and Guillermo R. Simari. "Incremental computation for structured argumentation over dynamic DeLP knowledge bases". In: *Artificial Intelligence* 300 (2021), p. 103553.

[4] Gianvincenzo Alfano, Sergio Greco, Francesco Parisi, and Irina Trubitsyna. "Incomplete argumentation frameworks: Properties and complexity". In: *Proceedings of the 36th AAAI Conference on Artificial Intelligence*. AAAI Press, 2022, pp. 5451–5460.

[5] Katie Atkinson, Pietro Baroni, Massimiliano Giacomin, Anthony Hunter, Henry Prakken, Chris Reed, Guillermo Simari, Matthias Thimm, and Serena Villata. "Towards artificial argumentation". In: *AI Magazine* 38.3 (2017), pp. 25–36.

[6] Pietro Baroni, Martin Caminada, and Massimiliano Giacomin. "An introduction to argumentation semantics". In: *The Knowledge Engineering Review* 26.4 (2011), pp. 365–410.

[7] Pietro Baroni, Dov Gabbay, Massimiliano Giacomin, and Leendert van der Torre, eds. *Handbook of Formal Argumentation*. College Publications, 2018.

[8] Ringo Baumann and Gerhard Brewka. "Expanding Argumentation Frameworks: Enforcing and Monotonicity Results". In: *Proceedings of the Third International Conference on Computational Models of Argument (COMMA)*. Ed. by Pietro Baroni, Federico Cerutti, Massimiliano Giacomin, and Guillermo Ricardo Simari. Vol. 216. Frontiers in Artificial Intelligence and Applications. IOS Press, 2010, pp. 75–86.

[9] Ringo Baumann, Sylvie Doutre, Jean-Guy Mailly, and Johannes P. Wallner. "Enforcement in Formal Argumentation". In: *Handbook of Formal Argumentation*. Ed. by Dov M. Gabbay, Massimiliano Giacomin, Guillermo R. Simari, and Matthias Thimm. Vol. 2. College Publications, 2021, pp. 445–510.

[10] Dorothea Baumeister, Matti Järvisalo, Daniel Neugebauer, Andreas Niskanen, and Jörg Rothe. "Acceptance in incomplete argumentation frameworks". In: *Artificial Intelligence* 295 (2021), p. 103470.

[11] Dorothea Baumeister, Daniel Neugebauer, Jörg Rothe, and Hilmar Schadrack. "Verification in incomplete argumentation frameworks". In: *Artificial Intelligence* 264 (2018), pp. 1–26.

[12] Matti Berthold, Anna Rapberger, and Markus Ulbricht. "Forgetting Aspects in Assumption-Based Argumentation". In: *Proceedings of the 20th International Conference on Principles of Knowledge Representation and Reasoning (KR)*. Ed. by Pierre Marquis, Tran Cao Son, and Gabriele Kern-Isberner. IJCAI, 2023, pp. 86–96.

[13] AnneMarie Borg and Floris Bex. "Enforcing Sets of Formulas in Structured Argumentation". In: *Proceedings of the 18th International Conference on Principles of Knowledge Representation and Reasoning (KR)*. Ed. by Meghyn Bienvenu, Gerhard Lakemeyer, and Esra Erdem. IJCAI, 2021, pp. 130–140.

[14] Martin Caminada and Leila Amgoud. "On the evaluation of argumentation formalisms". In: *Artificial Intelligence* 171.5-6 (2007), pp. 286–310.

[15] Claudette Cayrol, Caroline Devred, and Marie-Christine Lagasquie-Schiex. "Handling Ignorance in Argumentation: Semantics of Partial Argumentation Frameworks". In: *Proceedings of the Ninth European Conference on Symbolic and Quantitative Approaches to Reasoning with Uncertainty (ECSQARU)*. Ed. by Khaled Mellouli. Vol. 4724. Lecture Notes in Computer Science. Springer, 2007, pp. 259–270.

[16] Claudette Cayrol, Florence Dupin de Saint-Cyr, and Marie-Christine Lagasquie-Schiex. "Change in abstract argumentation frameworks: Adding an argument". In: *Journal of Artificial Intelligence Research* 38 (2010), pp. 49–84.

[17] Edmund M. Clarke, Orna Grumberg, Somesh Jha, Yuan Lu, and Helmut Veith. "Counterexample-guided abstraction refinement for symbolic model checking". In: *Journal of the ACM* 50.5 (2003), pp. 752–794.

[18] Edmund M. Clarke, Anubhav Gupta, and Ofer Strichman. "SAT-based counterexample-guided abstraction refinement". In: *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* 23.7 (2004), pp. 1113–1123.

[19] Phan Minh Dung. "On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games". In: *Artificial Intelligence* 77 (1995), pp. 321–357.

[20] Marcelo A. Falappa, Alejandro Javier García, Gabriele Kern-Isberner, and Guillermo Ricardo Simari. "On the evolving relation between Belief Revision and Argumentation". In: *The Knowledge Engineering Review* 26.1 (2011), pp. 35–43.

[21] Bettina Fazzinga, Sergio Flesca, and Filippo Furfaro. "Revisiting the Notion of Extension over Incomplete Abstract Argumentation Frameworks". In: *Proceedings of the 29th International Joint Conference on Artificial Intelligence (IJCAI)*. Ed. by Christian Bessiere. ijcai.org, 2020, pp. 1712–1718.

[22] Martin Gebser, Amelia Harrison, Roland Kaminski, Vladimir Lifschitz, and Torsten Schaub. "Abstract gringo". In: *Theory and Practice of Logic Programming* 15.4-5 (2015), pp. 449–463.

[23] Martin Gebser, Roland Kaminski, Benjamin Kaufmann, Max Ostrowski, Torsten Schaub, and Philipp Wanko. "Theory Solving Made Easy with Clingo 5". In: *Technical Communications of ICLP*. OASICS. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2016, 2:1–2:15.

[24] Martin Gebser, Roland Kaminski, Benjamin Kaufmann, and Torsten Schaub. "Multi-shot ASP solving with clingo". In: *Theory and Practice of Logic Programming* 19.1 (2019), pp. 27–82.

[25] Martin Gebser, Benjamin Kaufmann, and Torsten Schaub. "Conflict-driven answer set solving: From theory to practice". In: *Artificial Intelligence* 187 (2012), pp. 52–89.

