Artificial Intelligence 333 (2024) 104159

Contents lists available at ScienceDirect

Artificial Intelligence

journal homepage: www.elsevier.com/locate/artint

Credulous acceptance in high-order argumentation frameworks with necessities: An incremental approach

Gianvincenzo Alfano [a], Andrea Cohen [b,*], Sebastian Gottifredi [b], Sergio Greco [a], Francesco Parisi [a], Guillermo R. Simari [b]

[a] *Department of Informatics, Modeling, Electronics and System Engineering, University of Calabria, Italy*

[b] *Department of Computer Science and Engineering, Universidad Nacional del Sur (UNS) & Institute for Computer Science and Engineering (CONICET-UNS), Argentina*

ARTICLE INFO

Keywords:
Formal argumentation
Abstract argumentation
High-order interactions
Credulous acceptance
Argumentation dynamics
Incremental computation

ABSTRACT

Argumentation is an important research area in the field of AI. There is a substantial amount of work on different aspects of Dung's abstract Argumentation Framework (AF). Two relevant aspects considered separately so far are: i) extending the framework to account for recursive attacks and supports, and ii) considering dynamics, *i.e.*, AFs evolving over time. In this paper, we jointly deal with these two aspects. We focus on High-Order Argumentation Frameworks with Necessities (HOAFNs) which allow for attack and support relations (interpreted as *necessity*) not only between arguments but also targeting attacks and supports at any level. We propose an approach for the incremental evaluation of the credulous acceptance problem in HOAFNs, by "incrementally" computing an extension (a set of accepted arguments, attacks and supports), if it exists, containing a given goal element in an updated HOAFN. In particular, we are interested in monitoring the credulous acceptance of a given argument, attack or support (goal) in an evolving HOAFN. Thus, our approach assumes to have a HOAFN Δ, a goal ϱ occurring in Δ, an extension E for Δ containing ϱ, and an update u establishing some changes in the original HOAFN, and uses the extension for first checking whether the update is relevant; for relevant updates, an extension of the updated HOAFN containing the goal is computed by translating the problem to the AF domain and leveraging on AF solvers. We provide formal results for our incremental approach and empirically show that it outperforms the evaluation from scratch of the credulous acceptance problem for an updated HOAFN.

1. Introduction

Argumentation has become an attractive and effective paradigm for knowledge representation and reasoning within the field of Artificial Intelligence [1–4]. Most of the situations where argumentation takes place are inherently characterized by the presence of controversial information. Enabling automated systems to process such kind of information, much in the same way as organized human discussions are carried out, is an important challenge that has deserved increasing attention from the Artificial Intelligence community in the last decades. This has led to the development of an important and active research area called *formal argumentation* [5] that has been explored in several application contexts, e.g., legal reasoning [6], decision support systems [7],

* Corresponding author.
E-mail address: ac@cs.uns.edu.ar (A. Cohen).

https://doi.org/10.1016/j.artint.2024.104159
Received 19 July 2023; Received in revised form 7 February 2024; Accepted 18 May 2024

Available online 22 May 2024

E-Democracy [8], healthcare [9], medical applications [10], financial analysis [11], explanation of results [12–14], as well as multi-agent systems and social networks [15].

A powerful formalism in the above-mentioned research area is Dung's abstract Argumentation Framework (AF) [16]. The meaning of an AF is given in terms of argumentation semantics, e.g. the well-known *grounded* (gr), *complete* (co), *preferred* (pr), and *stable* (st) semantics, which intuitively tell us the sets of arguments (called s-*extensions*, with $s \in \{gr, co, pr, st\}$) that can be collectively accepted to sustain a point of view in a dispute. For instance, for an AF $\langle A, R\rangle = \langle\{\mathtt{a}, \mathtt{b}\}, \{(\mathtt{a}, \mathtt{b}), (\mathtt{b}, \mathtt{a})\}\rangle$ having two arguments, a and b, attacking each other, there are two preferred (also, stable) extensions: {a} and {b}. A well-studied reasoning problem is the *credulous acceptance* problem [17], denoted as CA. For a semantics $s \in \{gr, co, pr, st\}$, given an AF and a goal argument g, CA_s consists in deciding whether there exists an s-extension of the given AF containing g.

Following the work by Dung, there have been many extensions of AFs allowing for bipolar interactions [18], second-order attacks [19] or, more generally, recursive attacks [20]. The importance of having frameworks incorporating HOAFN features has been highlighted in the literature, as they contribute to addressing several kinds of real world problems. On the one hand, concerning support, the experimental study reported in [21] shows that the notion of support is important for expressing relations between the statements involved in dialogs. Similarly, work relying on argumentation for reasoning with medical knowledge suggests that, when evaluating the available medical evidence, attacks are not the only interactions that arise as there are practical situations where support relations are needed [22,23]. On the other hand, concerning high-order interactions, [24] shows how recursive interactions can be used to model and reasoning with conflicting opinions about the credibility of informant agents in an open multi-agent setting. Moreover, frameworks combining higher-order and bipolar interactions have been exploited in Biology for reasoning over metabolic networks; specifically, using their logical encodings for modeling the behavior of molecular interaction maps [25]. More in general, the interest in studying and using such kind of frameworks has increased over the last years, as observed in [26].

In line with the approaches discussed above, in this paper we propose to work with a High-Order Argumentation Framework with Necessities (HOAFN) which allows for attacks and supports not only towards arguments (called first-level interactions), but also targeting the attack and support relations at any level. In particular, a HOAFN will be interpreted under two semantics, leading to an Attack-Support Argumentation Framework (ASAF) [27] and a Recursive Argumentation Frameworks with Necessities (RAFN) [28], respectively. From a representational point of view a HOAFN has the advantage of enabling a straightforward representation of reasoning situations (*e.g.* in the areas discussed above and, more generally, for modeling decision processes) in a way that is more intuitive for humans than what one would get using other frameworks such as Dung's framework or flat Bipolar AFs. Moreover, the HOAFN provides a more declarative representation of the arguments and interactions under discussion than what would be obtained using one of the other frameworks. This is because, for instance, using a Dung's AF would require to create new auxiliary meta-arguments and interactions in order to model the same knowledge as with the HOAFN; this will become evident later in the paper, when we present how to translate a HOAFN into an AF.

The support relation in a HOAFN is interpreted as *necessity* [29]: if argument a supports argument b, then the acceptance of a is necessary to get the acceptance of b; equivalently, the acceptance of b implies the acceptance of a. However, it is worth mentioning that a support in a HOAFN can also target an attack or another support. In particular, a HOAFN can be represented by a graph-like structure, where arguments are denoted by nodes whose names are written with typewriter font, attacks are denoted by single arrows (→), and supports are denoted by double arrows (⇒). Also, attacks and supports are respectively labeled with Greek letters α and β, possibly with subscripts. An example of a HOAFN is given next.

Example 1. Consider the following scenario which can be synthesized by the HOAFN of Fig. 1 (left). Suppose John is planning to spend his winter holidays in Bariloche and has to decide whether to rent a car to drive during his stay (d) or make use of public transportation (pt). Arguments d and pt are two alternatives John has, and the conflict between them is represented by the attacks α_2 and α_3. In general, John has a preference towards driving over using public transport (p). This is encoded by the attack α_4 from p to α_3. However, he has been told that in order to drive safely in Bariloche he needs to put a snow traction device on the car (st), but this is only required during winter, the current season in Bariloche (w). Hence, st supports d (support β_0) as the acceptance of st is necessary for the acceptance of d, and w supports β_0 (support β_1). In addition, it may be the case that rental car services ran out of snow traction devices (rn). The mutual attacks α_0 and α_1 model such a situation: if the rental car services have no snow traction devices left, then John will be unable to use such devices on the car, whereas if John uses such devices it will be because they are available in the rental car services. As it will be clear in what follows (Example 8), in this context there are three complete extensions: one of them states that John will end up deciding to use public transportation in Bariloche (*i.e.* it contains argument pt), another one states that John will choose to drive (*i.e.* it contains argument d), and the other one takes no position in the discussion (*i.e.* neither pt nor d belong to the extension). □

However, in practice, argumentation frameworks are dynamic systems [30–38]. In fact, a HOAFN typically represents a temporary situation, and arguments, attacks and supports can be added or removed to take into account new available knowledge.

Example 2. Continuing with our running example, suppose now that John comes across with new information stating that it has not snowed in Bariloche for the last two months and will not snow during the time of his stay, represented by an argument ns. Then, when considering the previous arguments and attacks together with the new argument, a new conflict arises, needing to update John's knowledge: argument ns attacks β_0, the support from st to d, as it provides a context in which John will not need to put a snow traction device on his car in order to drive safely in Bariloche. The new scenario, with the addition of argument ns and then, of the attack α_5, corresponds to the updated HOAFN shown in Fig. 1 (right). As will be shown in Example 9, in this updated

Fig. 1. HOAFNs discussed in the Introduction.

context all (three) complete extensions contain argument d, prescribing that John will choose to rent and drive a car during his stay in Bariloche. □

Recently, there has been a growing interest in studying dynamics of different argumentation systems, including in the context of Dung's framework [39,40,33,41–43], Bipolar AFs and AFs with second-order attacks [44,45]. Notwithstanding this, none of the developments regarding dynamics in argumentation has so far considered bipolar recursive frameworks like the HOAFN. It should be noted that incremental computation techniques could improve performance, as they only require to reconsider the acceptance status of those arguments and interactions that are "affected" by the new information. For instance, in our running example, the acceptance status of rn and st, as well as that of α_0 and α_1, does not change after adding the attack α_5 from ns to the support β_0, while the acceptance statuses of other arguments, attacks and supports changes, yielding a different outcome.

With the aim of exploiting features like those mentioned above, in this paper we propose an incremental approach for efficiently solving the credulous acceptance problem defined as follows: given a HOAFN Δ, a goal element ϱ, a semantics $\mathcal{s}$, an $\mathcal{s}$-extension E for Δ containing ϱ and an update u for Δ, determine whether there exists an $\mathcal{s}$-extension $\widehat{E}$ for the updated HOAFN $u(\Delta)$ containing ϱ.

As most of the approaches for solving the credulous acceptance problem in AF contexts, we also compute an extension containing the goal ϱ (if there exists one) [46–48], in order to benefit from it in subsequent updates by performing an "incremental" computation.

Contributions. Our contributions can be summarized as follows:

1. We start by defining the class of HOAFNs that provides a uniform syntax for ASAFs and RAFNs[1] and accounts for their acceptability semantics, specifically considering the grounded, complete, preferred and stable semantics.
2. We characterize *updates* for a HOAFN to formally address dynamics in the context of this framework. We distinguish between elementary updates, which correspond to the addition or deletion of a single element, and (complex) updates which may involve the addition or deletion of multiple attacks and supports.
3. We identify and formally characterize *irrelevant updates* for a HOAFN (under the ASAF or RAFN semantics), for which the credulous acceptance of a given goal element (as well as an extension containing it) w.r.t. an updated HOAFN $u(\Delta)$ can be easily determined with reduced computational cost, under the grounded, complete, preferred, and stable semantics. These results can be used to avoid wasted effort for any incremental algorithm, not just the one proposed in this paper.
4. We characterize a HOAFN in terms of an AF (under the ASAF or RAFN semantics), and formally show that this AF yields *equivalent extensions* to those of the HOAFN, under the grounded, complete, preferred, and stable semantics. It is worth noting that this enables a computational strategy for determining credulous acceptance in a HOAFN, as well as for HOAFN's extensions computation, even in the static case (*i.e.*, without considering updates).
5. We define an incremental algorithm for computing the credulous acceptance w.r.t. an updated HOAFN $u(\Delta)$, accounting for early termination conditions implied by irrelevant updates and leveraging on the incremental technique proposed in [49,50] for the computation on the (extensions-equivalent) AF for the updated HOAFN $u(\Delta)$. The technique is able to incorporate any existing AF-solver to perform the incremental computation of credulous acceptance (and of a HOAFN's extension as a byproduct) under the grounded, complete, preferred, and stable semantics.
6. With the aim of evaluating the performance of our incremental approach, we propose a parametric dataset generator for HOAFNs that builds frameworks with specific ratios of attacks and supports at any given interaction level. Also in this case, the generator could be used in other contexts, including static scenarios where no updates are considered.
7. We perform an empirical analysis showing that our incremental approach generally outperforms the computation from scratch of credulous acceptance w.r.t. an updated HOAFN, where the fastest solvers from the fourth edition of the International Competition on Computational Models of Argumentation (ICCMA 2021)[2] are used as baseline. In fact, the experiments show that, on average, the incremental computation can be more than 100 times faster than the computation from scratch, for both ASAF and RAFN, under the grounded, complete, preferred, and stable semantics.

2. Preliminaries

In this section we will provide the essential background for Dung's abstract Argumentation Framework (AF) [16], and introduce the High-Order Argumentation Framework with Necessities (HOAFN) as well as the semantics corresponding to two alternative

[1] A restricted class of RAFNs is considered, to align with the definition of ASAFs, as shown in Section 2.3.2.

[2] http://argumentationcompetition.org.

approaches, namely the Attack-Support Argumentation Framework (ASAF) [27] and the Recursive Argumentation Framework with Necessities (RAFN) [28]. For a better understanding of the HOAFN, we briefly recall Bipolar Argumentation Frameworks where (first-level) supports are introduced. We also discuss an incremental computation technique for AFs.

To ease readability, a summary of the main notations used in the paper is reported in Table A.3 (in Appendix A).

2.1. Abstract argumentation framework

An AF consists of a set of arguments whose origin is left unspecified, and a set of conflicts between them [16].

Definition 1 *(AF).* An *abstract Argumentation Framework (AF)* is a pair $\langle \mathsf{A}, \mathsf{R}\rangle$, where A is a set of arguments and $\mathsf{R} \subseteq \mathsf{A} \times \mathsf{A}$ is an attack relation.

In [16] some semantic notions are defined leading to the characterization of collectively acceptable sets of arguments, called *extensions*. Given an AF $\langle \mathsf{A}, \mathsf{R}\rangle$ and a set $S \subseteq \mathsf{A}$ of arguments, an argument $a \in \mathsf{A}$ is said to be *i*) *defeated* w.r.t. S iff $\exists b \in S$ such that $(b, a) \in \mathsf{R}$, and *ii*) *acceptable* w.r.t. S iff for every argument $b \in \mathsf{A}$ with $(b, a) \in \mathsf{R}$, there is $c \in S$ such that $(c, b) \in \mathsf{R}$. The sets of defeated and acceptable arguments w.r.t. S are defined as follows (where the AF is understood).

Definition 2 *(Defeated and Acceptable Sets for AF).* Let $\langle \mathsf{A}, \mathsf{R}\rangle$ be an AF and $S \subseteq \mathsf{A}$:

- $\mathbf{def}(S) = \{a \in \mathsf{A} \mid \exists\, b \in S\,.\,(b, a) \in \mathsf{R}\}$;
- $\mathbf{acc}(S) = \{a \in \mathsf{A} \mid \forall\, b \in \mathsf{A}\,.\,(b, a) \in \mathsf{R} \Rightarrow b \in \mathbf{def}(S)\}$.

Then, the complete (co), preferred (pr), stable (st), and grounded (gr) extensions of an AF are defined as follows.

Definition 3 *(AF semantic notions and extensions).* Let $\langle \mathsf{A}, \mathsf{R}\rangle$ be an AF and $S \subseteq \mathsf{A}$. We say that S is:

- *conflict-free* iff $S \cap \mathbf{def}(S) = \emptyset$;
- *admissible* iff it is conflict-free and $S \subseteq \mathbf{acc}(S)$;
- a *complete extension* (co) iff it is conflict-free and $S = \mathbf{acc}(S)$;
- a *preferred extension* (pr) iff it is a maximal (w.r.t. $\subseteq$) complete extension;
- a *stable extension* (st) iff it is a preferred extension such that $S \cup \mathbf{def}(S) = \mathsf{A}$;
- the *grounded extension* (gr) iff it is the smallest (w.r.t. $\subseteq$) complete extension.

We use $\mathcal{E}_s(\langle \mathsf{A}, \mathsf{R}\rangle)$ to denote the set of s- extensions of an AF $\langle \mathsf{A}, \mathsf{R}\rangle$. Hereafter, if not explicitly stated, we denote with s any of the semantics in $\{co, pr, st, gr\}$. Given an AF, an argument a, and a semantics s, the *credulous (resp. skeptical) acceptance* problem, denoted as CA_s (resp. SA_s), is the problem of deciding whether argument a is credulously (resp. skeptically) accepted under semantics s, that is, deciding whether a belongs to at least one (resp. all) s-extension of the framework. Clearly, for the grounded semantics, which admits exactly one extension, these problems become identical.

2.2. Bipolar argumentation framework

Dung's framework has been extended in many different ways, including the introduction of new kinds of interactions between arguments and/or attacks. In particular, the class of Bipolar Argumentation Frameworks (BAFs) is an interesting extension of the AF which also allows to model the *support* between arguments [29,51].

Definition 4. A *Bipolar Argumentation Framework* (BAF) is a triple $\langle \mathsf{A}, \mathsf{R}, \mathsf{T}\rangle$, where A is a set of *arguments*, $\mathsf{R} \subseteq \mathsf{A} \times \mathsf{A}$ is a set of *attacks*, and $\mathsf{T} \subseteq \mathsf{A} \times \mathsf{A}$ is a set of *supports*.

A BAF can be represented by a directed graph with two types of edges: *attacks* and *supports*, denoted by $\rightarrow$ and $\Rightarrow$, respectively. A *support path* $a_0 \overset{+}{\Rightarrow} a_n$ from argument a_0 to argument a_n is a sequence of n edges $a_{i-1} \Rightarrow a_i$ with $0 < i \leq n$. For any binary relation B, we use B^+ to denote the transitive closure of B.

In this paper we focus on the "necessary" interpretation of supports. The necessary interpretation of a support $a \Rightarrow b$ is that the acceptance of b implies the acceptance of a and, conversely, the non-acceptance of a implies the non-acceptance of b. In this context, a BAF is also referred to as an AFN (AF with Necessities). In this regard, we observe that in AFNs no restriction is imposed on the disjointedness of the attack and support relations, that is, it may be the case that $\mathsf{R} \cap \mathsf{T} \neq \emptyset$. Although the aforementioned restriction is in line with the idea that, in an argumentation setting, it does not seem rational to put forward an argument which simultaneously attacks and supports another element (see [4, Ch. 4]), to keep the HOAFN general so that such situations are allowed but can be disregarded as needed, we do not impose such restriction in the following—our results still hold if we assume that the attack and support relations are disjoint.