[26] Michael Gelfond and Vladimir Lifschitz. "The Stable Model Semantics for Logic Programming". In: *Proceedings of ICLP/SLP*. MIT Press, 1988, pp. 1070–1080.

[27] Tuomo Lehtonen, Daphne Odekerken, Johannes P. Wallner, and Matti Järvisalo. "Complexity Results and Algorithms for Preferential Argumentative Reasoning in ASPIC+". In: *Proceedings of the 21st International Conference on Principles of Knowledge Representation and Reasoning (KR)*. Ed. by Pierre Marquis, Magdalena Ortiz, and Maurice Pagnucco. IJCAI, 2024, pp. 520–530.

[28] Tuomo Lehtonen, Johannes P. Wallner, and Matti Järvisalo. "An Answer Set Programming Approach to Argumentative Reasoning in the ASPIC+ Framework". In: *Proceedings of the 17th International Conference on Principles of Knowledge Representation and Reasoning (KR)*. Ed. by Diego Calvanese, Esra Erdem, and Michael Thielscher. IJCAI, 2020, pp. 636–646.

[29] Tuomo Lehtonen, Johannes P. Wallner, and Matti Järvisalo. "Computing Stable Conclusions under the Weakest-Link Principle in the ASPIC+ Argumentation Formalism". In: *Proceedings of the 19th International Conference on Principles of Knowledge Representation and Reasoning (KR)*. Ed. by Gabriele Kern-Isberner, Gerhard Lakemeyer, and Thomas Meyer. IJCAI, 2022, pp. 215–225.

[30] Tuomo Lehtonen, Johannes P. Wallner, and Matti Järvisalo. "Declarative algorithms and complexity results for assumption-based argumentation". In: *Journal of Artificial Intelligence Research* 71 (2021), pp. 265–318.
[31] Tuomo Lehtonen, Johannes P. Wallner, and Matti Järvisalo. "Harnessing Incremental Answer Set Solving for Reasoning in Assumption-Based Argumentation". In: *Theory and Practice of Logic Programming* 21.6 (2021), pp. 717–734.
[32] Jean-Guy Mailly and Julien Rossit. "Stability in Abstract Argumentation". In: *NMR 2020 Workshop Notes*. Ed. by Maria Vanina Martinez and Ivan Varzinczak. 2020, pp. 93–99.
[33] Sanjay Modgil and Henry Prakken. "A general account of argumentation with preferences". In: *Artificial Intelligence* 195 (2013), pp. 361–397.
[34] Sanjay Modgil and Henry Prakken. "Resolutions in Structured Argumentation". In: *Proceedings of the Fourth International Conference on Computational Models of Argument (COMMA)*. Ed. by Bart Verheij, Stefan Szeider, and Stefan Woltran. Vol. 245. Frontiers in Artificial Intelligence and Applications. IOS Press, 2012, pp. 310–321.
[35] Ilkka Niemelä. "Logic Programs with Stable Model Semantics as a Constraint Programming Paradigm". In: *Annals of Mathematics and Artificial Intelligence* 25.3-4 (1999), pp. 241–273.
[36] Andreas Niskanen and Matti Järvisalo. "Algorithms for Dynamic Argumentation Frameworks: An Incremental SAT-Based Approach". In: *Proceedings of the 24th European Conference on Artificial Intelligence (ECAI)*. Ed. by Giuseppe De Giacomo, Alejandro Catalá, Bistra Dilkina, Michela Milano, Senén Barro, Alberto Bugarín, and Jérôme Lang. Vol. 325. Frontiers in Artificial Intelligence and Applications. IOS Press, 2020, pp. 849–856.
[37] Daphne Odekerken. "Finding Relevant Updates in Incomplete Argumentation Frameworks". In: *Proceedings of the Tenth International Conference on Computational Models of Argument (COMMA)*. Ed. by Chris Reed, Matthias Thimm, and Tjitze Rienstra. Vol. 388. Frontiers in Artificial Intelligence and Applications. IOS Press, 2024, pp. 181–192.
[38] Daphne Odekerken, Floris Bex, AnneMarie Borg, and Bas Testerink. "Approximating Stability for Applied Argument-based Inquiry". In: *Intelligent Systems with Applications* 16 (2022), p. 200110.
[39] Daphne Odekerken, Floris Bex, AnneMarie Borg, and Bas Testerink. *Computing the justification status of literals in polynomial time*. Technical appendix, available at https://www.uu.nl/sites/default/files/Odekerken_etal-JustificationLabelAlgorithm.pdf. 2022.
[40] Daphne Odekerken, AnneMarie Borg, and Floris Bex. "Estimating Stability for Efficient Argument-Based Inquiry". In: *Proceedings of the Eighth International Conference on Computational Models of Argument (COMMA)*. Ed. by Henry Prakken, Stefano Bistarelli, Francesco Santini, and Carlo Taticchi. Vol. 326. Frontiers in Artificial Intelligence and Applications. IOS Press, 2020, pp. 307–318.
[41] Daphne Odekerken, AnneMarie Borg, and Floris Bex. "Justification, stability and relevance in incomplete argumentation frameworks". In: *Argument & Computation* 15.3 (2024), pp. 251–308.
[42] Daphne Odekerken, Tuomo Lehtonen, AnneMarie Borg, Johannes P. Wallner, and Matti Järvisalo. "Argumentative Reasoning in ASPIC+ under Incomplete Information". In: *Proceedings of the 20th International Conference on Principles of Knowledge Representation and Reasoning (KR)*. Ed. by Pierre Marquis, Tran Cao Son, and Gabriele Kern-Isberner. IJCAI, 2023, pp. 531–541.
[43] Simon Parsons, Michael J. Wooldridge, and Leila Amgoud. "An analysis of formal inter-agent dialogues". In: *Proceedings of the First International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS)*. ACM, 2002, pp. 394–401.
[44] Henry Prakken. "Formalising an Aspect of Argument Strength: Degrees of Attackability". In: *Proceedings of the Ninth International Conference on Computational Models of Argument (COMMA)*. Ed. by Francesca Toni, Sylwia Polberg, Richard Booth, Martin Caminada, and Hiroyuki Kido. Vol. 353. Frontiers in Artificial Intelligence and Applications. IOS Press, 2022, pp. 296–307.