Fig. 2. (From left to right) AFN from Example 3 and its equivalent AF.

Hereafter it is assumed that the support relation is acyclic. The reason for this assumption is twofold. First, as our attention is put on AFNs, we highlight that in [29] support cycles are syntactically admitted but semantically excluded through the concept of necessity-cycle freeness. This syntactic restriction is also adopted by the Higher-Order Argumentation Framework with Necessities extending the AFN, presented in the next section. Second, as mentioned in [26], the authors in [52] state that the irreflexive and transitive nature of supports excludes any risk to have a cycle of necessities, which are undesirable because they correspond to a kind of fallacy (begging the question). Although some semantics for AFNs with a cyclic support relation have been defined in the literature [29,53], they are essentially based on avoiding considering the contribution of arguments occurring in support-cycles or (transitively) supported by arguments in support-cycles. Thus, in the following, we decide to concentrate on AFNs without support-cycles.

It has been shown that AFNs without support-cycles can be easily rewritten into 'equivalent' AFs or 'equivalent' logic programs under partial stable model semantics [29,54] (recently, this equivalence has been shown also for the case of AFNs with support-cycles [53]). In particular, the semantics of an (acyclic) AFN in terms of an equivalent AF is obtained by removing the supports and augmenting the attack relation with the so-called *extended* attacks. Given an AFN $\langle \mathsf{A}, \mathsf{R}, \mathsf{T} \rangle$, there exists an *extended attack* from a to b if there exists an attack $a \rightarrow c$ and a support path $c \overset{+}{\Rightarrow} b$.[3]

Example 3. Consider the AFN $\langle \mathsf{A}, \mathsf{R}, \mathsf{T} \rangle$ reported in Fig. 2 (left), where $\mathsf{A} = \{\mathtt{a}, \mathtt{b}, \mathtt{c}, \mathtt{d}\}$, $\mathsf{R} = \{(\mathtt{a}, \mathtt{b}), (\mathtt{c}, \mathtt{d})\}$ and $\mathsf{T} = \{(\mathtt{b}, \mathtt{c})\}$. Under the necessary interpretation of support there is an extended attack $(\mathtt{a}, \mathtt{c})$ generating an equivalent AF $\langle \mathsf{A}, \mathsf{R}' \rangle$ reported in Fig. 2 (right), with $\mathsf{R}' = \{(\mathtt{a}, \mathtt{b}), (\mathtt{c}, \mathtt{d}), (\mathtt{a}, \mathtt{c})\}$, whose unique complete extension is $\{\mathtt{a}, \mathtt{d}\}$. $\square$

2.3. *High-order argumentation framework with necessities*

Further extensions of Dung's framework consider second-order interactions [51] (*e.g.*, attacks to attacks or supports), as well as more general forms of interactions such as recursive AFs where attacks can be recursively attacked [20,56] and recursive BAFs, where attacks or supports can be recursively attacked or supported [27,28] (see also [26] for an overview of all these kinds of frameworks).

In this section we introduce the definition of a High-Order Argumentation Framework with Necessities (HOAFN). HOAFN frameworks extend AFNs (and therefore AFs) by incorporating bipolar high-order interactions. In that way, a HOAFN allows for the representation and reasoning with attack and support relations not only between arguments, but also targeting the attack and support relations themselves.

The attacks and supports in a HOAFN will be identified by their names. The source of an attack or a support will always be an argument whereas the target can be an argument, an attack or a support.

We assume the existence of universal finite pairwise disjoint sets of arguments ($\mathbb{A}_{\mathbb{U}}$), attack names ($\mathbb{R}_{\mathbb{U}}$) and support names ($\mathbb{T}_{\mathbb{U}}$). These sets respectively contain every conceivable argument, attack name and support name that may appear in a HOAFN.[4] We will also assume the existence of the total functions $\mathbf{s} : \mathbb{R}_{\mathbb{U}} \cup \mathbb{T}_{\mathbb{U}} \mapsto \mathbb{A}_{\mathbb{U}}$, $\mathbf{t} : \mathbb{R}_{\mathbb{U}} \cup \mathbb{T}_{\mathbb{U}} \mapsto \mathbb{A}_{\mathbb{U}} \cup \mathbb{R}_{\mathbb{U}} \cup \mathbb{T}_{\mathbb{U}}$ and $\mathbf{b} : \mathbb{R}_{\mathbb{U}} \cup \mathbb{T}_{\mathbb{U}} \mapsto \mathbb{A}_{\mathbb{U}} \times (\mathbb{A}_{\mathbb{U}} \cup \mathbb{R}_{\mathbb{U}} \cup \mathbb{T}_{\mathbb{U}})$ mapping each attack and support to its source, target and pair (source, target), respectively. For any set of attacks and supports S, we shall also use the notations $\mathbf{s}(S) = \{\mathbf{s}(\gamma) \mid \gamma \in S\}$, $\mathbf{t}(S) = \{\mathbf{t}(\gamma) \mid \gamma \in S\}$ and $\mathbf{b}(S) = \{\mathbf{b}(\gamma) \mid \gamma \in S\}$ to denote the sets of source, target and pair (source, target) of the elements in S, respectively. To simplify, with a little abuse of notation, we will write $\mathbf{s}(\gamma)$, $\mathbf{t}(\gamma)$ and $\mathbf{b}(\gamma)$ (instead of $\mathbf{s}(\{\gamma\})$, $\mathbf{t}(\{\gamma\})$ and $\mathbf{b}(\{\gamma\})$, respectively) for singletons.

Definition 5. A High-Order Argumentation Framework with Necessities (HOAFN) is a tuple $\langle \mathbb{A}, \mathbb{R}, \mathbb{T} \rangle$, where $\mathbb{A} \subseteq \mathbb{A}_{\mathbb{U}}$ is a finite set of arguments, $\mathbb{R} \subseteq \mathbb{R}_{\mathbb{U}}$ is a finite set of attacks, $\mathbb{T} \subseteq \mathbb{T}_{\mathbb{U}}$ is a finite set of (necessary) supports and $\mathbf{b}(\mathbb{R} \cup \mathbb{T}) \subseteq \mathbb{A} \times (\mathbb{A} \cup \mathbb{R} \cup \mathbb{T})$.

Note that the source and target functions $\mathbf{s}(\cdot)$ and $\mathbf{t}(\cdot)$ are defined over the universal sets of arguments, attacks and supports that may appear in any HOAFN. Given that, the preceding definition imposes constraints on the sets $\mathbb{R}$ and $\mathbb{T}$ to prevent attacks and supports from originating in an argument (respectively, from targeting an argument or an attack or support) that does not appear in the framework.

Given a HOAFN $\Delta = \langle \mathbb{A}, \mathbb{R}, \mathbb{T} \rangle$, hereafter we will often refer to a generic element of Δ (*i.e.* either an argument, an attack or a support name) with ϑ, to either an attack or a support name with γ, and to an attack (resp. support) name with α (resp. β), possibly with subscripts. Moreover, we associate to all attacks and supports a unique natural number > 0, called *level*, defined as follows. For any attack or support γ, $level(\gamma)$ is equal to:

[3] A second kind of extended attack from a to b is proposed in the literature for AFNs, covering the case where $c \overset{+}{\Rightarrow} a$ and $c \rightarrow b$. However, it was shown in [55] that this kind of attack is somewhat redundant, and does not affect the semantics' outcome.

[4] As will be shown in Section 4, the universal sets will serve to define the updates that can be performed over a HOAFN.

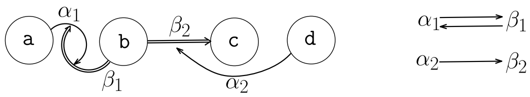


Fig. 3. HOAFN from Example 4 (left) and its corresponding directed graph used to define the $level$ function (right).

- 1 if *i)* $\mathbf{t}(\gamma)$ is an argument or *ii)* γ occurs in a cycle of the directed graph whose nodes are the attack and support names of the HOAFN and there exists an edge (γ_1, γ_2) in the graph for every pair of attack/support names γ_1 and γ_2 such that $\mathbf{t}(\gamma_1) = \gamma_2$ in the HOAFN;
- $level(\mathbf{t}(\gamma)) + 1$, otherwise.

Example 4. Consider the HOAFN represented in Fig. 3 (left). We have that $level(\beta_2) = 1$ since the target of β_2 is an argument (namely, $\mathbf{t}(\beta_2) = \mathtt{c}$), while $level(\alpha_2) = level(\beta_2) + 1 = 2$. Moreover, $level(\alpha_1) = level(\beta_1) = 1$ as both α_1 and β_1 belong to a cycle in the directed graph used to define the $level$ function, as shown in Fig. 3 (right). □

Regarding the semantics of HOAFN frameworks, two main different approaches have been proposed: *Attack-Support Argumentation Framework (ASAF)* [27] and *Recursive Argumentation Framework with Necessities (RAFN)* [28]. As stated before, we will consider the complete, preferred, stable and grounded semantics for HOAFNs, adopting both the ASAF and the RAFN approach.[5] Also, similarly to the notation for AFs, given a HOAFN Δ and a semantics $s \in \{co, pr, st, gr\}$, we will use $\mathcal{E}_s(\Delta)$ to denote the set of s-extensions of Δ.

Since the HOAFN aims at capturing the behavior of the AFN and the ASAF, we assume that the support relation is acyclic as in the latter. In other words, the support relation of the resulting AFN, which can be obtained by replacing every attack $a \to b$ with $a \Rightarrow \alpha$ and $\alpha \to b$, and replacing every support $a \Rightarrow b$ with $a \Rightarrow \beta$ and $\beta \Rightarrow b$, is acyclic. It is worth noting that the so-obtained auxiliary graph is only used to formally define and check acyclicity in HOAFNs, not to provide the semantics which are instead given in the subsequent sections. Nevertheless, we believe that it would be interesting to investigate cyclic supports, possibly by also changing the original semantics of AFN and ASAF, as initially done in [57] for evidential supports and subsequently in [58] for RAFN—we defer the investigation of forms of cyclic supports to future work.

2.3.1. Attack-support argumentation framework

The *Attack-Support Argumentation Framework (ASAF)* has been proposed in [59,27]. The ASAF combines the AFRA [60] interpretation of attacks with that of BAF under the necessary interpretation of supports (*i.e.* AFN [29]). In this section we will introduce the characterization of defeats on arguments, attacks and supports for the ASAF. It is worth noting that we do not introduce a specific syntax for ASAF, as we rely on the syntax introduced for HOAFN by treating the ASAF as a particular case of the HOAFN (the same holds for the RAFN, introduced in the next section). We also provide some basic semantic notions that will be useful for defining the semantics of the framework in terms of its extensions.

Since the ASAF combines the AFRA and the AFN, defeats in the ASAF are defined by combining direct and indirect defeats of the AFRA [20, Def. 5] with the extended defeats of the AFN introduced in Section 2.2. Formally, we can give the following definition.

Definition 6 *(ASAF defeats).* Let $\langle \mathbb{A}, \mathbb{R}, \mathbb{T} \rangle$ be an ASAF, $\vartheta \in (\mathbb{A} \cup \mathbb{R} \cup \mathbb{T})$, $\alpha \in \mathbb{R}$, and $S \subseteq (\mathbb{A} \cup \mathbb{R} \cup \mathbb{T})$. We say that α *defeats* ϑ *given* S (denoted as $\alpha \ \mathbb{def}_S \ \vartheta$) if either: *i)* $\mathbf{t}(\alpha) = \vartheta$, *ii)* $\vartheta \in \mathbb{R}$ and $\mathbf{t}(\alpha) = \mathbf{s}(\vartheta)$, *iii)* there exists $b \in \mathbb{A}$ such that $\mathbf{t}(\alpha) = b$ and $(b, \vartheta) \in \mathbf{b}(\mathbb{T} \cap S)^+$, or *iv)* $\vartheta \in \mathbb{R}$ and there exists $b \in \mathbb{A}$ such that $\mathbf{t}(\alpha) = b$ and $(b, \mathbf{s}(\vartheta)) \in \mathbf{b}(\mathbb{T} \cap S)^+$.

For any ASAF and a set of arguments, attacks and supports, the *defeated* and *acceptable* sets (with respect to a given set **S**) are defined as follows:

Definition 7 *(ASAF defeated and acceptable sets).* Let $\langle \mathbb{A}, \mathbb{R}, \mathbb{T} \rangle$ be an ASAF and $S \subseteq (\mathbb{A} \cup \mathbb{R} \cup \mathbb{T})$. We define:

- $\mathbf{def}(S) = \{\vartheta \in (\mathbb{A} \cup \mathbb{R} \cup \mathbb{T}) \mid \exists\, \alpha \in (\mathbb{R} \cap S) \,.\, \alpha \ \mathbb{def}_S \ \vartheta\}$.
- $\mathbf{acc}(S) = \{\vartheta \in (\mathbb{A} \cup \mathbb{R} \cup \mathbb{T}) \mid \forall \alpha \in \mathbb{R} \,.\, \alpha \ \mathbb{def}_S \ \vartheta$ implies $\alpha \in \mathbf{def}(S)\}$.

Intuitively, the set $\mathbf{def}(S)$ includes every argument, attack and support that is defeated by considering the elements in S. Then, the set $\mathbf{acc}(S)$ contains every argument, attack and support that is defended by S. This is shown in the example below.

Example 5. Consider the ASAF Δ shown in Fig. 4, and the set $S = \{\mathtt{a}, \alpha_1, \beta_1, \beta_2\}$. We have that α_1 defeats any $\vartheta \in \{\mathtt{b}, \mathtt{c}, \mathtt{d}, \alpha_2\}$ given S and $\alpha_2 \ \mathbb{def}_S \ \alpha_1$. Moreover, $\mathbf{def}(S) = \{\mathtt{b}, \mathtt{c}, \mathtt{d}, \alpha_2\}$ and $\mathbf{acc}(S) = S$ implying $S \in \mathcal{E}_{co}(\Delta)$, being a complete extension of Δ (this will be clearer after providing Definition 10, where complete extensions for HOAFNs are formally defined). □

[5] Note that, as stated in [26], the ASAF and RAFN frameworks are generalizations of Dung's AF, meaning that when no supports and no higher-interactions are considered they yield equivalent results to those obtained for an AF under the complete, preferred, stable and grounded semantics.

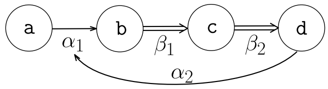


Fig. 4. HOAFN Δ from Examples 5-7.

2.3.2. Recursive argumentation framework with necessities

The *Recursive Argumentation Framework with Necessities (RAFN)* proposed in [28] also extends Dung's framework by incorporating bipolar high-order interactions. In that way, the RAFN allows, similarly to the case of ASAF, to have a framework where attack and support relations are defined not only between arguments, but also targeting the attack and support relations themselves. Note that, differently from [28], here we are considering a restricted version of the RAFN where the source of each support is a single argument.

The RAFN combines the RAF [56,61] interpretation of attacks with that of BAF under the necessity interpretation of supports (*i.e.* AFN [29]). Consequently, differently from the ASAF, the RAFN does not account for conflicts originated from the consideration of the indirect defeats of the AFRA. The defeats occurring in a RAFN can be formalized as follows.

Definition 8 *(RAFN defeats).* Let $\langle \mathbb{A}, \mathbb{R}, \mathbb{T} \rangle$ be a RAFN, $\vartheta \in (\mathbb{A} \cup \mathbb{R} \cup \mathbb{T})$, $a \in \mathbb{A}$, and $S \subseteq (\mathbb{A} \cup \mathbb{R} \cup \mathbb{T})$. We say that argument *$a$ defeats ϑ given S* (denoted as $a \ \mathbb{def}_S \ \vartheta$) if either: i) $(a, \vartheta) \in \mathbf{b}(\mathbb{R} \cap S)$, or ii) there exists $b \in \mathbb{A}$ such that $(a, b) \in \mathbf{b}(\mathbb{R} \cap S)$ and $(b, \vartheta) \in \mathbf{b}(\mathbb{T} \cap S)^+$.

For any RAFN the notions of *defeated set* and *acceptable set* (given a set S of arguments, attack names and support names) can be defined as follows:

Definition 9 *(RAFN defeated and acceptable sets).* Let $\langle \mathbb{A}, \mathbb{R}, \mathbb{T} \rangle$ be a RAFN and $S \subseteq (\mathbb{A} \cup \mathbb{R} \cup \mathbb{T})$. We define:

- $\mathbf{def}(S) = \{\vartheta \in (\mathbb{A} \cup \mathbb{R} \cup \mathbb{T}) \mid \exists\, b \in \mathbb{A} \cap S \,.\, b \ \mathbb{def}_S \ \vartheta\}$;
- $\mathbf{acc}(S) = \{\vartheta \in (\mathbb{A} \cup \mathbb{R} \cup \mathbb{T}) \mid \forall\, b \in \mathbb{A} \,.\, b \ \mathbb{def}_S \ \vartheta$ implies $b \in \mathbf{def}(S)\}$.

Example 6. Consider the RAFN Δ shown in Fig. 4. We have that for $S = \{\mathtt{a}, \alpha_1, \alpha_2, \beta_1, \beta_2\}$, argument $\mathtt{a}$ defeats all $\vartheta \in \{\mathtt{b}, \mathtt{c}, \mathtt{d}\}$ (given S) and $\mathtt{d} \ \mathbb{def}_S \ \alpha_1$. Also, $\mathbf{def}(S) = \{\mathtt{b}, \mathtt{c}, \mathtt{d}\}$ and $\mathbf{acc}(S) = S$ (thus we will have that $S \in \mathcal{E}_{co}(\Delta)$ is a complete extension of Δ, cf. Definition 10). Moreover, if we set $\mathbf{t}(\alpha_2) = \mathtt{b}$, we have that $\mathtt{d} \ \mathbb{def}_S \ \vartheta$ for every $\vartheta \in \{\mathtt{b}, \mathtt{c}, \mathtt{d}\}$. $\square$

Although similar in syntax, ASAF and RAFN may yield different outcomes (*e.g.* extensions), as discussed in the following section.

2.3.3. Semantics

The main difference between the ASAF and RAFN approaches, presented in the previous subsections, relies on the indirect defeat propagation from an argument a to the attacks having a as source argument, adopted by the ASAF (see Definition 6) to be consistent with the AFRA framework previously proposed in [20]. This form of indirect defeat is not present in the RAFN approach (see Definition 8), which extends the RAF.

Based on the general notion of HOAFN and the definition of the defeated and acceptable sets of elements of the framework, we next define the semantics of the ASAF and the RAFN in a unified way following the approach adopted in Section 2.1.[6]

Definition 10 *(HOAFN semantic notions and extensions).* Let $\Delta = \langle \mathbb{A}, \mathbb{R}, \mathbb{T} \rangle$ be a HOAFN and $S \subseteq (\mathbb{A} \cup \mathbb{R} \cup \mathbb{T})$. We say that S is:

- *conflict-free* iff $S \cap \mathbf{def}(S) = \emptyset$;
- *admissible* iff it is conflict-free and $S \subseteq \mathbf{acc}(S)$;
- a *complete extension* of Δ iff it is conflict-free and $S = \mathbf{acc}(S)$;
- a *preferred extension* of Δ iff it is a maximal (w.r.t. $\subseteq$) complete extension;
- a *stable extension* of Δ iff it is a preferred extension such that $S \cup \mathbf{def}(S) = (\mathbb{A} \cup \mathbb{R} \cup \mathbb{T})$;
- the *grounded extension* of Δ iff it is the smallest (w.r.t. $\subseteq$) complete extension.

It can be noted that the way in which semantics are presented in the preceding definition slightly deviates from the way in which they were originally presented for the ASAF and the RAFN in [27] and [28], respectively. On the one hand, for the ASAF, the difference relies on the characterization of the functions **def** and **acc**, instead of considering Dung's classical notion of acceptability. On the other hand, for the RAFN, our definition is not based on the notion of structures proposed in [28]. Instead, we define the extensions of the framework as a set of arguments, attacks and supports, in order to align with the approach adopted by the ASAF. Notwithstanding this, even though Definition 10 characterizes the extensions of a HOAFN in a different way, the extensions obtained following this definition are equivalent to the outcome obtained following the approaches proposed in [27] and [28].

[6] Given our unified approach to HOAFN, with a little abuse of notation, the semantic notions concerning a given HOAFN are intended to cover the case of ASAF or RAFN depending on the context. For instance, functions **def** and **acc** are based on Definition 7 or on Definition 9, depending on whether they are used in the ASAF or RAFN context, respectively.

As noted before, although the ASAF and the RAFN are syntactically equivalent frameworks, they might yield different outcomes in terms of acceptable elements. To illustrate the differences in the semantics depending on whether the HOAFN is an ASAF or a RAFN, let us consider the following examples.

Example 7. Consider the HOAFN Δ shown in Fig. 4 and the results illustrated in Examples 5 and 6. Under the RAFN semantics, the set $S = \{\mathtt{a}, \alpha_1, \alpha_2, \beta_1, \beta_2\}$ is a complete extension of Δ, while that is not the case under the ASAF semantics. This is due to the fact that, in the ASAF semantics, as α_1 defeats $\mathtt{d}$, it also defeats α_2. Conversely, in the RAFN semantics α_2 is not defeated as it is not attacked by any element. $\square$

As another example, consider the following.

Example 8. The initial example from the introduction can be represented by the HOAFN $\Delta = \langle \mathbb{A}, \mathbb{R}, \mathbb{T} \rangle$ in Fig. 1 (left), with $\mathbb{A} = \{\mathtt{d}, \mathtt{pt}, \mathtt{p}, \mathtt{st}, \mathtt{w}, \mathtt{rn}\}$, $\mathbb{R} = \{\alpha_0, \alpha_1, \alpha_2, \alpha_3, \alpha_4\}$, and $\mathbb{T} = \{\beta_0, \beta_1\}$; where $\alpha_0, \alpha_1, \alpha_2, \alpha_3$ are first-level attacks with $\mathbf{b}(\alpha_0) = (\mathtt{st}, \mathtt{rn})$, $\mathbf{b}(\alpha_1) = (\mathtt{rn}, \mathtt{st})$, $\mathbf{b}(\alpha_2) = (\mathtt{d}, \mathtt{pt})$, $\mathbf{b}(\alpha_3) = (\mathtt{pt}, \mathtt{d})$; β_0 is a first-level support with $\mathbf{b}(\beta_0) = (\mathtt{st}, \mathtt{d})$; the attack α_4 and the support β_1 are second-level interactions with $\mathbf{b}(\alpha_4) = (\mathtt{p}, \alpha_3)$ and $\mathbf{b}(\beta_1) = (\mathtt{w}, \beta_0)$. Under RAFN (resp. ASAF) semantics, there are three complete extensions of Δ:

- $E_0 = \{\mathtt{w}, \mathtt{p}, \alpha_0, \alpha_1, \alpha_2, \alpha_4, \beta_0, \beta_1\}$ (resp. $E_0 = \{\mathtt{w}, \mathtt{p}, \alpha_4, \beta_0, \beta_1\}$), which is also the grounded extension;
- $E_1 = \{\mathtt{st}, \mathtt{w}, \mathtt{d}, \mathtt{p}, \alpha_0, \alpha_1, \alpha_2, \alpha_4, \beta_0, \beta_1\}$ (resp. $E_1 = \{\mathtt{st}, \mathtt{w}, \mathtt{d}, \mathtt{p}, \alpha_0, \alpha_2, \alpha_4, \beta_0, \beta_1\}$), which is both preferred and stable; and
- $E_2 = \{\mathtt{rn}, \mathtt{w}, \mathtt{pt}, \mathtt{p}, \alpha_0, \alpha_1, \alpha_2, \alpha_4, \beta_0, \beta_1\}$ (resp. $E_2 = \{\mathtt{rn}, \mathtt{w}, \mathtt{pt}, \mathtt{p}, \alpha_1, \alpha_4, \beta_0, \beta_1\}$) which is also preferred and stable.

The difference on the acceptance of α_0 w.r.t. E_0 is that $\alpha_0 \notin \mathbf{acc}(E_0)$ under the ASAF semantics since $\alpha_1 \; \mathbb{def}_{E_0} \; \alpha_0$ (*i.e.* α_1 defeats α_0 given E_0, as it defeats the source of α_0, which is $\mathtt{st}$) and $\alpha_1 \notin \mathbf{def}(E_0)$. In contrast, under the RAFN semantics $\alpha_0 \in \mathbf{acc}(E_0)$, since there is no argument defeating α_0 given E_0 (in particular, since there is no attack targeting α_0). An analogous analysis can be done for α_1 and α_2. $\square$

Example 9. Continuing with the example from the introduction, the updated scenario described in Example 2 and illustrated in Fig. 1 (right) corresponds to the HOAFN from Example 8 with the addition of argument $\mathtt{ns}$ and attack α_5 such that $\mathbf{b}(\alpha_5) = (ns, \beta_0)$. Here, under RAFN (resp. ASAF) semantics, there exist three complete extensions:

- $E_0 = \{\mathtt{w}, \mathtt{p}, \mathtt{ns}, \mathtt{d}, \alpha_0, \alpha_1, \alpha_2, \alpha_4, \alpha_5, \beta_1\}$ (resp. $E_0 = \{\mathtt{w}, \mathtt{p}, \mathtt{ns}, \mathtt{d}, \alpha_4, \alpha_5, \beta_1\}$), which is also the grounded extension;
- $E_1 = \{\mathtt{st}, \mathtt{w}, \mathtt{d}, \mathtt{p}, \mathtt{ns}, \alpha_0, \alpha_1, \alpha_2, \alpha_4, \alpha_5, \beta_1\}$ (resp. $E_1 = \{\mathtt{st}, \mathtt{w}, \mathtt{d}, \mathtt{p}, \mathtt{ns}, \alpha_0, \alpha_2, \alpha_4, \alpha_5, \beta_1\}$), which is both preferred and stable; and
- $E_2 = \{\mathtt{rn}, \mathtt{w}, \mathtt{p}, \mathtt{ns}, \mathtt{d}, \alpha_0, \alpha_1, \alpha_2, \alpha_4, \alpha_5, \beta_1\}$ (resp. $E_2 = \{\mathtt{rn}, \mathtt{w}, \mathtt{p}, \mathtt{ns}, \mathtt{d}, \alpha_1, \alpha_2, \alpha_4, \alpha_5, \beta_1\}$) which is also preferred and stable.

An alternative way to express HOAFN outcomes (*i.e.*, extensions) is by means of *labellings*. Given a HOAFN, the labellings of the framework can be defined in terms of its extensions: a labeling is a triple containing the accepted, defeated and undecided elements, with respect to a given extension.

Definition 11. Let $\Delta = \langle \mathbb{A}, \mathbb{R}, \mathbb{T} \rangle$ be a HOAFN and E an $\mathcal{s}$-extension of Δ, with $\mathcal{s} \in \{co, gr, st, pr\}$. A labeling of Δ under semantics $\mathcal{s}$ is a function $L_E : (\mathbb{A} \cup \mathbb{R} \cup \mathbb{T}) \mapsto \{\mathtt{in}, \mathtt{out}, \mathtt{undec}\}$ such that for any $\vartheta \in (\mathbb{A} \cup \mathbb{R} \cup \mathbb{T})$ it holds that i) $L_E(\vartheta) = \mathtt{in}$ iff $\vartheta \in \mathbf{acc}(E)$, ii) $L_E(\vartheta) = \mathtt{out}$ iff $\vartheta \in \mathbf{def}(E)$, and iii) $L_E(\vartheta) = \mathtt{undec}$ iff $\vartheta \in (\mathbb{A} \cup \mathbb{R} \cup \mathbb{T}) \setminus (\mathbf{acc}(E) \cup \mathbf{def}(E))$.

With a little abuse of notation, a labeling L_E will be simply denoted by L whenever the $\mathcal{s}$-extension is understood, and it will also be denoted as a triple $\langle \mathtt{in}(L), \mathtt{out}(L), \mathtt{undec}(L) \rangle$, where $\mathtt{in}(L) = \{\vartheta \mid L(\vartheta) = \mathtt{in}\}$, $\mathtt{out}(L) = \{\vartheta \mid L(\vartheta) = \mathtt{out}\}$, and $\mathtt{undec}(L) = \{\vartheta \mid L(\vartheta) = \mathtt{undec}\}$.

The set of all $\mathcal{s}$-labellings of Δ will be denoted by $\mathfrak{L}_{\mathcal{s}}(\Delta) = \{\langle \mathtt{in}(L_E), \mathtt{out}(L_E), \mathtt{undec}(L_E) \rangle \mid E \in \mathcal{E}_{\mathcal{s}}(\Delta)\}$.

In the following we also use the concept of valuation w.r.t. an $\mathcal{s}$-extension E denoted as $\mathbf{v}_E$, or simply $\mathbf{v}$ whenever E is understood. Thus, $\mathbf{v}_E(\vartheta) = \mathtt{true}$ iff $L_E(\vartheta) = \mathtt{in}$, $\mathbf{v}_E(\vartheta) = \mathtt{false}$ iff $L_E(\vartheta) = \mathtt{out}$, and $\mathbf{v}_E(\vartheta) = \mathtt{unknown}$ iff $L_E(\vartheta) = \mathtt{undec}$.

Example 10. Let Δ be the HOAFN of Example 8. When instantiating Δ with the RAFN (resp. ASAF) semantics, there exist three complete labellings, one for each complete extension of Δ (namely, E_0, E_1 and E_2):

- $L_{E_0} = \langle \{\mathtt{w}, \mathtt{p}, \alpha_0, \alpha_1, \alpha_2, \alpha_4, \beta_0, \beta_1\}, \{\alpha_3\}, \{\mathtt{rn}, \mathtt{st}, \mathtt{d}, \mathtt{pt}\} \rangle$
(resp. $L_{E_0} = \langle \{\mathtt{w}, \mathtt{p}, \alpha_4, \beta_0, \beta_1\}, \{\alpha_3\}, \{\mathtt{rn}, \mathtt{st}, \mathtt{d}, \mathtt{pt}, \alpha_0, \alpha_1, \alpha_2\} \rangle$);
- $L_{E_1} = \langle \{\mathtt{st}, \mathtt{w}, \mathtt{d}, \mathtt{p}, \alpha_0, \alpha_1, \alpha_2, \alpha_4, \beta_0, \beta_1\}, \{\alpha_3\}, \emptyset \rangle$
(resp. $L_{E_1} = \langle \{\mathtt{st}, \mathtt{w}, \mathtt{d}, \mathtt{p}, \alpha_0, \alpha_2, \alpha_4, \beta_0, \beta_1\}, \{\alpha_1, \alpha_3\}, \emptyset \rangle$); and
- $L_{E_2} = \langle \{\mathtt{rn}, \mathtt{w}, \mathtt{pt}, \mathtt{p}, \alpha_0, \alpha_1, \alpha_2, \alpha_4, \beta_0, \beta_1\}, \{\alpha_3\}, \emptyset \rangle$
(resp. $L_{E_2} = \langle \{\mathtt{rn}, \mathtt{w}, \mathtt{pt}, \mathtt{p}, \alpha_1, \alpha_4, \beta_0, \beta_1\}, \{\alpha_0, \alpha_2, \alpha_3\}, \emptyset \rangle$). $\square$

Algorithm 1 INCR-AF-SE($\Lambda, u, \mathscr{s}, E$, AFS) [49].

Input: AF $\Lambda = \langle \mathsf{A}, \mathsf{R} \rangle$,
set of updates u for Λ,
semantics $\mathscr{s}$,
extension $E \in \mathcal{E}_{\mathscr{s}}(\Lambda)$,
AF solver AFS for the task SE.
Output: An $\mathscr{s}$-extension in $\mathcal{E}_{\mathscr{s}}(u(\Lambda))$ if there exists one; $\perp$ otherwise.

$I = \mathbf{inf}(u, \Lambda, E)$;
if $(I = \emptyset)$ **then**
  return E;
$\langle \mathsf{A}_d, \mathsf{R}_d \rangle = \mathbf{red}(u, \Lambda, E)$;
Let $E_d = \text{AFS}_{\mathscr{s}}^{\text{SE}}(\langle \mathsf{A}_d, \mathsf{R}_d \rangle)$;
if $(E_d \neq \perp)$ **then**
  return $(E \setminus I) \cup E_d$;
return $\text{AFS}_{\mathscr{s}}(u(\Lambda))$;

2.4. Incremental computation of AF extensions

Argumentation frameworks model dynamic situations where the existence of arguments as well as their interactions change over the time. As a consequence, the sets of accepted arguments (*i.e.*, extensions under a given $\mathscr{s}$-semantics) need to be repeatedly computed after performing updates on the framework. Thus, a relevant problem consists in checking credulous acceptance after such updates avoiding, whenever possible, the computation from scratch. This problem has been recently investigated and several "incremental" solvers have been proposed for AFs [62–64].

In this section, we briefly overview an existing technique addressing the problem of efficiently computing an extension of an AF after updates have been performed. This problem, denoted as $\text{SE}_{\mathscr{s}}$, consists in computing an extension of an AF under a given semantics $\mathscr{s}$, if there exists one. Algorithm 1 (proposed in [49]) solves the SE problem incrementally for an updated AF. As we shall see, it can be used as an orthogonal tool of an "incremental" solver for the CA problem in dynamic HOAFNs (see Algorithm 4).[7]

Formally, in the static setting, the $\text{SE}_{\mathscr{s}}$ problem is defined as follows: given an AF $\langle \mathsf{A}, \mathsf{R} \rangle$ and a semantics $\mathscr{s}$, compute an $\mathscr{s}$-extension of $\langle \mathsf{A}, \mathsf{R} \rangle$, if such an extension exists; $\perp$ otherwise.[8] The underlying idea of the incremental solution described in Algorithm 1 is that of using an initial $\mathscr{s}$-extension and the set of updates to identify a restricted portion of the AF whose arguments' status may change after the updates have been performed, and recompute the status of the arguments in such a restricted part. Here, an update consists in the addition or deletion of a set of arguments or a set of attacks; the former are referred to as 'positive' updates, whereas the latter are referred to as 'negative' updates. As defined in [49], to add an attack (a, b) to an AF, both a and b must belong to the framework, and only 'isolated' arguments may be deleted. As the addition and deletion of a set of (isolated) arguments is trivial, next we only discuss the case where a set of attacks is added or deleted.

Specifically, Algorithm 1 takes advantage of the specific updates performed and of the given initial $\mathscr{s}$-extension. It is based on the computation of the so-called *influenced set*. Intuitively, given an AF Λ, a set of positive (resp. negative) updates $u = \{+(a_1, b_1), \dots, +(a_n, b_n)\}$ (resp. $u = \{-(a_1, b_1), \dots, -(a_n, b_n)\}$), and an initial $\mathscr{s}$-extension E of Λ, the influenced set $I = \mathbf{inf}(u, \Lambda, E)$ consists of the arguments whose acceptance status (according to the semantics $\mathscr{s}$) may change after performing the updates.[9] More in detail, the influenced set consists of the arguments that are reachable from at least one argument b_i in the argumentation graph without passing through any `out`-argument whose status does not change after the update is performed—these are the `out`-arguments that are attacked by an argument in E that in turn is not reachable from some b_i argument. Thus, the influenced set depends, other than on the structure of the AF, both on the initial extension and the performed updates. Moreover, it is used to define the reduced AF (denoted as $\mathbf{red}(u, \Lambda, E)$, which contains all the arguments and attacks needed to compute the acceptance status of the arguments in the influenced set.

The reduced AF $\mathbf{red}(u, \Lambda, E)$ is defined as the AF $\langle \mathsf{A}_d, \mathsf{R}_d \rangle$ consisting of the following arguments and attacks:

1. the arguments in the influenced set (for the given update and extension);
2. the arguments in the initial $\mathscr{s}$-extension E which are not in the influenced set and that, in the updated AF, attack an argument in the influenced set;
3. the attacks of the updated AF between any pair of arguments considered in items 1 and 2; and
4. a self-attack for each argument in the influenced set that, in the updated AF, is attacked by an argument x which is undecided w.r.t. the extension E.