[45] Henry Prakken. "Relating Abstract and Structured Accounts of Argumentation Dynamics: the Case of Expansions". In: *Proceedings of the 20th International Conference on Principles of Knowledge Representation and Reasoning (KR)*. Ed. by Pierre Marquis, Tran Cao Son, and Gabriele Kern-Isberner. IJCAI, 2023, pp. 562–571.

[46] Henry Prakken and Michiel De Winter. "Abstraction in Argumentation: Necessary but Dangerous". In: *Proceedings of the Seventh International Conference on Computational Models of Argument (COMMA)*. Ed. by Sanjay Modgil, Katarzyna Budzynska, and John Lawrence. Vol. 305. Frontiers in Artificial Intelligence and Applications. IOS Press, 2018, pp. 85–96.

[47] Anna Rapberger and Markus Ulbricht. "On Dynamics in Structured Argumentation Formalisms". In: *Proceedings of the 19th International Conference on Principles of Knowledge Representation and Reasoning (KR)*. Ed. by Gabriele Kern-Isberner, Gerhard Lakemeyer, and Thomas Meyer. IJCAI, 2022, pp. 288–298.

[48] Anna Rapberger and Markus Ulbricht. "On Dynamics in Structured Argumentation Formalisms". In: *Journal of Artificial Intelligence Research* 77 (2023), pp. 563–643.

[49] Nikolaos I. Spanoudakis, Elena Constantinou, Adamos Koumi, and Antonis C. Kakas. "Modeling Data Access Legislation with Gorgias". In: *Proceedings of the 30th International Conference on Industrial Engineering and Other Applications of Applied Intelligent Systems (IEA/AIE)*. Ed. by Salem Benferhat, Karim Tabia, and Moonis Ali. Vol. 10351. Lecture Notes in Computer Science. Springer, 2017, pp. 317–327.

[50] Larry J. Stockmeyer. "The Polynomial-Time Hierarchy". In: *Theoretical Computer Science* 3.1 (1976), pp. 1–22.

[51] Hannes Strass, Adam Wyner, and Martin Diller. "*EMIL*: Extracting Meaning from Inconsistent Language: Towards argumentation using a controlled natural language interface". In: *International Journal of Approximate Reasoning* 112 (2019), pp. 55–84.

[52] Bas Testerink, Daphne Odekerken, and Floris Bex. "A Method for Efficient Argument-Based Inquiry". In: *Proceedings of the 13th International Conference on Flexible Query Answering Systems (FQAS)*. Ed. by Alfredo Cuzzocrea, Sergio Greco, Henrik Legind Larsen, Domenico Saccà, Troels Andreasen, and Henning Christiansen. Vol. 11529. Lecture Notes in Computer Science. Springer, 2019, pp. 114–125.

[53] Johannes P. Wallner. "Structural constraints for dynamic operators in abstract argumentation". In: *Argument & Compututation* 11.1-2 (2020), pp. 151–190.

[54] Johannes P. Wallner, Andreas Niskanen, and Matti Järvisalo. "Complexity Results and Algorithms for Extension Enforcement in Abstract Argumentation". In: *Journal of Artificial Intelligence Research* 60 (2017), pp. 1–40.

[55] Douglas Walton and Erik CW Krabbe. *Commitment in dialogue: Basic concepts of interpersonal reasoning*. State University of New York Press, 1995.

[56] Celia Wrathall. "Complete Sets and the Polynomial-Time Hierarchy". In: *Theoretical Computer Science* 3.1 (1976), pp. 23–33.

[57] Antonio Yuste-Ginel and Carlo Proietti. "On the Instantiation of Argument-Incomplete Argumentation Frameworks". In: *Proceedings of the Seventh Workshop on Advances in Argumentation in Artificial Intelligence*. Ed. by Gianvincenzo Alfano and Stefano Ferilli. Vol. 3546. CEUR Workshop Proceedings. CEUR-WS.org, 2023.

A Remaining Hardness Proofs for Theorem 2

Completing the Σ_2^{P}-completeness result stated as Theorem 2, we provide here proofs for Σ_2^{P}-hardness of deciding whether a queryable is j-relevant for a literal in an AT w.r.t. a set of queryables for $j \in \{\text{unsatisfiable, out, blocked}\}$. Membership and the hardness for $j = \textit{defended}$ was established in the main text.

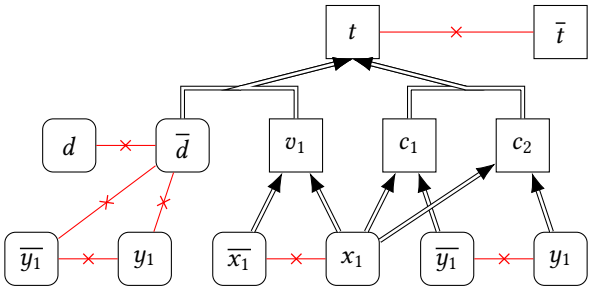

Fig. 13. Illustration of the reduction used in Theorem 2 for the formula $\phi = (x_1 \vee y_1) \wedge (x_i \vee \neg y_1)$. The queryables $\overline{y_1}$ and y_1 are shown twice for readability.

All reductions are from the Σ_2^{P}-complete problem of deciding whether a given 2-QBF formula $\Phi = \exists X \forall Y \neg\phi$ is True, where ϕ is a propositional formula in CNF and $X = \{x_1, \ldots, x_n\}$ and $Y = \{y_1, \ldots, y_m\}$ are pairwise disjoint sets of variables.

Proof of Theorem 2 for *unsatisfiable* status. To establish Σ_2^{P}-hardness, construct the AT T and queryables Q defined via

$$
\begin{aligned}
Q &= X \cup \overline{X} \cup Y \cup \overline{Y} \cup \{d, \overline{d}\},\\
\mathcal{L} &= Q \cup C \cup \overline{C} \cup V \cup \overline{V} \cup \{t, \overline{t}\},\\
\overline{\ } &= \{(x, \overline{x}), (\overline{x}, x) \mid x \in X \cup Y \cup V \cup C \cup \{d, t\}\} \cup\\
&\quad \{(y, \overline{d}), (\overline{y}, \overline{d}), (\overline{d}, y), (\overline{d}, \overline{y}) \mid y \in Y\},\\
\mathcal{R} &= \{(\overline{d}, v_1, \ldots, v_n \Rightarrow t)\} \cup\\
&\quad \{(x \Rightarrow c) \mid x \in c\} \cup \{(\overline{x} \Rightarrow c) \mid \neg x \in c\} \cup\\
&\quad \{(y \Rightarrow c) \mid y \in c\} \cup \{(\overline{y} \Rightarrow c) \mid \neg y \in c\} \cup\\
&\quad \{(c_1, \ldots, c_p \Rightarrow t)\} \cup\\
&\quad \{(x_i \Rightarrow v_i), (\overline{x_i} \Rightarrow v_i) \mid x_i \in X\},\\
\mathcal{K} &= \emptyset\\
\leq &= \emptyset,
\end{aligned}
$$

with $C = \{c_1, \ldots, c_p\}$ the set of clauses in ϕ, $\overline{X} = \{\overline{x} \mid x \in X\}$, $\overline{Y} = \{\overline{y} \mid y \in Y\}$, $\overline{C} = \{\overline{c} \mid c \in C\}$, and $V = \{v_i \mid x_i \in X\}$ and $\overline{V} = \{\overline{v_i} \mid x_i \in X\}$. The reduction is illustrated by an example in Figure 13.