Example 11. Consider the AF $\Lambda = \langle \{\mathtt{a}, \mathtt{b}, \mathtt{c}, \mathtt{d}\}, \{(\mathtt{b}, \mathtt{c}), (\mathtt{d}, \mathtt{c})\} \rangle$, the (complete, preferred, stable and grounded) extension $E = \{\mathtt{a}, \mathtt{b}, \mathtt{d}\}$ and the update $u = \{+(\mathtt{a}, \mathtt{b}), +(\mathtt{c}, \mathtt{b})\}$. We have that $\mathbf{inf}(u, \Lambda, E) = \{\mathtt{b}\}$ and, indeed, only argument b may change its status after the

[7] We believe that, in principle, the idea behind the incremental approach proposed in [49] could also be exploited by approaches participating in the ICCMA dynamic track [62–64], though this could require revising the original strategy of each solver.

[8] $\perp$ is a possible outcome only under the stable semantics, as for the other semantics considered in this paper the existence of an extension is always guaranteed.

[9] In [49] the computation of the influenced set w.r.t. a set of updates has been shown to be equivalent to the computation of the influenced set w.r.t. a single attack update.

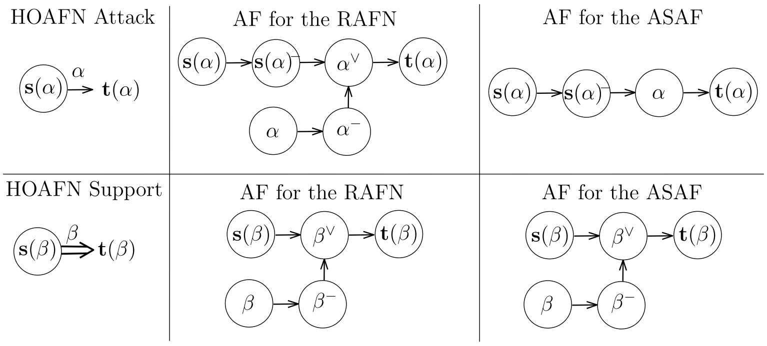


Fig. 5. Illustration of Definitions 12 and 13. Note that, in the first column, $\mathbf{t}(\alpha)$ and $\mathbf{t}(\beta)$ are not depicted within circles because they can be either an argument, an attack or a support.

update. The reduced AF is $\mathbf{red}(u, \Lambda, E) = \langle \{\mathsf{a}, \mathsf{b}\}, \{(\mathsf{a}, \mathsf{b})\} \rangle$ and also includes, other than the argument in the influenced set, argument a which may influence the status of argument b (*i.e.* the only argument belonging to the influenced set). □

Some key points of Algorithm 1 are summarized next. After the reduced AF $\Lambda_d = \langle \mathsf{A}_d, \mathsf{R}_d \rangle$ has been computed (Line 4), an $\mathcal{S}$-extension E_d of Λ_d is computed (if there exists one) and 'combined' with the set of arguments in $E \setminus I$ (Line 7) whose status did not change. Considering the stable semantics, if the reduced AF does not admit extensions, then the algorithm computes an extension from scratch for the whole updated AF (Line 8) by calling the solver AFS.

In [49] it has been shown that, assuming the solver AFS is sound and complete and $\mathcal{E}_{\mathcal{S}}(u(\Lambda)) \neq \emptyset$ (resp. $\mathcal{E}_{st}(u(\Lambda)) = \emptyset$), then Algorithm 1 returns an $\mathcal{S}$-extension of $\mathcal{E}_{\mathcal{S}}(u(\Lambda))$ (resp. $\perp$), where $u(\Lambda)$ denotes the updated AF obtained by performing the set of updates u on AF Λ.

3. Translating HOAFN into AF

In this section we present an approach showing how a HOAFN Δ can be encoded into an 'equivalent' AF Δ_{AF}. The term equivalent here means that there exists a one-to-one mapping between the complete extensions of Δ and those of Δ_{AF}, and that from a complete extension of Δ (resp. Δ_{AF}) the corresponding complete extension of Δ_{AF} (resp. Δ) can be immediately derived. We first introduce the translation for the case of a RAFN, and then for an ASAF. Later, we formally show that there exists a one-to-one correspondence between the $\mathcal{S}$-extensions of a HOAFN and the $\mathcal{S}$-extensions of its derived AF.

Definition 12 *(AF for RAFN).* Let $\Delta = \langle \mathbb{A}, \mathbb{R}, \mathbb{T} \rangle$ be a RAFN. The *AF for* Δ is $\Delta_{AF} = \langle \mathsf{A}, \mathsf{R} \rangle$, where:

$$\begin{aligned} \mathsf{A} = \; & \mathbb{A} \cup \{\alpha, \mathbf{s}(\alpha)^-, \alpha^\vee, \alpha^- \mid \alpha \in \mathbb{R}\} \cup \{\beta, \beta^\vee, \beta^- \mid \beta \in \mathbb{T}\} \\ \mathsf{R} = \; & \{(\mathbf{s}(\alpha), \mathbf{s}(\alpha)^-), (\mathbf{s}(\alpha)^-, \alpha^\vee), (\alpha, \alpha^-), (\alpha^-, \alpha^\vee), (\alpha^\vee, \mathbf{t}(\alpha)) \mid \alpha \in \mathbb{R}\} \cup \\ & \{(\mathbf{s}(\beta), \beta^\vee), (\beta, \beta^-), (\beta^-, \beta^\vee), (\beta^\vee, \mathbf{t}(\beta)) \mid \beta \in \mathbb{T}\}. \end{aligned}$$

The translation proposed in the preceding definition is illustrated in the second column of Fig. 5 and is carried out as follows: each attack α, including $\mathbf{s}(\alpha)$ and $\mathbf{t}(\alpha)$, is mapped into the AF $\langle \{\mathbf{s}(\alpha), \mathbf{t}(\alpha), \alpha, \alpha^-, \alpha^\vee, \mathbf{s}(\alpha)^-\}, \{(\mathbf{s}(\alpha), \mathbf{s}(\alpha)^-), (\mathbf{s}(\alpha)^-, \alpha^\vee), (\alpha, \alpha^-), (\alpha^-, \alpha^\vee), (\alpha^\vee, \mathbf{t}(\alpha))\} \rangle$ consisting of 6 arguments (4 of which are new) and 5 attacks, whereas each support β, including $\mathbf{s}(\beta)$ and $\mathbf{t}(\beta)$, is mapped into the AF $\langle \{\mathbf{s}(\beta), \mathbf{t}(\beta), \beta, \beta^-, \beta^\vee\}, \{(\mathbf{s}(\beta), \beta^\vee), (\beta, \beta^-), (\beta^-, \beta^\vee), (\beta^\vee, \mathbf{t}(\beta))\} \rangle$ consisting of 5 arguments (3 of which are new) and 4 attacks.

Intuitively, when mapping an attack α of the RAFN Δ, the argument α in the AF for Δ determines whether the attack is accepted or not, while arguments α^- and $\mathbf{s}(\alpha)^-$ are used to channel the impact of defeats on α or $\mathbf{s}(\alpha)$ to $\mathbf{t}(\alpha)$, through $\alpha^\vee$. When mapping a support β, the argument β in the AF represents the support itself and is used to determine whether it is accepted or not, whereas argument $\beta^\vee$ is used to propagate defeats on $\mathbf{s}(\beta)$ to $\mathbf{t}(\beta)$, and β^- is used to stop from propagating those defeats in case β is itself defeated.

Definition 13 *(AF for ASAF).* Let $\Delta = \langle \mathbb{A}, \mathbb{R}, \mathbb{T} \rangle$ be an ASAF. The *AF for* Δ is $\Delta_{AF} = \langle \mathsf{A}, \mathsf{R} \rangle$, where:

$$\begin{aligned} \mathsf{A} = \; & \mathbb{A} \cup \{\alpha, \mathbf{s}(\alpha)^- \mid \alpha \in \mathbb{R}\} \cup \{\beta, \beta^-, \beta^\vee \mid \beta \in \mathbb{T}\} \\ \mathsf{R} = \; & \{(\mathbf{s}(\alpha), \mathbf{s}(\alpha)^-), (\mathbf{s}(\alpha)^-, \alpha), (\alpha, \mathbf{t}(\alpha)) \mid \alpha \in \mathbb{R}\} \cup \\ & \{(\mathbf{s}(\beta), \beta^\vee), (\beta, \beta^-), (\beta^-, \beta^\vee), (\beta^\vee, \mathbf{t}(\beta)) \mid \beta \in \mathbb{T}\}. \end{aligned}$$

The translation for the ASAF is shown in the third column of Fig. 5, and is carried out as follows: each attack α is mapped into the AF $\langle \{\mathbf{s}(\alpha), \mathbf{t}(\alpha), \alpha, \mathbf{s}(\alpha)^-\}, \{(\mathbf{s}(\alpha), \mathbf{s}(\alpha)^-), (\mathbf{s}(\alpha)^-, \alpha), (\alpha, \mathbf{t}(\alpha))\} \rangle$ consisting of 4 arguments and 3 attacks, whereas each support β is mapped into the same AF used for the RAFN (see last row of Fig. 5).

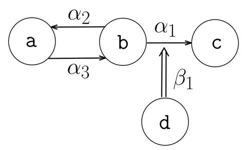


Fig. 6. HOAFN Δ of Example 12.

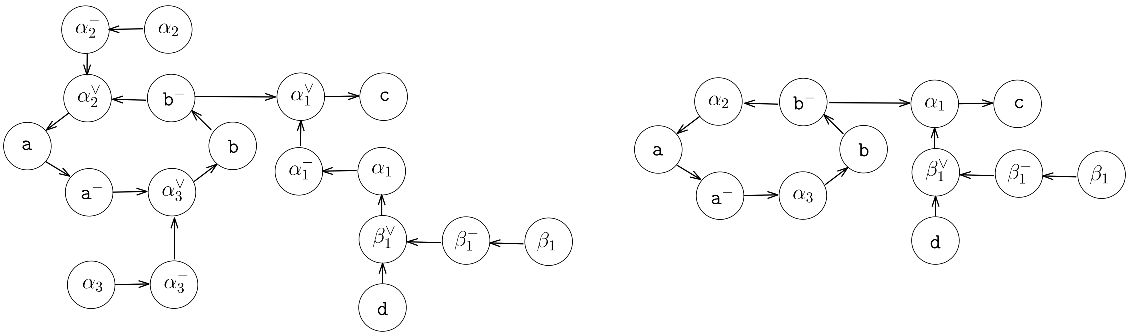


Fig. 7. On the left-hand side (resp. right-hand side), AF for the RAFN (resp. ASAF) Δ of Example 12.

Here, for the case of an attack α of Δ, arguments α and $\mathbf{s}(\alpha)^-$ in the AF for Δ determine whether α is accepted or not, and are used to propagate defeats on $\mathbf{s}(\alpha)$ to the attack itself, and to channel the impact of those defeats to $\mathbf{t}(\alpha)$.

Example 12. Let $\Delta = \langle \mathbb{A}, \mathbb{R}, \mathbb{T} \rangle$ be the HOAFN shown in Fig. 6 with $\mathbb{A} = \{\mathtt{a}, \mathtt{b}, \mathtt{c}, \mathtt{d}\}$, $\mathbb{R} = \{\alpha_1, \alpha_2, \alpha_3\}$ and $\mathbb{T} = \{\beta_1\}$, where $\mathbf{b}(\alpha_1) = (\mathtt{b}, \mathtt{c})$, $\mathbf{b}(\alpha_2) = (\mathtt{b}, \mathtt{a})$ and $\mathbf{b}(\alpha_3) = (\mathtt{a}, \mathtt{b})$ are first-level attacks, and $\mathbf{b}(\beta_1) = (\mathtt{d}, \alpha_1)$ is a second-level support. The AF for the RAFN (resp. ASAF) Δ is shown in Fig. 7 on the left-hand side (resp. right-hand side). For instance, attack $\mathbf{b}(\alpha_3) = (\mathtt{a}, \mathtt{b})$ corresponds to the chains of attacks from a and α_3 to b in the AF for the RAFN, and to the chain of attacks from a to b in the AF for the ASAF. In both translations, the support $\mathbf{b}(\beta_1) = (\mathtt{d}, \alpha_1)$ corresponds to the attacks $(\mathtt{d}, \beta_1^\vee)$, (β_1, β_1^-), $(\beta_1^-, \beta_1^\vee)$ and $(\beta_1^\vee, \alpha_1)$ in the AF. □

Before showing the equivalence between $\mathcal{s}$-extensions of a HOAFN Δ and $\mathcal{s}$-extensions of its AF Δ_{AF}, we introduce functions mapping extensions of Δ into extensions of Δ_{AF} and vice-versa. We distinguish the case where the HOAFN is a RAFN from the case where it is an ASAF.

Definition 14 *(***RAFNtoAF** *and* **AFtoRAFN** *functions).* Let $\Delta = \langle \mathbb{A}, \mathbb{R}, \mathbb{T} \rangle$ be a RAFN and $\Delta_{AF} = \langle \mathsf{A}, \mathsf{R} \rangle$ be the AF derived from Δ (cf. Definition 12). We define the functions $\mathbf{RAFNtoAF}_\Delta$ and $\mathbf{AFtoRAFN}_\Delta$, where E is a complete extension of Δ and E_{AF} is a complete extension of Δ_{AF}, as follows:

- $\mathbf{RAFNtoAF}_\Delta(E) = \ E \ \cup \{\mathbf{s}(\alpha)^- \mid \alpha \in \mathbb{R} \wedge \mathbf{s}(\alpha) \in \mathbf{def}(E)\} \cup$
  $\{\alpha^- \mid \alpha \in \mathbb{R} \cap \mathbf{def}(E)\} \cup$
  $\{\alpha^\vee \mid \alpha \in \mathbb{R} \wedge \{\alpha, \mathbf{s}(\alpha)\} \subseteq E\} \cup$
  $\{\beta^- \mid \beta \in \mathbb{T} \cap \mathbf{def}(E)\} \cup$
  $\{\beta^\vee \mid \beta \in \mathbb{T} \cap E \wedge \mathbf{s}(\beta) \in \mathbf{def}(E)\}$
- $\mathbf{AFtoRAFN}_\Delta(E_{AF}) = E_{AF} \cap (\mathbb{A} \cup \mathbb{R} \cup \mathbb{T})$.

The following example illustrates the use of the mapping functions for a HOAFN that is a RAFN.

Example 13. Let Δ be the RAFN from Example 12 and Δ_{AF} be the AF for Δ. The set of complete extensions of Δ are: $\mathcal{E}_{co}(\Delta) = \{E = \{\alpha_1, \alpha_2, \alpha_3, \beta_1, \mathtt{d}\},\ E' = E \cup \{\mathtt{a}, \mathtt{c}\},\ E'' = E \cup \{\mathtt{b}\}\}$. Considering E', we have that $\mathbf{def}(E') = \{\mathtt{b}\}$ and thus $E'_{AF} = \mathbf{RAFNtoAF}_\Delta(E') = E' \cup \{\mathtt{b}^-\} \cup \{\alpha_3^\vee\}$.

Clearly, $\mathbf{AFtoRAFN}_\Delta(E'_{AF}) = E'_{AF} \cap (\mathbb{A} \cup \mathbb{R} \cup \mathbb{T}) = E'$. Moreover, $E_{AF} = \mathbf{RAFNtoAF}_\Delta(E) = E$, and $E''_{AF} = \mathbf{RAFNtoAF}_\Delta(E'') = E'' \cup \{a^-\} \cup \{\alpha_1^\vee, \alpha_2^\vee\}$. □

Definition 15 *(***ASAFtoAF** *and* **AFtoASAF** *functions).* Let $\Delta = \langle \mathbb{A}, \mathbb{R}, \mathbb{T} \rangle$ be an ASAF and $\Delta_{AF} = \langle \mathsf{A}, \mathsf{R} \rangle$ be the AF derived from Δ (cf. Definition 13). We define the functions $\mathbf{ASAFtoAF}_\Delta$ and $\mathbf{AFtoASAF}_\Delta$, where E is a complete extension of Δ and E_{AF} is a complete extension of Δ_{AF}, as follows:

- $\mathbf{ASAFtoAF}_\Delta(E) = E \cup \{\mathbf{s}(\alpha)^- \mid \alpha \in \mathbb{R} \wedge \mathbf{s}(\alpha) \in \mathbf{def}(E)\} \cup \{\beta^- \mid \beta \in \mathbb{T} \cap \mathbf{def}(E)\} \cup \{\beta^\vee \mid \beta \in \mathbb{T} \cap E \wedge \mathbf{s}(\beta) \in \mathbf{def}(E)\}$
- $\mathbf{AFtoASAF}_\Delta(E_{AF}) = E_{AF} \cap (\mathbb{A} \cup \mathbb{R} \cup \mathbb{T})$.

The following example illustrates the translation of the HOAFN from Example 12 in the case that the framework is an ASAF.

Example 14. Let Δ be the ASAF from Example 12 and the AF Δ_{AF} for Δ. The set of complete extensions of Δ are: $\mathcal{E}_{co}(\Delta) = \{E = \{\mathtt{d}, \beta_1\},\ E' = E \cup \{\mathtt{a}, \mathtt{c}, \alpha_3\},\ E'' = E \cup \{\mathtt{b}, \alpha_1, \alpha_2\}\}$. Considering E', we have that $\mathbf{def}(E') = \{\mathtt{b}, \alpha_1, \alpha_2\}$ and thus, $E'_{AF} = \mathbf{ASAFtoAF}_\Delta(E') = E' \cup \{\mathtt{b}^-\}$. In addition, it is clear that $\mathbf{AFtoASAF}_\Delta(E'_{AF}) = E'_{AF} \cap (\mathbb{A} \cup \mathbb{R} \cup \mathbb{T}) = E'$. Moreover, $E_{AF} = \mathbf{ASAFtoAF}_\Delta(E) = E$, and $E''_{AF} = \mathbf{ASAFtoAF}_\Delta(E'') = E'' \cup \{\mathtt{a}^-\}$. $\square$

Hereafter, whenever we use $\mathbf{AFtoHOAFN}_\Delta$ (resp. $\mathbf{HOAFNtoAF}_\Delta$) we refer to either $\mathbf{AFtoRAFN}_\Delta$ (resp. $\mathbf{RAFNtoAF}_\Delta$) or $\mathbf{AFtoASAF}_\Delta$ (resp. $\mathbf{ASAFtoAF}_\Delta$); we will omit Δ in the notation whenever it is understood. The following theorem states that the AF for a given HOAFN yields extensions which are equivalent to those of the HOAFN. The proof of this theorem, as well as relevant definitions required for it, are included in Appendix B. The same holds for the proof of all other formal results given in the paper.

Theorem 1. *Let $\Delta = \langle \mathbb{A}, \mathbb{R}, \mathbb{T} \rangle$ be a HOAFN, $\Delta_{AF} = \langle \mathsf{A}, \mathsf{R} \rangle$ its AF, $E \subseteq (\mathbb{A} \cup \mathbb{R} \cup \mathbb{T})$, $E_{AF} \subseteq \mathsf{A}$, and $\mathcal{s} \in \{co, pr, st, gr\}$ a semantics. It holds that E is an $\mathcal{s}$-extension of Δ iff $\mathbf{HOAFNtoAF}_\Delta(E)$ is an $\mathcal{s}$-extension of Δ_{AF}. Equivalently, it holds that E_{AF} is an $\mathcal{s}$-extension of Δ_{AF} iff $\mathbf{AFtoHOAFN}_\Delta(E_{AF})$ is an $\mathcal{s}$-extension of Δ.*

The translations of HOAFN to AF imply an increase in the size of the graph. This increase is polynomial w.r.t. the size of the input HOAFN (*i.e.*, $|\mathbb{A}| + |\mathbb{R}| + |\mathbb{T}|$). Particularly, given a HOAFN $\Delta = \langle \mathbb{A}, \mathbb{R}, \mathbb{T} \rangle$, let $\Delta_{AF} = \langle \mathsf{A}, \mathsf{R} \rangle$ be the AF for Δ. We have that under the RAFN interpretation $|\mathsf{A}| = |\mathbb{A}| + |\mathbf{s}(\mathbb{R})$[10]$| + 3 \cdot |\mathbb{R}| + 3 \cdot |\mathbb{T}|$ and $|\mathsf{R}| = |\mathbf{s}(\mathbb{R})| + 4 \cdot |\mathbb{R}| + 4 \cdot |\mathbb{T}|$, whereas under the ASAF interpretation we have $|\mathsf{A}| = |\mathbb{A}| + |\mathbf{s}(\mathbb{R})| + |\mathbb{R}| + 3 \cdot |\mathbb{T}|$ and $|\mathsf{R}| = |\mathbf{s}(\mathbb{R})| + 2 \cdot |\mathbb{R}| + 4 \cdot |\mathbb{T}|$.

Given this, the mappings from HOAFN to AF and vice-versa provided in this section express that HOAFN problems have the same complexity as those for AF [65]. This result allows to use a framework that is able to express in a more natural way some relationships of the argumentation process that can be modeled through either meta-knowledge (losing in intuitiveness, which is one of the key aspects of abstract argumentation), or by resorting to more complex abstract frameworks such as Preference-based AFs [66,67] or Abstract Dialectical Frameworks [68].

We point out that the mapping from a RAFN into an AF introduced in Definition 12 can be optimized when attacks and supports are not target elements of other attacks and supports. Specifically, *i)* if an attack α is neither attacked nor supported, then the mapping can be simplified by only adding to the AF an unattacked argument α; and *ii)* if a support β is neither attacked nor supported, then the mapping can be simplified by deleting the argument β^- as well as its incoming and outcoming attacks. This second optimization can also be applied to the case of ASAF frameworks. Both optimizations allow to reduce the size of the AF for the HOAFN (see Example 15 below).

The following two definitions capture the above-described intuitions.

Definition 16 *(Optimized AF for RAFN).* Let $\Delta = \langle \mathbb{A}, \mathbb{R}, \mathbb{T} \rangle$ be a RAFN. The *AF for* Δ is $\Delta_{AF} = \langle \mathsf{A}, \mathsf{R} \rangle$, where:

$$\begin{aligned}
\mathsf{A} = {} & \mathbb{A} \cup \{\alpha \mid \alpha \in \mathbb{R}\} \cup \\
& \{\beta, \beta^\vee \mid \beta \in \mathbb{T}\} \cup \\
& \{\mathbf{s}(\alpha)^-, \alpha^\vee, \alpha^- \mid \alpha \in \mathbb{R} \wedge \exists \gamma \in (\mathbb{R} \cup \mathbb{T}) \,.\, \mathbf{t}(\gamma) = \alpha\} \cup \\
& \{\beta^- \mid \beta \in \mathbb{T} \wedge \exists \gamma \in (\mathbb{R} \cup \mathbb{T}) \,.\, \mathbf{t}(\gamma) = \beta\}
\end{aligned}$$

$$\begin{aligned}
\mathsf{R} = {} & \{(\mathbf{s}(\alpha), \mathbf{t}(\alpha)) \mid \alpha \in \mathbb{R} \wedge \nexists\, \gamma \in (\mathbb{R} \cup \mathbb{T}) \,.\, \mathbf{t}(\gamma) = \alpha\} \cup \\
& \{(\mathbf{s}(\beta), \beta^\vee), (\beta^\vee, \mathbf{t}(\beta)) \mid \beta \in \mathbb{T}\} \cup \\
& \{(\beta, \beta^-), (\beta^-, \beta^\vee) \mid \beta \in \mathbb{T} \wedge \exists \gamma \in (\mathbb{R} \cup \mathbb{T}) \,.\, \mathbf{t}(\gamma) = \beta\} \cup \\
& \{(\mathbf{s}(\alpha), \mathbf{s}(\alpha)^-), (\mathbf{s}(\alpha)^-, \alpha^\vee), (\alpha, \alpha^-), (\alpha^-, \alpha^\vee), (\alpha^\vee, \mathbf{t}(\alpha)) \mid \alpha \in \mathbb{R} \ \wedge \exists\, \gamma \in (\mathbb{R} \cup \mathbb{T}) \,.\, \mathbf{t}(\gamma) = \alpha\}.
\end{aligned}$$

Definition 17 *(Optimized AF for ASAF).* Let $\Delta = \langle \mathbb{A}, \mathbb{R}, \mathbb{T} \rangle$ be an ASAF. The *AF for* Δ is $\Delta_{AF} = \langle \mathsf{A}, \mathsf{R} \rangle$, where:

$$\begin{aligned}
\mathsf{A} = {} & \mathbb{A} \cup \{\alpha, \mathbf{s}(\alpha)^- \mid \alpha \in \mathbb{R}\} \cup \\
& \{\beta, \beta^\vee \mid \beta \in \mathbb{T}\} \cup \\
& \{\beta^- \mid \beta \in \mathbb{T} \wedge \exists \gamma \in (\mathbb{R} \cup \mathbb{T}) \,.\, \mathbf{t}(\gamma) = \beta\}
\end{aligned}$$

$$\begin{aligned}
\mathsf{R} = {} & \{(\mathbf{s}(\alpha), \mathbf{s}(\alpha)^-), (\mathbf{s}(\alpha)^-, \alpha), (\alpha, \mathbf{t}(\alpha)) \mid \alpha \in \mathbb{R}\} \cup \\
& \{(\mathbf{s}(\beta), \beta^\vee), (\beta^\vee, \mathbf{t}(\beta)) \mid \beta \in \mathbb{T}\} \cup \\
& \{(\beta, \beta^-), (\beta^-, \beta^\vee), \mid \beta \in \mathbb{T} \wedge \exists \gamma \in (\mathbb{R} \cup \mathbb{T}) \,.\, \mathbf{t}(\gamma) = \beta\}.
\end{aligned}$$

[10] With a little abuse of notation, $\mathbf{s}(\mathbb{R}) = \bigcup_{\alpha \in \mathbb{R}} \mathbf{s}(\alpha)$.

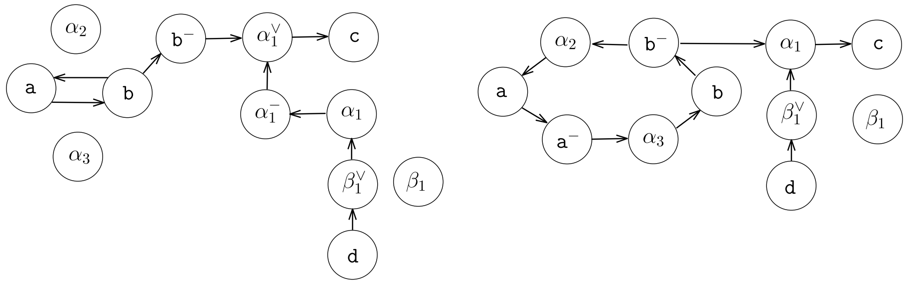


Fig. 8. AF for the RAFN (resp. ASAF) Δ of Example 12 on the left-hand side (resp. right-hand side), obtained by applying Definition 16 (resp. Definition 17).

Example 15. Continuing with Example 12, the left-hand side (resp. right-hand side) of Fig. 8 illustrates the AF for the RAFN (resp. ASAF) Δ obtained by using Definition 16 (resp. Definition 17). Observe that the number of arguments and attacks in Fig. 8 is significantly smaller than that of Fig. 7, where Definitions 12 and 13 are used. $\square$

As stated by the following Proposition, the complete, preferred, stable, and grounded extensions of the AFs for a HOAFN Δ obtained by applying Definitions 12 and 16 (resp. Definitions 13 and 17) are in one-to-one correspondence, as they share the same arguments and relations of Δ, and differ only in the fact that the optimized AF may contain less auxiliary arguments.

Proposition 1. *Let $\Delta = \langle \mathbb{A}, \mathbb{R}, \mathbb{T} \rangle$ be a HOAFN, $\Delta_{AF} = \langle \mathsf{A}, \mathsf{R} \rangle$ be the AF for Δ obtained from Definition 12 or 13, and $\Delta'_{AF} = \langle \mathsf{A}', \mathsf{R}' \rangle$ be the AF for Δ obtained from Definition 16 or 17, respectively. For any semantics $\mathcal{s} \in \{co, pr, st, gr\}$, it holds that:*

- $\mathcal{E}_{\mathcal{s}}(\Delta_{AF}) = \{\mathbf{HOAFNtoAF}(\mathbf{AFtoHOAFN}(E'_{AF})) \mid E'_{AF} \in \mathcal{E}_{\mathcal{s}}(\Delta'_{AF})\}$*; and*
- $\mathcal{E}_{\mathcal{s}}(\Delta'_{AF}) = \{E_{AF} \cap \mathsf{A}' \mid E_{AF} \in \mathcal{E}_{\mathcal{s}}(\Delta_{AF})\}$.

As a consequence we have that, starting from the $\mathcal{s}$-extensions of the AFs for a HOAFN Δ obtained by applying Definitions 12 and 16 (resp. Definitions 13 and 17), we get the same set of $\mathcal{s}$-extensions of Δ. In other words, we have $\{\mathbf{AFtoHOAFN}(E_{AF}) \mid E_{AF} \in \mathcal{E}_{\mathcal{s}}(\Delta_{AF})\} = \{\mathbf{AFtoHOAFN}(E'_{AF}) \mid E'_{AF} \in \mathcal{E}_{\mathcal{s}}(\Delta'_{AF})\}$.

For the sake of the presentation, in the rest of the paper we will focus on the initial definitions of AF for HOAFN (namely, Definitions 12 and 13), although the optimization of the mappings (*i.e.*, Definitions 16 and 17) is considered in the implementation of the incremental algorithm (Algorithm 4) presented in Section 5.

4. Dynamics: updates for HOAFN

In this section we address the issue of dynamics in the context of a HOAFN, which can either be an ASAF or a RAFN. We distinguish between elementary updates, that is adding or deleting a single element, and (complex) updates which may involve the addition or deletion of multiple attacks and supports. Elementary updates consist of the addition or deletion of one "isolated" argument, or one attack or support that is not attacked or supported by other elements of the framework.

4.1. Interaction set and updates

A (complex) update can be, for instance, the deletion of an argument a not necessarily isolated, or of an attack/support γ which can be in turn attacked or supported. Clearly, the process of deleting an element is iterative and, when we delete an element ϑ, all attacks/supports having ϑ as source or target must be deleted as well. Regarding the addition operation, we consider the addition of any set of elements such that after they have been added to the HOAFN, they can be removed by deleting only one element. To this end, we introduce the concept of interaction set.

Definition 18 *(Interaction set)*. Let $\Delta = \langle \mathbb{A}, \mathbb{R}, \mathbb{T} \rangle$ be a HOAFN and $\vartheta \in (\mathbb{A} \cup \mathbb{R} \cup \mathbb{T})$. The interaction set of ϑ w.r.t. Δ (denoted $\overline{\vartheta}_\Delta$) is the (minimal) set satisfying the following condition:

$$\overline{\vartheta}_\Delta = \{\vartheta\} \cup \begin{cases} \bigcup_{\gamma \in (\mathbb{R} \cup \mathbb{T}) \wedge \mathbf{t}(\gamma) = \vartheta} \overline{\gamma}_\Delta & \text{if } \vartheta \in (\mathbb{R} \cup \mathbb{T}) \\ \bigcup_{\gamma \in (\mathbb{R} \cup \mathbb{T}) \wedge (\mathbf{t}(\gamma) = \vartheta \vee \mathbf{s}(\gamma) = \vartheta)} \overline{\gamma}_\Delta & \text{if } \vartheta \in \mathbb{A} \end{cases}$$

Thus, $\overline{\vartheta}_\Delta$ is the minimal set of elements that must be deleted after deleting ϑ so that the resulting HOAFN is consistent. Consequently, in the resulting HOAFN there is no attack/support γ such that its target $\mathbf{t}(\gamma)$ and source $\mathbf{s}(\gamma)$ are not in the HOAFN. By definition, $\overline{\vartheta}_\Delta$ can contain at most one argument. On the one hand, when ϑ is an argument, $\overline{\vartheta}_\Delta$ contains exactly one argument; otherwise, it contains only attacks and supports. $\overline{\vartheta}_\Delta$ will be simply denoted by $\overline{\vartheta}$ whenever Δ is understood. A set $C \subseteq (\mathbb{A} \cup \mathbb{R} \cup \mathbb{T})$ is said to be an interaction set if there exists some $\vartheta \in C$ such that $C = \overline{\vartheta}$, *i.e.* C is the interaction set of ϑ.

Fig. 9. Interaction graphs $\mathcal{G}_{\overline{\beta_0}}$ (left) and $\mathcal{G}_{\overline{\mathtt{d}}}$ (right) of Example 16.

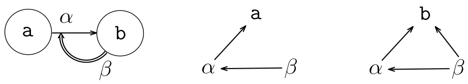


Fig. 10. HOAFN Δ of Example 17 (left) and interaction graphs $\mathcal{G}_{\overline{\mathtt{a}}}$ (center) and $\mathcal{G}_{\overline{\mathtt{b}}}$ (right).

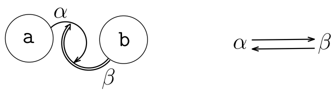


Fig. 11. HOAFN Δ of Example 18 (left) and interaction graph $\mathcal{G}_{\overline{\alpha}}$ (right).

In the following we shall often use the term *interaction graph*, instead of interaction set, for $\overline{\vartheta}$, as it can also be seen as a directed graph $\mathcal{G}_{\overline{\vartheta}} = \langle \overline{\vartheta}, H \rangle$ with $H = \{(\gamma, \mathbf{t}(\gamma)) \mid \gamma \in \overline{\vartheta} \cap (\mathbb{R} \cup \mathbb{T})\} \cup \{(\gamma, \vartheta) \mid \gamma \in \overline{\vartheta} \cap (\mathbb{R} \cup \mathbb{T}) \wedge \mathbf{s}(\gamma) = \vartheta\}$.

Note that whereas H includes an attack from γ to $\mathbf{t}(\gamma)$ whenever both γ and $\mathbf{t}(\gamma)$ belong to $\overline{\vartheta}$, it only includes an attack from γ to $\mathbf{s}(\gamma)$ whenever $\mathbf{s}(\gamma) = \vartheta$. For any interaction set C, we denote by $\mathbf{r}(C)$ the set of elements $\vartheta \in C$ such that $\overline{\vartheta} = C$, also called set of *roots* of C. Observe also that $\overline{\vartheta} \subseteq C$ holds for any element $\vartheta \in C$.

Example 16. Considering the HOAFN Δ shown in Fig. 1 (right) we have that $\overline{\beta_0} = \{\beta_0\} \cup \overline{\alpha_5} \cup \overline{\beta_1} = \{\beta_0, \alpha_5, \beta_1\}$ and $\overline{\mathtt{d}} = \{\mathtt{d}\} \cup \overline{\beta_0} \cup \overline{\alpha_2} \cup \overline{\alpha_3} = \{\mathtt{d}, \beta_0, \alpha_5, \beta_1, \alpha_2, \alpha_3, \alpha_4\}$. Considering the related interaction graphs, we have that $\mathcal{G}_{\overline{\beta_0}} = \langle \{\beta_0, \alpha_5, \beta_1\}, \{(\alpha_5, \beta_0), (\beta_1, \beta_0)\} \rangle$ and $\mathcal{G}_{\overline{\mathtt{d}}} = \langle \{\mathtt{d}, \beta_0, \alpha_5, \beta_1, \alpha_2, \alpha_3, \alpha_4\}, \{(\beta_0, \mathtt{d}), (\alpha_2, \mathtt{d}), (\alpha_3, \mathtt{d}), (\alpha_5, \beta_0), (\beta_1, \beta_0), (\alpha_4, \alpha_3)\} \rangle$, respectively depicted in Fig. 9 (left) and Fig. 9 (right). $\square$

The following example illustrates a case in which the resulting interaction graph is acyclic, but is not a tree.