Without loss of generality, we assume that d, $\overline{d}$, t, and $\overline{t}$ do not occur in Φ, that is, are fresh variables. It follows that $T = (AS, \mathcal{K})$ with $AS = (\mathcal{L}, \overline{\ }, \mathcal{R}, \leq)$ and Q can be constructed in polynomial time w.r.t. Φ. We claim that Φ is True if and only if d is unsatisfiable-relevant for t w.r.t. T.

- *From left to right.* Assume that Φ is True. Then there is an assignment τ'_X to variables of X such that for each assignment τ'_Y to variables of Y, $\phi[\tau'_X, \tau'_Y]$ is False. Let τ_X be such an assignment. Construct the knowledge base $\mathcal{K}' = \{x \in X \mid \tau_X[x] = \text{True}\} \cup \{\overline{x} \in X \mid \tau_X[x] = \text{False}\}$. Note that $\mathcal{K}'$ must be consistent, as no $x \in X$ can be assigned both True and False by τ_X. Therefore $T \sqsubseteq_Q (AS, \mathcal{K}')$. We observe the following.
  - t is not stable-unsatisfiable w.r.t. $(AS, \mathcal{K}')$ and Q, since t is not unsatisfiable w.r.t. $(AS, \mathcal{K}' \cup \{\overline{d}\})$. To see this, note that the contradictory of $\overline{d}$ is not in $\mathcal{K}'$, and hence $\mathcal{K}' \cup \{\overline{d}\}$ is a consistent knowledge base.

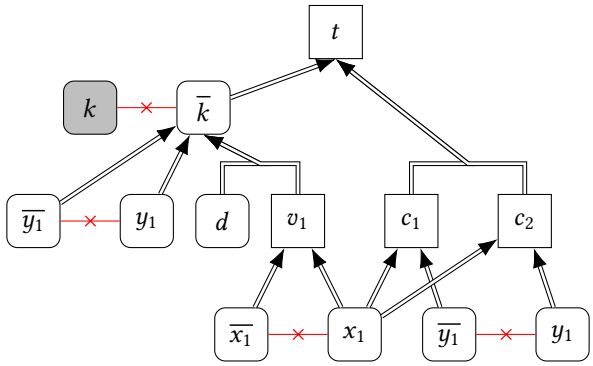

Fig. 14. Illustration of the reduction used in Theorem 2 for *out* status for the CNF formula $\phi = (x_1 \vee y_1) \wedge (x_i \vee \neg y_1)$. The queryables $\overline{y_1}$ and y_1 are shown twice for readability.

Given that for each $x \in X$ either $x \in \mathcal{K}'$ or $\overline{x} \in \mathcal{K}'$, it must be that for each $x \in X$ either $x \in \mathcal{K}' \cup \{\overline{d}\}$ or $\overline{x} \in \mathcal{K}' \cup \{\overline{d}\}$. This implies that there is an argument for t with top rule $(\overline{d}, v_1, \ldots, v_n \Rightarrow t)$ in $Arg_{(AS,\mathcal{K}' \cup \{\overline{d}\})}$.

– t is stable-unsatisfiable w.r.t. $(AS, \mathcal{K}' \cup \{d\})$ and $\mathcal{Q}$. To see this, let $T'' = (AS, \mathcal{K}' \cup \{d\} \cup \mathcal{K}'')$ be an arbitrary AT such that $(AS, \mathcal{K}' \cup \{d\}) \sqsubseteq_{\mathcal{Q}} T''$. Note that $\mathcal{K}'' \subseteq Y \cup \overline{Y}$. As there is no assignment τ_Y to variables in Y such that $\phi[\tau_X, \tau_Y]$ is True, there is no argument for t with top rule $(c_1, \ldots, c_p \Rightarrow t)$ in $Arg_{T''}$. Further, since d is in the knowledge base of T'', there is no argument for t with top rule $(\overline{d}, v_1, \ldots, v_n \Rightarrow t)$, in $Arg_{T''}$. Since there are no other rules for t and t is not in $\mathcal{Q}$, t is unsatisfiable w.r.t. T''. As T'' was chosen arbitrarily from all T''' such that $(AS, \mathcal{K}' \cup \{d\}) \sqsubseteq_{\mathcal{Q}} T'''$, we conclude that t is stable-unsatisfiable w.r.t. $(AS, \mathcal{K}' \cup \{d\})$ and $\mathcal{Q}$.

Hence by Lemma 2, d is unsatisfiable-relevant for t w.r.t. T.

- *From right to left.* Assume that d is unsatisfiable-relevant for t w.r.t. T. By Definition 15 there is some minimal stable-unsatisfiable future theory $T' = (AS, \mathcal{K}' \cup \{d\})$ w.r.t. T and $\mathcal{Q}$. Since t is stable-unsatisfiable w.r.t. $(AS, \mathcal{K}' \cup \{d\})$ and $\mathcal{Q}$, t is unsatisfiable w.r.t. $(AS, \mathcal{K}' \cup \{d\})$. Hence there is no argument for t in $Arg_{(AS,\mathcal{K}' \cup \{d\})}$.