Example 17. Consider the HOAFN $\Delta = \langle \mathbb{A}, \mathbb{R}, \mathbb{T} \rangle$ shown in Fig. 10 (left) with $\mathbb{A} = \{\mathtt{a}, \mathtt{b}\}$, $\mathbb{R} = \{\alpha\}$ and $\mathbb{T} = \{\beta\}$, where $\mathbf{b}(\alpha) = (\mathtt{a}, \mathtt{b})$ is a first-level attack, and $\mathbf{b}(\beta) = (\mathtt{b}, \alpha)$ is a second-level support. The interaction graph of $\mathtt{a}$ w.r.t. Δ is $\mathcal{G}_{\overline{\mathtt{a}}} = \langle \{\mathtt{a}, \alpha, \beta\}, \{(\alpha, \mathtt{a}), (\beta, \alpha)\} \rangle$, and is shown in Fig. 10 (center). The interaction graph of $\mathtt{b}$ w.r.t. Δ is $\mathcal{G}_{\overline{\mathtt{b}}} = \langle \{\mathtt{b}, \alpha, \beta\}, \{(\alpha, \mathtt{b}), (\beta, \mathtt{b}), (\beta, \alpha)\} \rangle$, and is shown in Fig. 10 (right). $\square$

Next we illustrate a case where the interaction graph is cyclic.

Example 18. Consider the HOAFN $\Delta = \langle \{\mathtt{a}, \mathtt{b}\}, \{\alpha\}, \{\beta\} \rangle$ shown in Fig. 11 (left), with $\mathbf{b}(\alpha) = (\mathtt{a}, \beta)$ and $\mathbf{b}(\beta) = (\mathtt{b}, \alpha)$. The interaction graph $\mathcal{G}_{\overline{\alpha}} = \mathcal{G}_{\overline{\beta}}$, shown in Fig. 11 (right) is cyclic and $\mathbf{r}(\overline{\alpha}) = \overline{\alpha}$. $\square$

Proposition 2. *For any interaction set C we have that:*

1. *if $\mathcal{G}_C$ contains an argument a, then $\mathbf{r}(C) = \{a\}$,*
2. *otherwise, (i) if $\mathcal{G}_C$ is acyclic, then $\mathbf{r}(C)$ is unique, (ii) if $\mathcal{G}_C$ is a (reversed) tree, then $\mathbf{r}(C)$ coincides with the root of $\mathcal{G}_C$, and (iii) if $\mathcal{G}_C$ is cyclic, then $\mathcal{G}_C$ contains only one cycle and $\mathbf{r}(C)$ contains all elements in the cycle.*

Next, we formalize the notion of (complex) update for a HOAFN.

Definition 19. A (complex) *update u* for a HOAFN $\Delta = \langle \mathbb{A}, \mathbb{R}, \mathbb{T} \rangle$ consists of either:

1. the deletion of an element $\vartheta \in (\mathbb{A} \cup \mathbb{R} \cup \mathbb{T})$, denoted by $u = -\,\vartheta$, yielding the updated HOAFN derived from Δ by deleting all elements in $\overline{\vartheta}$; or
2. the addition of a set C of elements, denoted as $u = +\,C$, such that, let $\widehat{\Delta} = \langle \widehat{\mathbb{A}}, \widehat{\mathbb{R}}, \widehat{\mathbb{T}} \rangle$ be the triple derived from Δ by setting $\widehat{\mathbb{A}} = \mathbb{A} \cup (C \cap \mathbb{A}_{\mathbb{U}})$, $\widehat{\mathbb{R}} = \mathbb{R} \cup (C \cap \mathbb{R}_{\mathbb{U}})$ and $\widehat{\mathbb{T}} = \mathbb{T} \cup (C \cap \mathbb{T}_{\mathbb{U}})$, the following conditions hold:
   $i)$ $\widehat{\Delta}$ is a HOAFN,
   $ii)$ $C \cap (\mathbb{A} \cup \mathbb{R} \cup \mathbb{T}) = \emptyset$,
   $iii)$ C is an interaction set w.r.t. $\widehat{\Delta}$.

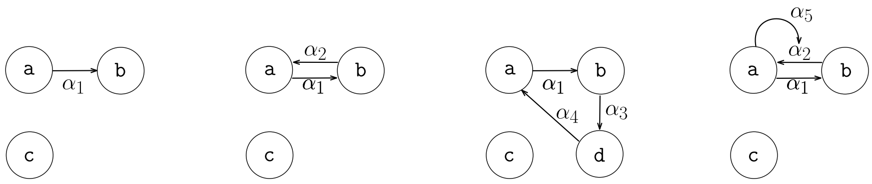


Fig. 12. (left) RAFN Δ, (center-left) $u_1(\Delta)$ and (center-right) $u_2(\Delta)$ of Example 19; (right) RAFN $u_3(\Delta)$ of Example 20.

Basically, Condition 2 in the preceding definition states that: *i*) the resulting HOAFN does not contain cycles with only supports, *ii*) all inserted elements are new, and *iii*) by deleting any element in $\mathbf{r}(C)$ from $\widehat{\Delta}$ (see Item 1) we obtain Δ. As an example, consider the HOAFN Δ discussed in the introduction and depicted in Fig. 1 (right). The update consisting of deleting the support β_0 (resp. argument $\mathtt{d}$) yields the updated HOAFN where all elements in $\overline{\beta_0}$ (resp. $\overline{\mathtt{d}}$), reported in Example 16, are removed from Δ.

Therefore, given a HOAFN $\Delta = \langle \mathbb{A}, \mathbb{R}, \mathbb{T} \rangle$ and an update u for Δ, the resulting (updated) HOAFN is $u(\Delta) = \langle \widehat{\mathbb{A}}, \widehat{\mathbb{R}}, \widehat{\mathbb{T}} \rangle$ where: *i*) for $u = +C$, $\widehat{\mathbb{A}} = \mathbb{A} \cup (C \cap \mathbb{A}_{\mathbb{U}})$, $\widehat{\mathbb{R}} = \mathbb{R} \cup (C \cap \mathbb{R}_{\mathbb{U}})$ and $\widehat{\mathbb{T}} = \mathbb{T} \cup (C \cap \mathbb{T}_{\mathbb{U}})$, whereas *ii*) for $u = -\vartheta$, $\widehat{\mathbb{A}} = \mathbb{A} \setminus \{\vartheta\}$, $\widehat{\mathbb{R}} = \mathbb{R} \setminus \overline{\vartheta}$ and $\widehat{\mathbb{T}} = \mathbb{T} \setminus \overline{\vartheta}$.

Similarly to the case for AFs, we will refer to updates characterized by Item 1 (resp. Item 2) of Definition 19 as 'negative' (resp. 'positive') updates. Moreover, we also say that an update is cyclic (resp. acyclic) if the associated interaction graph is cyclic (resp. acyclic). Sometimes, to simplify the notation, we write $+\vartheta$ instead of $+\{\vartheta\}$ to denote the update adding an element which is neither attacked nor supported, *i.e.* whose interaction set consists of the singleton ϑ.

Next, we identify updates that can be considered as *irrelevant*, since they do not require to recompute the credulous acceptance status of the goal w.r.t. the updated HOAFN.

4.2. Irrelevant updates

It should be noted that not every update has a significant impact on the acceptability of the elements of the HOAFN it is applied on. That is, there are some situations in which an update is *irrelevant* for the CA problem, meaning that: *i*) the answer to the problem for the same goal continues to be "yes", and *ii*) an extension of the updated HOAFN can be obtained (possibly in polynomial time), without requiring the overall recomputation of the credulous acceptance status of the given goal.

Definition 20 *(Irrelevant update).* Given a HOAFN $\Delta = \langle \mathbb{A}, \mathbb{R}, \mathbb{T} \rangle$, a semantics $s \in \{co, pr, st, gr\}$, an s-extension E of Δ, a goal $\varrho \in (\mathbb{A} \cup \mathbb{R} \cup \mathbb{T})$ with $\varrho \in E$, and an update u such that $u(\Delta) = \langle \widehat{\mathbb{A}}, \widehat{\mathbb{R}}, \widehat{\mathbb{T}} \rangle$, we say that u is irrelevant for Δ w.r.t. the extension E and the goal ϱ, if there exists an s-extension $\widehat{E}$ of $u(\Delta)$ such that $\varrho \in \widehat{E}$ and $L_E(\vartheta) = L_{\widehat{E}}(\vartheta)$ for every $\vartheta \in \Big((\mathbb{A} \cap \widehat{\mathbb{A}}) \cup (\mathbb{R} \cap \widehat{\mathbb{R}}) \cup (\mathbb{T} \cap \widehat{\mathbb{T}})\Big)$.

It is important to recall that our purpose is to incrementally check the credulous acceptance of a goal element in an updated HOAFN by exploiting an initial extension. Also, as a byproduct, we give as output an extension of the updated framework containing the goal (if it exists). Thus, our updates are irrelevant not only if the status of the goal does not change, but it is also required that the statuses of all elements not belonging to the interaction set defining the update are maintained. As a result, one could easily define irrelevant updates for goal elements whose status is unaccepted and such that the answer to the CA problem continues to be "no" in the updated HOAFN.

Example 19. Consider the RAFN $\Delta = \langle \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}, \{\alpha_1\}, \{\} \rangle$ reported in Fig. 12 (left), with $\mathbf{b}(\alpha_1) = (\mathtt{a}, \mathtt{b})$ having only one preferred extension $E = \{\mathtt{a}, \mathtt{c}, \alpha_1\}$. Assuming that our goal is $\mathtt{c}$, the update $u_1 = +\alpha_2$, with $\mathbf{b}(\alpha_2) = (\mathtt{b}, \mathtt{a})$, is irrelevant for Δ under the preferred semantics w.r.t. E and goal $\mathtt{c}$, as $E_1 = E \cup \{\alpha_2\}$ is a preferred extension of $u_1(\Delta)$ (see Fig. 12(center-left)). However, the update $u_2 = +\{\mathtt{d}, \alpha_3, \alpha_4\}$, with $\mathbf{b}(\alpha_3) = (\mathtt{b}, \mathtt{d})$ and $\mathbf{b}(\alpha_4) = (\mathtt{d}, \mathtt{a})$, is not irrelevant for Δ under the preferred semantics w.r.t. E and the goal $\mathtt{c}$. This is because, even though $\mathtt{c}$ continues to be credulously accepted in $u_2(\Delta)$ (shown in Fig. 12(center-right)), the updated HOAFN has only one preferred extension $E_2 = \{\mathtt{c}, \alpha_1, \alpha_3, \alpha_4\}$ (*i.e.* arguments $\mathtt{a}, \mathtt{b}, \mathtt{d}$ are undecided). $\square$

As the aim of this section is to identify updates that are irrelevant, hereafter, whenever we have an update $u = -\vartheta$ we require that the goal ϱ is not part of the elements to be removed, *i.e.*, ϱ does not occur in $\overline{\vartheta}$. In the following we shall use the symbols E and $\widehat{E}$ to denote extensions before and after the update, respectively.

Cyclic updates We first explore the case where the interaction graph associated with the update is cyclic and none of the roots of the interaction graph is an argument. In such a case, as shown by the following theorem, the update is irrelevant whenever the updated HOAFN admits some extension.

Theorem 2. *Given a HOAFN $\Delta = \langle \mathbb{A}, \mathbb{R}, \mathbb{T} \rangle$, a semantics $s \in \{co, pr, st, gr\}$, an s-extension E of Δ, a goal $\varrho \in E$, and a cyclic update $u = -\gamma$ (resp. $u = +C$) such that $\gamma \notin \mathbb{A}$ (resp. $\mathbf{r}(C) \cap \mathbb{A}_{\mathbb{U}} = \emptyset$) and $\varrho \notin \overline{\gamma}$. Then u is irrelevant w.r.t. E and ϱ iff (i) u is negative or $s \in \{co, pr, gr\}$, or (ii) u is positive, $s = st$, and there exists either $\alpha \in (C \cap \mathbb{R}_U)$ s.t. $\mathbf{v}_E(\mathbf{s}(\alpha)) = \mathtt{false}$ or $\beta \in (C \cap \mathbb{T}_U)$ s.t. $\mathbf{v}_E(\mathbf{s}(\alpha)) = \mathtt{true}$.*

As for any semantics $\mathcal{s} \in \{co, pr, gr\}$ the existence of at least one extension is guaranteed, cyclic updates (whose root is not an argument) are always irrelevant. Clearly, even if u is irrelevant, we have to compute an extension for $u(\Delta)$, for instance, by restricting (when $u = -\vartheta$) or extending (when $u = +C$) the given extension E. In particular, for a negative cyclic update $u = -\gamma$ with $\gamma \notin \mathbb{A}$ and $\varrho \notin \overline{\gamma}$, we have that for all semantics $\mathcal{s} \in \{co, pr, st, gr\}$, u is irrelevant and $\widehat{E} = E \setminus \overline{\gamma}$ is an $\mathcal{s}$-extension of $u(\Delta)$.

For a positive update $u = +C$ the computation is not as straightforward. Given a HOAFN $\Delta = \langle \mathbb{A}, \mathbb{R}, \mathbb{T} \rangle$, a semantics $\mathcal{s} \in \{co, pr, st, gr\}$, an $\mathcal{s}$-extension E of Δ and a cyclic irrelevant update $+C$ such that $(\mathbf{r}(C) \cap \mathbb{A}_{\mathbb{U}}) = \emptyset$, let us denote with (i) $\Delta_u = \langle \mathbb{A}_u, \mathbb{R}_u, \mathbb{T}_u \rangle$ the HOAFN such that $\mathbb{R}_u = (C \cap \mathbb{R}_{\mathbb{U}})$, $\mathbb{T}_u = (C \cap \mathbb{T}_{\mathbb{U}})$ and $\mathbb{A}_u = \{\mathbf{s}(\gamma) | \gamma \in (\mathbb{R}_u \cup \mathbb{T}_u)\}$, and (ii) Λ_u the AF derived from the AF for Δ_u by (1) deleting every unattacked argument a such that $\mathbf{v}_E(a) = \texttt{false}$ and (2) adding self-attack loops (a, a) for every unattacked argument such that $\mathbf{v}_E(a) = \texttt{unknown}$. Then, we have the following result.

Theorem 3. *Given a HOAFN $\Delta = \langle \mathbb{A}, \mathbb{R}, \mathbb{T} \rangle$, a semantics $\mathcal{s} \in \{co, pr, st, gr\}$, an $\mathcal{s}$-extension E of Δ, a goal $\varrho \in E$, a cyclic irrelevant update $u = -\gamma$ (resp. $u = +C$) w.r.t. E, and ϱ such that $\gamma \notin \mathbb{A}$ (resp. $(\mathbf{r}(C) \cap \mathbb{A}_{\mathbb{U}}) = \emptyset$) and $\varrho \notin \overline{\gamma}$. Then, $\widehat{E} = E \setminus \overline{\gamma}$ if $u = -\gamma$ (resp. $\widehat{E} = E \cup$ **AFtoHOAFN**(E') if $u = +C$, where E' is an $\mathcal{s}$-extension of Λ_u), is an $\mathcal{s}$-extension of $u(\Delta)$.*

Therefore, for a positive update $u = +C$, first the HOAFN Δ_u is derived. Next, by mapping Δ_u to an AF and applying the above steps (1) and (2) the AF Λ_u is obtained; then, a σ-extension E' for Λ_u is computed. Finally, an extension of Δ_u is derived from E' using the function **AFtoHOAFN**. Observe that, for the stable semantics, if Δ_u does not have extensions then the update is not irrelevant and for checking the credulous acceptance of the goal ϱ the whole updated HOAFN $u(\Delta)$ must be considered.

Acyclic updates We next consider the case where the interaction graph associated with the update is acyclic. To this end, the following definition introduces the notion of 'temporary' valuation, that will be exploited in order to identify irrelevant updates (cf. Theorem 5). Recall that the concept of valuation $\mathbf{v}$ was introduced after that of labeling at the end of Section 2.3.3.

Definition 21. Given a HOAFN $\Delta = \langle \mathbb{A}, \mathbb{R}, \mathbb{T} \rangle$, a semantics $\mathcal{s} \in \{co, pr, st, gr\}$, an $\mathcal{s}$-extension E of Δ, and an acyclic update $u = +C$ (*i.e.*, $\mathcal{G}_C$ is acyclic), the temporary valuation $\mathbf{v}'$ for $u(\Delta)$ w.r.t. E (and u) is defined as follows:

- $\mathbf{v}'(\vartheta) = \mathbf{v}(\vartheta)$ for any $\vartheta \in (\mathbb{A} \cup \mathbb{R} \cup \mathbb{T})$;
- $\mathbf{v}'(\vartheta)$ is obtained by evaluating the logical formula of Equation (1) (resp. Equation (2)) for any $\vartheta \in C$ under the RAFN (resp. ASAF) interpretation.

$$\mathbf{v}'(\vartheta) = \bigwedge_{\substack{\alpha \in (\mathbb{R}_{\mathbb{U}} \cap C) \\ \wedge\, \mathbf{t}(\alpha) = \vartheta}} \Big(\neg \mathbf{v}'(\alpha) \vee \neg \mathbf{v}'(\mathbf{s}(\alpha)) \Big) \wedge \bigwedge_{\substack{\beta \in (\mathbb{T}_{\mathbb{U}} \cap C) \\ \wedge\, \mathbf{t}(\beta) = \vartheta}} \Big(\neg \mathbf{v}'(\beta) \vee \mathbf{v}'(\mathbf{s}(\beta)) \Big) \tag{1}$$

$$\mathbf{v}'(\vartheta) = \bigwedge_{\vartheta \in (\mathbb{R}_{\mathbb{U}} \cap C)} \mathbf{v}'(\mathbf{s}(\vartheta)) \wedge \bigwedge_{\substack{\alpha \in (\mathbb{R}_{\mathbb{U}} \cap C) \\ \wedge\, \mathbf{t}(\alpha) = \vartheta}} \neg \mathbf{v}'(\alpha) \wedge \bigwedge_{\substack{\beta \in (\mathbb{T}_{\mathbb{U}} \cap C) \\ \wedge\, \mathbf{t}(\beta) = \vartheta}} \Big(\neg \mathbf{v}'(\beta) \vee \mathbf{v}'(\mathbf{s}(\beta)) \Big) \tag{2}$$

The aim of the temporary valuation is to consistently expand the $\mathcal{s}$-extension E to new elements in the added interaction set C when $+C$ is an irrelevant update. The intuition behind Equations (1) and (2) is to evaluate elements ϑ in the interaction graph $\mathcal{G}_C$. Thus, elements in C are evaluated according to the RAFN or ASAF interpretation, following the topological order defined over C (*i.e.* the precedence relation defined by $\mathcal{G}_C$). For instance, in the case of RAFN, the status of an element $\vartheta \in C$ is $\texttt{true}$ if both (i) for every attack α s.t. $\mathbf{t}(\alpha) = \vartheta$, the temporary valuation of either α or $\mathbf{s}(\alpha)$ is $\texttt{false}$, and (ii) for every support β s.t. $\mathbf{t}(\beta) = \vartheta$, either the temporary valuation of β is $\texttt{false}$ or the temporary valuation of $\mathbf{s}(\beta)$ is $\texttt{true}$.

The following theorem tells us how to compute an $\mathcal{s}$-extension $\widehat{E}$ of the updated HOAFN $u(\Delta)$ in the presence of an irrelevant update u.

Theorem 4. *Given a HOAFN $\Delta = \langle \mathbb{A}, \mathbb{R}, \mathbb{T} \rangle$, a semantics $\mathcal{s} \in \{co, pr, st, gr\}$, an $\mathcal{s}$-extension E of Δ, a goal $\varrho \in E$, and a negative (resp. positive) update $u = -\vartheta$ (resp. $u = +C$) which is acyclic and irrelevant w.r.t. E and ϱ, and $\varrho \notin \overline{\vartheta}$. Let $u(\Delta) = \langle \widehat{\mathbb{A}}, \widehat{\mathbb{R}}, \widehat{\mathbb{T}} \rangle$ be the updated HOAFN. Then, $\widehat{E} = E \cap (\widehat{\mathbb{A}} \cup \widehat{\mathbb{R}} \cup \widehat{\mathbb{T}})$ (resp. $\widehat{E} = \{\varphi \mid \varphi \in (\widehat{\mathbb{A}} \cup \widehat{\mathbb{R}} \cup \widehat{\mathbb{T}}) \wedge \mathbf{v}'(\varphi) = \texttt{true}\}$, where $\mathbf{v}'$ is the temporary valuation for $u(\Delta)$ w.r.t. E) is an $\mathcal{s}$-extension of $u(\Delta)$ containing ϱ.*

Example 20. Let Δ be the RAFN of Example 19, and $u_3 = +\{\alpha_2, \alpha_5\}$, with $\mathbf{b}(\alpha_2) = (\mathrm{b}, \mathrm{a})$ and $\mathbf{b}(\alpha_5) = (\mathrm{a}, \alpha_2)$. First, recalling that the goal is c and $E = \{\mathrm{a}, \mathrm{c}, \alpha_1\}$ is the only preferred extension of Δ, note that u_3 is irrelevant for Δ under the preferred semantics w.r.t. E and c since $\widehat{E} = E \cup \{\alpha_5\}$ is a preferred extension of $u_3(\Delta)$ (shown in Fig. 12 (right)). Then, let $\mathbf{v}'$ be the temporary valuation for $u_3(\Delta)$ under the preferred semantics w.r.t. E, $\mathbf{v}'(\alpha_5) = \texttt{true}$ as it is neither attacked nor supported, and $\mathbf{v}'(\alpha_2) = \neg \mathbf{v}'(\alpha_5) \vee \neg \mathbf{v}'(\mathrm{a}) = \texttt{false} \vee \texttt{false} = \texttt{false}$. □

Observe that under the RAFN semantics, the temporary valuation of all attacks and supports which are associated with leaf nodes (*i.e.* nodes without incoming edges) in the interaction graph is $\texttt{true}$ (*i.e.* their label is $\texttt{in}$); in contrast, under the ASAF semantics, attacks take the same value as their source argument.

Table 1
Cases in which the update $u = -\gamma$, with $\gamma \in (\mathbb{R} \cup \mathbb{T})$, is irrelevant. $\mathbf{V_s}(\gamma) = \mathbf{v}(\mathbf{s}(\gamma)) \wedge \mathbf{v}(\gamma)$ if $\gamma \in \mathbb{R}$, $\mathbf{V_s}(\gamma) = \neg\mathbf{v}(\mathbf{s}(\gamma)) \wedge \mathbf{v}(\gamma)$ if $\gamma \in \mathbb{T}$, and $\mathbf{V_t}(\gamma) = \mathbf{v}(\mathbf{t}(\gamma))$.

Update $u = -\gamma$		$\mathbf{V_t}(\gamma)$		
		`true`	`unknown`	`false`
$\mathbf{V_s}(\gamma)$	`true`	N/A	N/A	
	`unknown`	N/A		co,pr,gr
	`false`	co,pr,st,gr	co,pr,gr	co,pr,st,gr

Table 2
Cases in which the update $u = +C$, with $\gamma = \mathbf{r}(C) \in (\mathbb{R}_{\mathbb{U}} \cup \mathbb{T}_{\mathbb{U}})$, is irrelevant. $\mathbf{V_s}(\gamma) = \mathbf{v}'(\mathbf{s}(\gamma)) \wedge \mathbf{v}'(\gamma)$ if $\gamma \in \mathbb{R}_{\mathbb{U}}$, $\mathbf{V_s}(\gamma) = \neg\mathbf{v}'(\mathbf{s}(\gamma)) \wedge \mathbf{v}'(\gamma)$ if $\gamma \in \mathbb{T}_{\mathbb{U}}$, and $\mathbf{V_t}(\gamma) = \mathbf{v}'(\mathbf{t}(\gamma))$.

Update $u = +C$		$\mathbf{V_t}(\gamma)$		
		`true`	`unknown`	`false`
$\mathbf{V_s}(\gamma)$	`true`			co,pr,st,gr
	`unknown`		co,gr	co,pr,gr
	`false`	co,pr,st	co,pr,gr	co,pr,st,gr

The following theorem tells us, for each kind of update where arguments are neither added nor deleted, for each combination of the status valuation of the source and target elements of the root of the interaction set (in the case of negative updates), and for each semantics, whether the update is irrelevant or not for a HOAFN w.r.t. an initial extension.

Theorem 5. *Let Δ be a HOAFN, $s \in \{co, pr, st, gr\}$ a semantics, E an s-extension of Δ, $\varrho \in E$ a goal, and $u = -\gamma$ (resp. $u = +C$) an acyclic update for Δ such that $\gamma \notin \mathbb{A}$ (resp. $\mathbf{r}(C) \notin \mathbb{A}_{\mathbb{U}}$) and $\varrho \notin \overline{\gamma}$. Also, let $\mathbf{v}$ be the valuation for Δ w.r.t. E, and $\mathbf{v}'$ be the temporary valuation for $u(\Delta)$ w.r.t. E and u. Then, u is irrelevant for Δ w.r.t. the s-extension E and the goal ϱ if the semantics s occurs in the cell identified by row $\mathbf{V_s}(\gamma)$ and column $\mathbf{V_t}(\gamma)$ of Table 1 (resp. Table 2).*

In the previous theorem, for both negative and positive updates (*i.e.* $u = -\gamma$ or $u = +C$ and $\mathbf{r}(C) = \gamma$), $\mathbf{V_s}(\gamma)$ takes into account the status of $\mathbf{s}(\gamma)$ and γ, whereas $\mathbf{V_t}(\gamma)$ takes into account the status of $\mathbf{t}(\gamma)$. Thus, irrelevant updates are identified by looking at the temporary valuation $\mathbf{v}'$ for positive updates, and by looking at the valuation $\mathbf{v}$ for negative updates. It is worth noting that, for any of the cases of update considered, if a semantics does not appear in a cell of Tables 1-2 then it means that there exists a counter-example proving that such irrelevant conditions cannot exist. N/A in a cell of Tables 1-2 means not applicable (*e.g.* in Table 1 it cannot be the case that $u = -\alpha$ with $\alpha \in \mathbb{R}$, $\mathbf{v}(\mathbf{s}(\alpha)) = \texttt{true}$ and $\mathbf{v}(\mathbf{t}(\alpha)) = \texttt{true}$). Tables 1 and 2 generalize similar results for AFs [49] to HOAFNs, where the target of an interaction can also be an attack or a support.

The following theorem checks if an update consisting of the deletion of an argument is irrelevant.

Theorem 6. *Let $\Delta = \langle \mathbb{A}, \mathbb{R}, \mathbb{T} \rangle$ be a HOAFN, $s \in \{co, pr, st, gr\}$ a semantics, E an s-extension of Δ, $\varrho \in E$ a goal, and $u = -a$ with $a \in \mathbb{A}$ an acyclic update for Δ such that $\varrho \notin \overline{a}$. Then, u is irrelevant for Δ w.r.t. E and ϱ if, for every $\gamma \in \overline{a}$ such that $\mathbf{s}(\gamma) = a$, it holds that the update $-\gamma$ is irrelevant for Δ w.r.t. E and ϱ as per Theorem 5.*

The previous theorem states that the update $u = -a$, with a being an argument, is irrelevant if every update consisting in the deletion of an attack/support having a as source is irrelevant.

The following corollary states that a negative update removing an argument is irrelevant if the argument had no impact on the acceptance status of all other elements in the HOAFN, either because its valuation was `false` (labeled `out`) and only originated attacks, or because its valuation was `true` (labeled `in`) and only originated supports.

Corollary 1. *Let $\Delta = \langle \mathbb{A}, \mathbb{R}, \mathbb{T} \rangle$ be a HOAFN, $s \in \{co, pr, st, gr\}$ a semantics, E an s-extension of Δ, $\varrho \in E$ a goal, $u = -a$ with $a \in \mathbb{A}$ an acyclic update for Δ such that $\varrho \notin \overline{a}$, and $\Gamma = \{\gamma \mid \gamma \in \overline{a} \wedge \mathbf{s}(\gamma) = a\}$. Then, u is irrelevant for Δ w.r.t. E and ϱ if either the condition (i) $\mathbf{v}_E(a) = \texttt{false}$ and $\Gamma \subseteq \mathbb{R}$, or (ii) $\mathbf{v}_E(a) = \texttt{true}$ and $\Gamma \subseteq \mathbb{T}$ holds.*

Similarly to Theorem 6, the following theorem checks if an update consisting of the addition of an interaction set whose root is an argument is irrelevant. In particular, it states that the update $+C$ can be split into two parts $+(C \setminus C')$ and $+C'$ and apply them in sequence; C' denotes the union of interactions sets $\overline{\gamma}$ with γ being an attack or a support having the argument $\mathbf{r}(C)$ as source.

Theorem 7. *Let $\Delta = \langle \mathbb{A}, \mathbb{R}, \mathbb{T} \rangle$ be a HOAFN, $s \in \{co, pr, st, gr\}$ a semantics, E an s-extension of Δ, $\varrho \in E$ a goal, and $u = +C$ with $\mathbf{r}(C) \in \mathbb{A}_{\mathbb{U}}$ an acyclic update for Δ. Let $C' = \bigcup_{\gamma \in C \wedge \mathbf{s}(\gamma) = \mathbf{r}(C)} \overline{\gamma}_{u(\Delta)}$, $u' = +(C \setminus C')$, $\Delta' = u'(\Delta) = \langle \mathbb{A}', \mathbb{R}', \mathbb{T}' \rangle$, $\mathbf{v}'$ be the temporary valuation for Δ' w.r.t. E and u', and $E' = \{\vartheta \mid \vartheta \in (\mathbb{A}' \cup \mathbb{R}' \cup \mathbb{T}') \wedge \mathbf{v}'(\vartheta) = \texttt{true}\}$ be the s-extension of Δ' expanding E (obtained as in Theorem 4). Then, u is irrelevant for Δ w.r.t. E and ϱ if for every $\gamma \in C'$ the update $+\overline{\gamma}_{u(\Delta)}$ is irrelevant for Δ' w.r.t. E' and ϱ.*

It is important to note that in the previous theorem, since the interaction graph $\mathcal{G}_C$ is acyclic, there is only one extension of Δ'.

The next corollary considers a special case of the previous theorem.

Corollary 2. *Let $\Delta = \langle \mathbb{A}, \mathbb{R}, \mathbb{T} \rangle$ be a HOAFN, $\mathscr{s} \in \{co, pr, st, gr\}$ a semantics, E an $\mathscr{s}$-extension of Δ, $\varrho \in E$ a goal, and $u = +C$ with $\mathbf{r}(C) \in \mathbb{A}$ an acyclic update for Δ. Then, u is irrelevant for Δ w.r.t. E and ϱ if $\mathbf{v}_E(\mathbf{t}(\gamma)) = \texttt{false}$ for every $\gamma \in C$ such that $\mathbf{s}(\gamma) = \mathbf{r}(C)$.*

The following proposition states that, if u is irrelevant, then computing an updated extension containing the goal element can be done in polynomial time in several cases.

Proposition 3. *Let Δ be a HOAFN, $\mathscr{s} \in \{co, pr, st, gr\}$ a semantics, u an irrelevant update for Δ w.r.t. a given $\mathscr{s}$-extension E, and $\varrho \in E$ a goal. Then, for u acyclic or negative, or $\mathscr{s} \in \{co, gr\}$, the computation of an $\mathscr{s}$-extension $\widehat{E}$ of $u(\Delta)$ such that $\varrho \in \widehat{E}$ can be done in polynomial time.*

Thus, for the cases considered in the preceding proposition, an $\mathscr{s}$-extension $\widehat{E}$ of the updated HOAFN $u(\Delta)$ containing ϱ can be deterministically computed in polynomial time. For the other cases, an extension of $u(\Delta)$ expanding E must be computed non-deterministically. That is, we have to compute an $\mathscr{s}$-extension E' for Δ_u and merge E and E'. This allows us to restrict the computation to Δ_u, whose size is possibly smaller than that of $u(\Delta)$.

5. Computing credulous acceptance of updated HOAFNs

In this section we present an incremental approach for the efficient computation of the *Credulous Acceptance Problem* in dynamic HOAFNs.

This problem has been addressed in the context of dynamic AFs and several "incremental" solvers have been proposed [62]. These solvers, in addition to the boolean answer, give as output an extension containing the goal (if such an extension exists). In the same vein, we exploit this information for the efficient recalculation of the status of credulous acceptance of the goal element, which can either be an argument, an attack or a support from a HOAFN.

In the static setting, $\text{CA}_{\mathscr{s}}$ is the following problem: given a HOAFN $\Delta = \langle \mathbb{A}, \mathbb{R}, \mathbb{T} \rangle$, a goal element $\varrho \in (\mathbb{A} \cup \mathbb{R} \cup \mathbb{T})$ and a semantics $\mathscr{s}$, compute an $\mathscr{s}$-extension E of Δ with $\varrho \in E$, if such an extension exists (*i.e.*, ϱ is credulously accepted); $\perp$ otherwise (*i.e.*, ϱ is not credulously accepted in Δ under $\mathscr{s}$). In the dynamic setting, $\text{CA}_{\mathscr{s}}$ is the following problem: given a HOAFN Δ, a goal element ϱ, an $\mathscr{s}$-extension E of Δ containing ϱ, and an update u, determine an $\mathscr{s}$-extension $\widehat{E}$ of $u(\Delta)$ containing ϱ, if such an extension exists; $\perp$ otherwise.

We first present the algorithm solving the problem in the static setting (also called computation from scratch in what follows), and then present our incremental algorithm which solves the problem incrementally in a dynamic setting with the aim of minimizing the wasted effort. Section 6 is devoted to the empirical analysis comparing the two algorithms. As no direct HOAFN solver exists, we exploit the translations presented in Section 3. In the following, we denote by AFS an AF solver, and by $\text{AFS}^{\text{CA}}_{\mathscr{s}}$ a function solving the $\text{CA}_{\mathscr{s}}$ problem for AF, *i.e.* a function receiving as input an AF, a semantics $\mathscr{s}$ and a goal argument g, and returning an $\mathscr{s}$-extension containing g, if such an extension exists; $\perp$ otherwise.

5.1. Computing from scratch

The algorithm for the computation from scratch (Algorithm 2) receives as input a HOAFN Δ, a goal element $\varrho \in \mathbb{A} \cup \mathbb{R} \cup \mathbb{T}$, a semantics $\mathscr{s}$, and the name of an AF solver AFS (making available the function $\text{AFS}^{\text{CA}}_{\mathscr{s}}$ for the task $\text{CA}_{\mathscr{s}}$), and gives as output an $\mathscr{s}$-extension containing ϱ, if there exists one, $\perp$ otherwise. The algorithm works as follows. It first rewrites the input HOAFN Δ into the AF Δ_{AF} (Line 1). Then it computes the $\text{CA}_{\mathscr{s}}$ problem for the AF Δ_{AF} and goal ϱ under semantics $\mathscr{s}$ by means of function $\text{AFS}^{\text{CA}}_{\mathscr{s}}$ which returns an $\mathscr{s}$-extension E_{AF} containing ϱ if ϱ is credulously accepted, $\perp$ otherwise (Line 2). Next, if the goal element ϱ is credulously accepted w.r.t. $\mathscr{s}$ and Δ (*i.e.*, $\varrho \in E$ or, equivalently, $E_{AF} \neq \perp$), from the extension E_{AF} of Δ_{AF} the extension E of Δ is derived using function **AFtoHOAFN**; in this case, extension E is given as output (Line 4). Otherwise, $\perp$ is returned, stating that ϱ is not credulously accepted w.r.t. $\mathscr{s}$ and Δ (Line 5).

Algorithm 2 SCRATCH-HOAFN-CA($\Delta, \varrho, \mathscr{s}$, AFS).

Input: HOAFN $\Delta = \langle \mathbb{A}, \mathbb{R}, \mathbb{T} \rangle$,
goal element $\varrho \in \mathbb{A} \cup \mathbb{R} \cup \mathbb{T}$,
semantics $\mathscr{s}$,
AF solver AFS for the task CA.

Output: An $\mathscr{s}$-extension $E \in \mathcal{E}_{\mathscr{s}}(\Delta)$ with $\varrho \in E$, if there exists one; $\perp$ otherwise.

1: Let $\Delta_{AF} = \langle \mathsf{A}, \mathsf{R} \rangle$ be the AF for Δ;
2: Let $E_{AF} = \text{AFS}^{\text{CA}}_{\mathscr{s}}(\Delta_{AF}, \varrho)$;
3: **if** $E_{AF} \neq \perp$ **then**
4:     **return** $E = \mathbf{AFtoHOAFN}_{\Delta_{AF}}(E_{AF})$;
5: **return** $\perp$;

Algorithm 3 INCR-AF-CA($\Lambda, u, g, \mathscr{s}, E$, AFS).