  Further, by minimality of $(AS, \mathcal{K}' \cup \{d\})$, t cannot be stable-unsatisfiable w.r.t. $(AS, \mathcal{K}')$ and $\mathcal{Q}$. It follows that there must be some future argumentation theory of $(AS, \mathcal{K}')$ for which there is some argument for t with the observation-based argument for $\overline{d}$ as a subargument. In particular, this must be an argument with top rule $(\overline{d}, v_1, \ldots, v_n \Rightarrow t)$. It follows that for each $x \in X$ either $x \in \mathcal{K}'$ or $\overline{x} \in \mathcal{K}'$. In addition, for each $y \in Y$ we have $y \notin \mathcal{K}'$ and $\overline{y} \notin \mathcal{K}'$ (since these are contradictories of $\overline{d}$). Now let τ_X be the assignment to variables in X corresponding to $\mathcal{K}'$: for each $x \in X$, $\tau_X[x] = \text{True}$ if and only if $x \in \mathcal{K}'$ and $\tau_X[x] = \text{False}$ if and only if $\overline{x} \in \mathcal{K}'$.

  Next, we argue that $\phi[\tau_X, \tau_Y]$ is False for each assignment τ_Y to variables in Y. Towards a contradiction, assume that there is some τ_Y such that $\phi[\tau_X, \tau_Y]$ is True. Let $\mathcal{K}' \cup \{d\} \cup \mathcal{K}''$ be the corresponding knowledge base: $\mathcal{K}'' = \{y \in Y \mid \tau_Y[y] = \text{True}\} \cup \{\overline{y} \in Y \mid \tau_Y[y] = \text{False}\}$. Since $\phi[\tau_X, \tau_Y]$ is True, there is an argument for t in $Arg_{(AS,\mathcal{K}' \cup \{d\} \cup \mathcal{K}'')}$. This implies that t is not unsatisfiable w.r.t. $(AS, \mathcal{K}' \cup \{d\} \cup \mathcal{K}'')$. However, then t is not stable-unsatisfiable w.r.t. $(AS, \mathcal{K}' \cup \{d\})$ and $\mathcal{Q}$; contradiction. Therefore $\phi[\tau_X, \tau_Y]$ is False for each assignment τ_Y to variables in Y, that is, Φ is True. □

Proof of Theorem 2 for *out* status. Construct the AT T and queryables Q defined via

$$
\begin{aligned}
Q &= X \cup \overline{X} \cup Y \cup \overline{Y} \cup \{d, \overline{d}, k, \overline{k}\},\\
\mathcal{L} &= Q \cup C \cup \overline{C} \cup V \cup \overline{V} \cup \{t, \overline{t}\},\\
\overline{} &= \{(x, \overline{x}), (\overline{x}, x) \mid x \in X \cup Y \cup V \cup C \cup \{d, t, k\}\},\\
\mathcal{R} &= \{(d, v_1, \ldots, v_n \Rightarrow \overline{k})\} \cup\\
&\quad \{(x \Rightarrow c) \mid x \in c\} \cup \{(\overline{x} \Rightarrow c) \mid \neg x \in c\} \cup\\
&\quad \{(y \Rightarrow c) \mid y \in c\} \cup \{(\overline{y} \Rightarrow c) \mid \neg y \in c\} \cup\\
&\quad \{(c_1, \ldots, c_p \Rightarrow t)\} \cup\\
&\quad \{(x_i \Rightarrow v_i), (\overline{x_i} \Rightarrow v_i) \mid x_i \in X\}\\
&\quad \{(y \Rightarrow \overline{k}), (\overline{y} \Rightarrow \overline{k}) \mid y \in Y\}\\
&\quad \{(\overline{k} \Rightarrow t)\},\\
\mathcal{K} &= \{k\}\\
\leq &= \emptyset,
\end{aligned}
$$

with $C = \{c_1, \ldots, c_p\}$ the set of clauses in ϕ, $\overline{X} = \{\overline{x} \mid x \in X\}$, $\overline{Y} = \{\overline{y} \mid y \in Y\}$, $\overline{C} = \{\overline{c} \mid c \in C\}$, and $V = \{v_i \mid x_i \in X\}$ and $\overline{V} = \{\overline{v_i} \mid x_i \in X\}$. The reduction is illustrated by an example in Figure 14.

We again assume without loss of generality that d, $\overline{d}$, k, and $\overline{k}$ are fresh variables not occurring in Φ. Hence $T = (AS, \mathcal{K})$ with $AS = (\mathcal{L}, \overline{}, \mathcal{R}, \leq)$ and Q can be constructed in polynomial time w.r.t. Φ.

We argue that Φ is True if and only if d is out-relevant for t w.r.t. T.

- *From left to right.* Assume that Φ is True. Then there is an assignment τ'_X to variables of X such that for each assignment τ'_Y to variables of Y, $\phi[\tau'_X, \tau'_Y]$ is False. Let τ_X be such an assignment. Construct the knowledge base $\mathcal{K}' = \{k\} \cup \{x \in X \mid \tau_X[x] = \text{True}\} \cup \{\overline{x} \in X \mid \tau_X[x] = \text{False}\}$. Note that $\mathcal{K} \subseteq \mathcal{K}'$ and that $\mathcal{K}'$ is consistent since no $x \in X$ can be assigned both True and False by τ_X. Hence $T \sqsubseteq_Q (AS, \mathcal{K}')$. We observe the following
  - t is not stable-out w.r.t. $(AS, \mathcal{K}')$ and Q, since t is not out w.r.t. $(AS, \mathcal{K}')$. To see this, not that since $d \notin \mathcal{K}'$ and for each $y \in Y$ both $y \notin \mathcal{K}'$ and $\overline{y} \notin \mathcal{K}'$, there is no argument for t in $Arg_{(AS,\mathcal{K}')}$.
  - t is stable-out w.r.t. $(AS, \mathcal{K}' \cup \{d\})$ and Q. To see this, let $T'' = (AS, \mathcal{K}' \cup \{d\} \cup \mathcal{K}'')$ be an arbitrary AT such that $(AS, \mathcal{K}' \cup \{d\}) \sqsubseteq_Q T''$. Note that $\mathcal{K}'' \subseteq Y \cup \overline{Y}$. As there is no assignment τ_Y to variables in Y for which $\phi[\tau_X, \tau_Y]$ is True, there is no argument for t with top rule $(c_1, \ldots, c_p \Rightarrow t)$ in $Arg_{T''}$. On the other hand, there is an argument for t in $Arg_{T''}$ with top rule $(\overline{k} \Rightarrow t)$. Every argument for t in $Arg_{T''}$ must have $(\overline{k} \Rightarrow t)$ as its top rule and is therefore defeated by the observation-based (undefeated) argument k which must be in $G(T'')$. Since there is an argument for t in Arg_T and, furthermore, every argument for t in Arg_T is defeated by an argument in $G(T'')$, t is out w.r.t. T''. As T'' was chosen arbitrarily from all T''' such that $(AS, \mathcal{K}' \cup \{d\}) \sqsubseteq_Q T'''$, we have that t is stable-out w.r.t. $(AS, \mathcal{K}' \cup \{d\})$ and Q.

  Hence by Lemma 2 d is out-relevant for t w.r.t. T.
- *From right to left.* Assume that d is out-relevant for t w.r.t. T. Then by Definition 15 there is a minimal stable-out future theory $T' = (AS, \mathcal{K}' \cup \{d\})$ w.r.t. T and Q. Since t is stable-out w.r.t. $(AS, \mathcal{K}' \cup \{d\})$ and Q, t is out w.r.t. $(AS, \mathcal{K}' \cup \{d\})$. Hence there is an argument for t in $Arg_{T'}$ and, furthermore, each argument for t in $Arg_{T'}$ is defeated by an argument in $G(T')$. This implies that there is no argument with top rule $(c_1, \ldots, c_p \Rightarrow t)$, as such an argument for t would be undefeated.

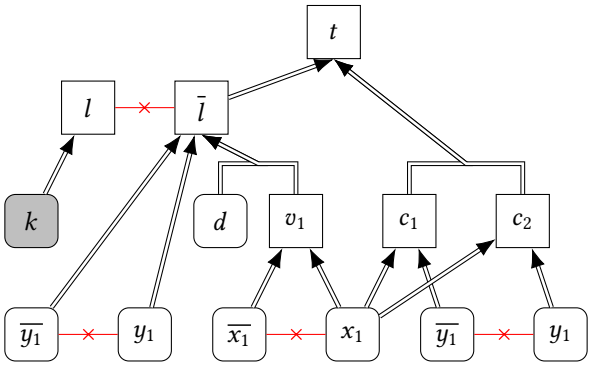

Fig. 15. Illustration of the reduction used in Theorem 2 for *blocked* status for the CNF formula $\phi = (x_1 \vee y_1) \wedge (x_i \vee \neg y_1)$. The queryables $\overline{y_1}$ and y_1 are shown twice for readability.

Further, by minimality of $(AS, \mathcal{K}' \cup \{d\})$, t cannot be stable-out w.r.t. $(AS, \mathcal{K}')$ and $\mathcal{Q}$. Hence there is an AT $(AS, \mathcal{K}'')$ such that $(AS, \mathcal{K}') \sqsubseteq_Q (AS, \mathcal{K}'')$ and t is not out w.r.t. $(AS, \mathcal{K}'')$. Note that there is no argument with top rule $(c_1, \ldots, c_p \Rightarrow t)$ in $Arg_{(AS,\mathcal{K}'')}$, as the existence of such an argument would imply that there would be an argument with top rule $(c_1, \ldots, c_p \Rightarrow t)$ in $Arg_{(AS,\mathcal{K}''\cup\{d\})}$, while $(AS, \mathcal{K}' \cup \{d\}) \sqsubseteq_Q (AS, \mathcal{K}'' \cup \{d\})$ and t is stable-out w.r.t. $(AS, \mathcal{K}' \cup \{d\})$. Therefore t is not defended w.r.t. $(AS, \mathcal{K}'')$. Moreover, t cannot be blocked w.r.t. $(AS, \mathcal{K}'')$ either, as there is no "equally strong" argument defeating any argument for t—the only way in which an argument for t can be defeated is by the observation-based argument k which must be in the grounded extension. This implies that t is unsatisfiable w.r.t. $(AS, \mathcal{K}'')$. Hence t is unsatisfiable w.r.t. $(AS, \mathcal{K}')$ as well. This implies that there is no argument for t, and hence there can be no argument with top rule $y \Rightarrow \overline{k}$ or $\overline{y} \Rightarrow \overline{k}$ in $Arg_{(AS,\mathcal{K}')}$ for any $y \in Y$. Consequently, for each $y \in Y$ we have $y \notin \mathcal{K}'$ and $\overline{y} \notin \mathcal{K}'$.

Since there is an argument for t in $Arg_{T'}$ and $T' = (AS, \mathcal{K}' \cup \{d\})$, the argument for t in $Arg_{(AS,\mathcal{K}'\cup\{d\})}$ are based on a set of rules including $(d, v_1, \ldots, v_n \Rightarrow \overline{k})$ and $(\overline{k} \Rightarrow t)$. This implies that for each $x \in X$ we have either $x \in \mathcal{K}'$ or $\overline{x} \in \mathcal{K}'$.

Now let τ_X be the assignment to the variables in X corresponding to $\mathcal{K}'$: for each $x \in X$, $\tau_X[x] = \text{True}$ if and only if $x \in \mathcal{K}'$ and $\tau_X[x] = \text{False}$ if and only if $\overline{x} \in \mathcal{K}'$. We argue that $\phi[\tau_X, \tau_Y]$ is False for every assignment τ_Y to variables in Y. Towards a contradiction, assume that there is an assignment τ_Y such that $\phi[\tau_X, \tau_Y]$ is True. Let $\mathcal{K}' \cup \{d\} \cup \mathcal{K}^*$ be the corresponding knowledge base: $\mathcal{K}^* = \{y \in Y \mid \tau_Y[y] = \text{True}\} \cup \{\overline{y} \in Y \mid \tau_Y[y] = \text{False}\}$. Since $\phi[\tau_X, \tau_Y]$ is True, there is an argument for t in $Arg_{(AS,\mathcal{K}'\cup\{d\}\cup\mathcal{K}^*)}$. This implies that t is defended w.r.t. $(AS, \mathcal{K}' \cup \{d\} \cup \mathcal{K}^*)$. However, then t is not stable-out w.r.t. $(AS, \mathcal{K}' \cup \{d\})$ and $\mathcal{Q}$; contradiction. Hence $\phi[\tau_X, \tau_Y]$ is False for every assignment τ_Y to variables in Y, that is, Φ is True. □

Proof of Theorem 2 for *blocked* status. Construct the AT T and queryables Q defined via

$$
\begin{aligned}
Q &= X \cup \overline{X} \cup Y \cup \overline{Y} \cup \{d, \overline{d}, k, \overline{k}\},\\
\mathcal{L} &= Q \cup C \cup \overline{C} \cup V \cup \overline{V} \cup \{t, \bar{t}, l, \bar{l}\},\\
\bar{\ } &= \{(x, \overline{x}), (\overline{x}, x) \mid x \in X \cup Y \cup V \cup C \cup \{d, t, k, l\}\},\\
\mathcal{R} &= \{(d, v_1, \ldots, v_n \Rightarrow \bar{l})\} \cup\\
&\quad \{(x \Rightarrow c) \mid x \in c\} \cup \{(\overline{x} \Rightarrow c) \mid \neg x \in c\} \cup\\
&\quad \{(y \Rightarrow c) \mid y \in c\} \cup \{(\overline{y} \Rightarrow c) \mid \neg y \in c\} \cup\\
&\quad \{(c_1, \ldots, c_p \Rightarrow t)\} \cup\\
&\quad \{(x_i \Rightarrow v_i), (\overline{x_i} \Rightarrow v_i) \mid x_i \in X\} \cup\\
&\quad \{(y \Rightarrow \bar{l}), (\overline{y} \Rightarrow \bar{l}) \mid y \in Y\} \cup\\
&\quad \{(k \Rightarrow l), (\bar{l} \Rightarrow t)\},\\
\mathcal{K} &= \{k\}\\
\leq &= \emptyset,
\end{aligned}
$$

with $C = \{c_1, \ldots, c_p\}$ the set of clauses in ϕ, $\overline{X} = \{\overline{x} \mid x \in X\}$, $\overline{Y} = \{\overline{y} \mid y \in Y\}$, $\overline{C} = \{\overline{c} \mid c \in C\}$, and $V = \{v_i \mid x_i \in X\}$ and $\overline{V} = \{\overline{v_i} \mid x_i \in X\}$. This reduction is illustrated by an example in Figure 15.

We yet again assume without loss of generality that d, $\overline{d}$, k, and $\overline{k}$ are fresh variables that do not occur in Φ and hence $T = (AS, \mathcal{K})$ with $AS = (\mathcal{L}, \bar{\ }, \mathcal{R}, \leq)$ and Q can be constructed in polynomial time w.r.t. Φ. We argue that Φ is True if and only if d is blocked-relevant for t w.r.t. T.

- *From left to right.* Assume that Φ is True. Then there is an assignment τ'_X to variables of X such that for each assignment τ'_Y to variables of Y, $\phi[\tau'_X, \tau'_Y]$ is False. Let τ_X be such an assignment. Construct the knowledge base $\mathcal{K}' = \{k\} \cup \{x \in X \mid \tau_X[x] = \text{True}\} \cup \{\overline{x} \in X \mid \tau_X[x] = \text{False}\}$. Note that $\mathcal{K} \subseteq \mathcal{K}'$ and that $\mathcal{K}'$ is be consistent as no $x \in X$ can be assigned both True and False by τ_X. Hence $T \sqsubseteq_Q (AS, \mathcal{K}')$. We make the following observations.
  - t is not stable-blocked w.r.t. $(AS, \mathcal{K}')$ and Q since t is not blocked w.r.t. $(AS, \mathcal{K}')$. To see this, note that since $d \notin \mathcal{K}'$ and for each $y \in Y$ both $y \notin \mathcal{K}'$ and $\overline{y} \notin \mathcal{K}'$, there is no argument for t in $Arg_{(AS, \mathcal{K}')}$.
  - t is stable-blocked w.r.t. $(AS, \mathcal{K}' \cup \{d\})$ and Q. To see this, let $T'' = (AS, \mathcal{K}' \cup \{d\} \cup \mathcal{K}'')$ be an arbitrary AT such that $(AS, \mathcal{K}' \cup \{d\}) \sqsubseteq_Q T''$. Note that $\mathcal{K}'' \subseteq Y \cup \overline{Y}$. As there is no assignment τ_Y to variables in Y such that $\phi[\tau_X, \tau_Y]$ is True, there is no argument for t with top rule $(c_1, \ldots, c_p \Rightarrow t)$ in $Arg_{T''}$. On the other hand, there is an argument for t in $Arg_{T''}$ with top rule $(\bar{l} \Rightarrow t)$. Every argument for t in $Arg_{T''}$ must have the rule $(\bar{l} \Rightarrow t)$ as its top rule and is therefore defeated by the argument with top rule $(k \Rightarrow l)$ which is itself defeated by all arguments for $\bar{l}$. Since there is an argument for t in Arg_T and, furthermore, every argument for t in Arg_T is defeated by an argument in $Arg_{T''}$ that is not in or defeated by any argument in $G(T'')$, it follows that t is blocked w.r.t. T''. As T'' was chosen arbitrarily from all T''' such that $(AS, \mathcal{K}' \cup \{d\}) \sqsubseteq_Q T'''$, we have that t is stable-blocked w.r.t. $(AS, \mathcal{K}' \cup \{d\})$ and Q.

  Hence by Lemma 2 d is blocked-relevant for t w.r.t. T.
- *From right to left.* Assume that d is blocked-relevant for t w.r.t. T. Then by Definition 15 there is a minimal stable-blocked future theory $T' = (AS, \mathcal{K}' \cup \{d\})$ w.r.t. T and Q. Since t is stable-blocked w.r.t. $(AS, \mathcal{K}' \cup \{d\})$ and Q, t is blocked w.r.t. $(AS, \mathcal{K}' \cup \{d\})$. Further, there is an argument for t in $Arg_{T'}$ and each argument

for t in $Arg_{T'}$ is defeated by an argument in $Arg_{T'}$. This implies that there is no argument with top rule $(c_1, \ldots, c_p \Rightarrow t)$, as such an argument for t would be undefeated.
By minimality of $(AS, \mathcal{K}' \cup \{d\})$, t cannot be stable-blocked w.r.t. $(AS, \mathcal{K}')$ and $\mathcal{Q}$. Hence there is an AT $(AS, \mathcal{K}'')$ such that $(AS, \mathcal{K}') \sqsubseteq_Q (AS, \mathcal{K}'')$ and t is not blocked w.r.t. $(AS, \mathcal{K}'')$. Note that there can be no argument with top rule $(c_1, \ldots, c_p \Rightarrow t)$ in $Arg_{(AS,\mathcal{K}'')}$, as the existence of such an argument would imply that there would be an argument with top rule $(c_1, \ldots, c_p \Rightarrow t)$ in $Arg_{(AS,\mathcal{K}''\cup\{d\})}$, while $(AS, \mathcal{K}' \cup \{d\}) \sqsubseteq_Q (AS, \mathcal{K}'' \cup \{d\})$ and t is stable-blocked w.r.t. $(AS, \mathcal{K}' \cup \{d\})$. Hence t is not defended w.r.t. $(AS, \mathcal{K}'')$. Moreover, t cannot be out w.r.t. $(AS, \mathcal{K}'')$, as there is no "stronger" argument defeating any argument for t. This implies that t is unsatisfiable w.r.t. $(AS, \mathcal{K}'')$, and hence t is unsatisfiable w.r.t. $(AS, \mathcal{K}')$. Hence there is no argument for t, and therefore there is no argument with top rule $y \Rightarrow \bar{l}$ or $\overline{y} \Rightarrow \bar{l}$ in $Arg_{(AS,\mathcal{K}')}$ for any $y \in Y$. Consequently $y \notin \mathcal{K}'$ and $\overline{y} \notin \mathcal{K}'$ for each $y \in Y$.
Since there is an argument for t in $Arg_{T'}$ and $T' = (AS, \mathcal{K}' \cup \{d\})$, the argument for t in $Arg_{(AS,\mathcal{K}'\cup\{d\})}$ is based on a set of rules including $(d, v_1, \ldots, v_n \Rightarrow \bar{l})$ and $(\bar{l} \Rightarrow t)$. This implies for each $x \in X$ $x \in \mathcal{K}'$ or $\overline{x} \in \mathcal{K}'$.
Now let τ_X be the assignment to the variables in X corresponding to $\mathcal{K}'$: for each $x \in X$, $\tau_X[x] =$ True if and only if $x \in \mathcal{K}'$ and $\tau_X[x] =$ False if and only if $\overline{x} \in \mathcal{K}'$. We argue that $\phi[\tau_X, \tau_Y]$ is False for ever assignment τ_Y to the variables in Y. Towards a contradiction, assume that there is an assignment τ_Y such that $\phi[\tau_X, \tau_Y]$ is True. Let $\mathcal{K}' \cup \{d\} \cup \mathcal{K}^*$ be the corresponding knowledge base: $\mathcal{K}^* = \{y \in Y \mid \tau_Y[y] =$ True$\} \cup \{\overline{y} \in Y \mid \tau_Y[y] =$ False$\}$. As$\phi[\tau_X, \tau_Y]$ is True, there is an argument for t in $Arg_{(AS,\mathcal{K}'\cup\{d\}\cup\mathcal{K}^*)}$. This implies that t is defended w.r.t. $(AS, \mathcal{K}' \cup \{d\} \cup \mathcal{K}^*)$. However, then t would not be stable-blocked w.r.t. $(AS, \mathcal{K}' \cup \{d\})$ and $\mathcal{Q}$; contradiction. Hence $\phi[\tau_X, \tau_Y]$ is False for every assignment τ_Y to that variables in Y, that is, Φ is True. □

Table 4. Number of solved instances, mean run times over solved instances, and maximum run times for deciding blocked-relevance of a single queryable.

		#solved (mean/max run time (s))			
Dataset	$\lvert\mathcal{L}\rvert$	ASP no prefs		ASP under prefs	
Real	60	351	(0.2/4.2)	351	(0.4/12.6)
Synthetic	50	75	(0.1/0.1)	75	(0.2/0.3)
	60	75	(0.4/1.1)	75	(0.5/3.9)
	70	75	(0.6/7.2)	75	(1.4/13.4)
	80	75	(1.5/16.1)	75	(6.5/79.5)
	90	75	(3.4/78.0)	75	(11.5/479.3)
	100	71	(17.8/-)	72	(30.6/-)
	110	69	(21.5/-)	71	(68.5/-)
	120	68	(60.6/-)	72	(192.6/-)
	130	68	(208.0/-)	48	(380.1/-)
	140	3	(0.1/-)	1	(1.1/-)
	150	1	(0.1/-)	0	(-/-)

Table 5. Number of solved instances, mean run times over solved instances, and maximum run times for deciding out-relevance of a single queryable.

		#solved (mean/maximum run time (s))			
Dataset	$\lvert\mathcal{L}\rvert$	ASP no prefs		ASP under prefs	
Real	60	351	(0.2/1.8)	351	(0.3/2.7)
Synthetic	50	75	(0.1/0.2)	75	(0.2/0.7)
	60	75	(0.3/1.8)	75	(1.3/12.1)
	70	75	(0.5/6.0)	75	(2.0/30.6)
	80	75	(1.4/16.1)	75	(7.4/124.9)
	90	75	(12.3/153.0)	75	(48.6/560.6)
	100	68	(54.9/-)	60	(55.1/-)
	110	61	(30.7/-)	59	(59.5/-)
	120	66	(68.7/-)	64	(185.6/-)
	130	53	(192.4/-)	43	(341.8/-)
	140	0	(-/-)	2	(0.5/-)
	150	4	(0.1/-)	4	(0.7/-)

Table 6. Number of solved instances, mean run times over solved instances, and maximum run times for deciding unsatisfiable-relevance of a single queryable.

		#solved (mean/maximum run time (s))			
Dataset	$\lvert\mathcal{L}\rvert$	ASP no prefs		ASP under prefs	
Real	60	351	(8.8/454.8)	351	(15.0/538.3)
Synthetic	50	75	(0.1/0.3)	75	(0.2/1.0)
	60	75	(0.4/2.0)	75	(1.1/5.2)
	70	75	(1.3/6.7)	75	(5.1/28.3)
	80	75	(2.1/29.1)	75	(8.2/111.7)
	90	75	(15.4/158.3)	74	(57.5/-)
	100	54	(63.4/-)	49	(64.2/-)
	110	56	(13.2/-)	55	(45.6/-)
	120	55	(45.8/-)	52	(105.0/-)
	130	49	(124.1/-)	32	(175.1/-)
	140	22	(1.7/-)	22	(4.9/-)
	150	22	(0.1/-)	22	(0.4/-)

B Additional Empirical Data

Complementing the empirical results presented in the main text for the problem of deciding j-relevance of a given single queryable for the *defended* status, the analogous data from the experiments for the *blocked*, *out* and *unsatisfiable* statuses are shown in Table 4, Table 5 and Table 6, respectively.

Received 21 February 2025; accepted 20 June 2025