Input: AF $\Lambda = \langle \mathsf{A}, \mathsf{R} \rangle$,
set u of updates for Λ,
goal argument $g \in \mathsf{A}$,
semantics $\mathscr{s}$,
extension $E \in \mathcal{E}_{\mathscr{s}}(\Lambda)$,
AF solver AFS for the tasks CA and SE.
Output: An $\mathscr{s}$-extension $\widehat{E} \in \mathcal{E}_{\mathscr{s}}(u(\Lambda))$ with $g \in \widehat{E}$, if there exists one; $\bot$ otherwise.
1: $I = \mathbf{inf}(u, \Lambda, E)$;
2: **if** $(I = \emptyset)$ **then**
3: **return** $\widehat{E} = E$;
4: $\langle \mathsf{A}_d, \mathsf{R}_d \rangle = \mathbf{red}(u, \Lambda, E)$;
5: **if** $g \in I$ **then**
6: $E_d = \text{AFS}_{\mathscr{s}}^{\mathsf{CA}}(\langle \mathsf{A}_d, \mathsf{R}_d \rangle, g)$;
7: **else**
8: $E_d = \text{AFS}_{\mathscr{s}}^{\mathsf{SE}}(\langle \mathsf{A}_d, \mathsf{R}_d \rangle)$;
9: **if** $(E_d \neq \bot)$ **then**
10: $\widehat{E} = (E \setminus I) \cup E_d$;
11: **if** $(g \in \widehat{E})$ **then**
12: **return** $\widehat{E}$;
13: **return** $\text{AFS}_{\mathscr{s}}^{\mathsf{CA}}(u(\Lambda), g)$;

5.2. Incremental algorithm for AF

As said before, since no direct HOAFN solver exists, we first present Algorithm 3 (named INCR-AF-CA) that incrementally solves the problem for an AF and is used as a subroutine in Algorithm 4 that incrementally solves the CA problem for HOAFN. INCR-AF-CA takes as input the same parameters as INCR-AF-SE (cf. Algorithm 1) plus a goal argument g. Here, AFS is an AF solver that makes available the two functions $\text{AFS}_{\mathscr{s}}^{\mathsf{CA}}$ and $\text{AFS}_{\mathscr{s}}^{\mathsf{SE}}$ solving the tasks CA and SE for any semantics $\mathscr{s}$.

As in the case of INCR-AF-SE, INCR-AF-CA first identifies the influenced set and the reduced AF w.r.t. u and E (Lines 1-4). Then it computes an $\mathscr{s}$-extension E_d of the reduced AF by using $\text{AFS}_{\mathscr{s}}^{\mathsf{CA}}$ if the goal argument g is in the influenced set, in which case $g \in E_d$ (Line 5); otherwise, $\text{AFS}_{\mathscr{s}}^{\mathsf{SE}}$ is used to compute any $\mathscr{s}$-extension E_d of the reduced AF independently of the given goal argument (Line 8).[11] This difference in the use of the auxiliary functions is due to the fact that we need to incrementally maintain an extension to be exploited in future updates. Next, if an extension E_d has been computed, E_d is combined with the part of the initial extension E which is not influenced by the update operation, to obtain an extension $\widehat{E}$ of $u(\Delta)$ (Line 10). Finally, this extension is then given as output (Line 12) if it contains the goal argument g; otherwise, the function solving the CA problem from scratch on the whole updated AF $u(\Lambda)$ is called (Line 13). The last case might occur when either the reduced AF $\langle \mathsf{A}_d, \mathsf{R}_d \rangle$ does not admit extensions (under the stable semantics) or the goal does not belong to the extension E_d that was incrementally computed.

5.3. Incremental algorithm for HOAFN

We now present our incremental algorithm for solving the $\mathsf{CA}_{\mathscr{s}}$ problem for a HOAFN in a dynamic context, *i.e.*, when updates are applied to an initial HOAFN. The underlying idea, implemented in Algorithm 4, is to compute an extension $\widehat{E}$ of the updated HOAFN $u(\Delta)$ containing the goal element ϱ (if such an extension exists) by exploiting an $\mathscr{s}$-extension E of Δ. For the sake of simplicity, Algorithm 4 refers to the mappings introduced in Definitions 12 and 13, although it can be easily modified to use the optimized mappings presented in Definitions 16 and 17.[12]

Algorithm 4 first checks if the update is irrelevant, that is if some of the conditions of Theorems 2, 5-7 hold (Line 1). If this is the case, Algorithm 4 returns the updated extension $\widehat{E}$ directly obtained from E according to Theorems 3 and 4 (Line 2). It is worth noting that, as stated in Proposition 3, the computation of $\widehat{E}$ is polynomial in all cases apart from those in which the irrelevant update is positive, cyclic and the semantics is either stable or preferred. Otherwise, the update is not irrelevant and Algorithm 4 computes the AF Δ_{AF} for Δ (Line 3), according to either Definition 12 or Definition 13, and obtains the corresponding extension E_{AF} of Δ_{AF} w.r.t. E (Line 4). Then, if the update is positive, a set of isolated arguments is added to Δ_{AF} (Line 6); these arguments are also added to E_{AF} at Line 7 as they are isolated in Δ_{AF}. The new additional arguments are computed through the function **AdditionalArgs**(u,Δ) which returns the arguments of the AF for the HOAFN $u(\Delta)$ that are not in Δ_{AF}. Then, the set u_{AF} of updates over Δ_{AF} corresponding to u is built (at Line 8) by calling the function **UpdatesOverAF**(u,Δ) which returns attacks (prefixed with +) of the AF for the HOAFN $u(\Delta)$ that are not in Δ_{AF} if u is positive; otherwise, **UpdatesOverAF**(u,Δ) returns attacks (prefixed with −) of Δ_{AF} that are not in the AF for the HOAFN $u(\Delta)$. Next, the function INCR-AF-CA encoded in Algorithm 3 that incrementally solves the CA problem is called, which returns an $\mathscr{s}$-extension $\widehat{E}_{AF}$ of the updated AF $u_{AF}(\Delta_{AF})$ containing the goal ϱ, if there exists one; $\bot$ otherwise. Thus, if function INCR-AF-CA has returned an extension containing the goal ϱ, then the extension $\mathbf{AFtoHOAFN}_{u(\Delta)}(\widehat{E}_{AF})$ is returned (Line 11). As mentioned before, function INCR-AF-CA might invoke function $\text{AFS}_{\mathscr{s}}^{\mathsf{CA}}$ (cf. Algorithm 3, Line 13) to solve CA from scratch over $u_{AF}(\Delta_{AF})$, *i.e.* over the whole updated AF for the HOAFN $u(\Delta)$.

[11] In both cases, if the corresponding extension does not exist, the functions $\text{AFS}_{\mathscr{s}}^{\mathsf{CA}}$ and $\text{AFS}_{\mathscr{s}}^{\mathsf{SE}}$ return $\bot$.

[12] The implementation used in the experiments reported in Section 6 is based on the optimized mappings.

Algorithm 4 DYN-HOAFN-CA($\Delta, u, \varrho, \mathscr{s}, E$, AFS).

Input: HOAFN $\Delta = \langle \mathbb{A}, \mathbb{R}, \mathbb{T} \rangle$,
update u for Δ,
goal element $\varrho \in \mathbb{A} \cup \mathbb{R} \cup \mathbb{T}$,
semantics $\mathscr{s}$,
$\mathscr{s}$-extension E of Δ containing ϱ,
AF solver AFS for the tasks CA and SE.
Output: An $\mathscr{s}$-extension $\widehat{E} \in \mathcal{E}_{\mathscr{s}}(u(\Delta))$ with $\varrho \in \widehat{E}$, if there exists one; $\bot$ otherwise.

1: **if** u is irrelevant for Δ w.r.t. $\mathscr{s}$ (according to Theorems 2, 5-7) **then**
2: **return** $\widehat{E}$ obtained from E by using Theorems 3 and 4;
3: Let $\Delta_{AF} = \langle \mathsf{A}, \mathsf{R} \rangle$ be the AF for Δ;
4: $E_{AF} = \mathbf{HOAFNtoAF}_{\Delta}(E)$;
5: **if** $u = +C$ **then**
6: $\Delta_{AF} = \langle \mathsf{A} \cup \mathbf{AdditionalArgs}(u, \Delta), \mathsf{R} \rangle$;
7: $E_{AF} = E_{AF} \cup \mathbf{AdditionalArgs}(u, \Delta)$;
8: $u_{AF} = \mathbf{UpdatesOverAF}(u, \Delta)$;
9: $\widehat{E}_{AF} =$ INCR-AF-CA($\Delta_{AF}, u_{AF}, \varrho, \mathscr{s}, E_{AF}$, AFS);
10: **if** $\widehat{E}_{AF} \neq \bot$ **then**
11: **return** $\widehat{E} = \mathbf{AFtoHOAFN}_{u(\Delta)}(\widehat{E}_{AF})$;
12: **return** $\bot$;

The following theorem states the soundness of Algorithm 4.

Theorem 8. *For any HOAFN $\Delta = \langle \mathbb{A}, \mathbb{R}, \mathbb{T} \rangle$, $\mathscr{s}$-extension $E \in \mathcal{E}_{\mathscr{s}}(\Delta)$ with $\mathscr{s} \in \{\mathit{co}, \mathit{pr}, \mathit{st}, \mathit{gr}\}$, update u, and goal $\varrho \in \mathbb{A} \cup \mathbb{R} \cup \mathbb{T}$, Algorithm 4 computes $\widehat{E} \in \mathcal{E}_{\mathscr{s}}(u(\Delta))$ with $\varrho \in \widehat{E}$ if ϱ is credulously accepted in $u(\Delta)$ w.r.t. semantics $\mathscr{s}$, otherwise it returns $\bot$.*

An example of usage of Algorithm 4 is given below.

Example 21. Consider the RAFN Δ shown in Fig. 1 (left), and the goal argument d, which is credulously accepted under the complete semantics as there exists extension $E = \{\mathtt{st}, \mathtt{w}, \mathtt{d}, \mathtt{p}, \alpha_0, \alpha_1, \alpha_2, \alpha_4, \beta_0, \beta_1\}$ containing d.

Suppose Algorithm 4 is called for recomputing the credulous acceptance of d w.r.t. complete semantics after performing an (acyclic, positive) update $u = +\{\mathtt{ns}, \alpha_5\}$ with ns being a new argument and $\mathbf{b}(\alpha_5) = (\mathtt{ns}, \alpha_3)$. The update u is identified as irrelevant at Line 1 by using the result of Theorem 7 (as $\mathbf{v}'(\mathbf{s}(\alpha_5)) = \mathbf{v}'(\mathtt{ns}) = \mathtt{true}$ and $\mathbf{v}'(\mathbf{t}(\alpha_5)) = \mathbf{v}'(\alpha_3) = \mathtt{false}$), and thus $\widehat{E} = E \cup \{\mathtt{ns}, \alpha_5\}$ is a complete extension of the updated RAFN $u(\Delta)$ (Line 2) and d is still credulously accepted.

Assume now that $\mathbf{b}(\alpha_5) = (\mathtt{ns}, \beta_0)$, as in the case of Fig. 1 (right). In this case u is not identified as irrelevant (as $\mathbf{v}'(\beta_0) = \mathtt{true}$). Thus, AF Δ_{AF} computed at Line 3 is obtained from that in Fig. 13 without considering the blue arguments and the dashed attacks, and $E_{AF} = E \cup \{\alpha_0^{\vee}, \alpha_1^{\vee}, \mathtt{rn}^-, \alpha_2^{\vee}, \mathtt{pt}^-, \alpha_3^-, \alpha_4^{\vee}\}$. As the update is positive, at Lines 6-7 the set of arguments $\mathbf{AdditionalArgs}(u, \Delta) = \{\alpha_5, \alpha_5^-, \alpha_5^{\vee}, \mathtt{ns}, \mathtt{ns}^-\}$, corresponding to blue nodes in Fig. 13, is added to both Δ_{AF} and E_{AF}. The updates u_{AF} for Δ_{AF} are the dashed attacks in Fig. 13. Then, INCR-AF-CA computes the influenced set $I = \mathbf{inf}(u_{AF}, \Delta_{AF}, E_{AF}) = \{\beta_0, \beta_0^-, \alpha_5^-, \mathtt{ns}^-, \alpha_5^{\vee}\}$ (*i.e.*, the arguments whose name is colored in red in Fig. 13), and the reduced AF $\mathbf{red}(u_{AF}, \Delta_{AF}, E_{AF})$ that consists of the arguments and attacks of Δ_{AF} that are colored in red or blue in Fig. 13.

Observe that, whereas arguments ns and α_5 (which attack some argument in the influenced set) are part of the reduced AF, argument $\beta_1^{\vee}$ (which also attacks an argument in the influenced set) is not part of the reduced AF because its valuation is $\mathtt{false}$. Since the goal argument d is not influenced by the update (as it is not part of the influenced set), it will continue to be credulously accepted in $u(\Delta)$ under co. In fact, a co-extension $E_d = \{\mathtt{ns}, \alpha_5, \alpha_5^{\vee}, \beta_0^-\}$ of the reduced AF is computed by calling function $\mathrm{AFS}^{\mathrm{SE}}_{\mathit{co}}$ and by merging its result with the part of the initial extension E_{AF} that is not influenced. Consequently, $\widehat{E}_{AF} = E_{AF} \setminus I \cup E_d$ is a complete extension of $u_{AF}(\Delta_{AF})$ that contains d, and thus the co-extension $\mathbf{AFtoHOAFN}_{u(\Delta)}(\widehat{E}_{AF}) = \widehat{E}_{AF} \cap (\mathbb{A} \cup \mathbb{R} \cup \mathbb{T})$ of the updated RAFN $u(\Delta)$ is returned at Line 11. □

Although the worst-case time complexity of Algorithm 4 is the same as that of recomputing an extension of the updated HOAFN from scratch, in practice the improvement derived from the incremental computation is relevant, as discussed in the following section.

6. Experimental evaluation

In this section we first present a parametric dataset generator for HOAFNs and then discuss the empirical analysis carried out, showing that our incremental approach generally outperforms the recomputation of credulous acceptance from scratch.

6.1. Dataset generator

There are several benchmark generators and solvers for Dung's AFs [69]; however, to the best of our knowledge, no dataset is available for HOAFNs. To fill this gap, we propose a dataset generator that transforms any AF into a HOAFN, with the aim of preserving as much as possible the structure of the initial AF.

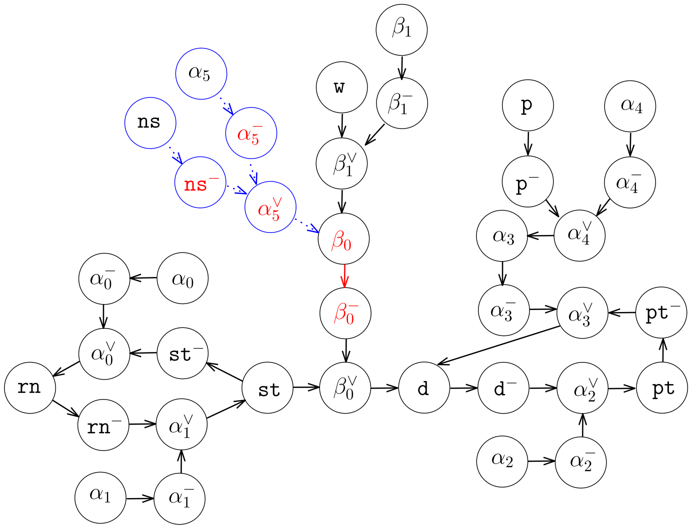


Fig. 13. AF for the RAFN $u(\Delta)$ of Example 21 with $u = +\{\mathtt{ns}, \alpha_5\}$ and $\mathbf{b}(\alpha_5) = (\mathtt{ns}, \beta_0)$. Influenced arguments are colored in red. Blue nodes correspond to the additional arguments returned by function **AdditionalArgs**(u, Δ). Dashed attacks correspond to the updates to be performed over the AF Δ_{AF} (*i.e.*, u_{AF}) to obtain the AF $u_{AF}(\Delta_{AF})$ for the updated RAFN $u(\Delta)$. (For interpretation of the colors in the figure(s), the reader is referred to the web version of this article.)

Our HOAFN Dataset Generator (Algorithm 5) receives as input an AF $\langle \mathsf{A}, \mathsf{R} \rangle$ and five rational numbers $P_{\mathbb{T}}$, $P_{\mathbb{R}}$, $P_{\mathbb{RR}}$, $P_{\mathbb{RT}}$, $P_{\mathbb{TR}}$, $P_{\mathbb{TT}}$ in the interval $[0, 1]$ that will be used for determining the number of attacks, supports, attacks to attacks, attacks to supports, supports to attacks and supports to supports in a HOAFN, respectively. The output is the HOAFN $\Delta = \langle \mathbb{A} = \mathsf{A} \cup \{x, y\}, \mathbb{R}, \mathbb{T} \cup \{\beta_{xy}\} \rangle$ having a number of supports $|\mathbb{T}| = \lfloor P_{\mathbb{T}} \cdot |\mathsf{R}| \rfloor$ and a number of attacks $|\mathbb{R}| = |\mathsf{R}| - |\mathbb{T}|$. Moreover, among the attacks $\mathbb{R}$ (resp. supports $\mathbb{T}$), the target of $\lfloor P_{\mathbb{RR}} \cdot |\mathsf{R}| \rfloor$ and $\lfloor P_{\mathbb{RT}} \cdot |\mathsf{R}| \rfloor$ attacks (resp. $\lfloor P_{\mathbb{TR}} \cdot |\mathsf{R}| \rfloor$ and $\lfloor P_{\mathbb{TT}} \cdot |\mathsf{R}| \rfloor$ supports) will be modified to be an attack or a support, respectively. The (meta) support β_{xy} is added to guarantee that the set of supports having a support as target is acyclic. For instance, consider an AF having 100 attacks (*i.e.*, $|\mathsf{R}| = 100$), and the values $P_{\mathbb{T}} = 0.45$, $P_{\mathbb{TR}} = 0.15$ and $P_{\mathbb{TT}} = 0.10$. The derived HOAFN will have 55 attacks and 45 supports. Among the 45 supports, the target of 15 (resp. 10, 20) of them is an attack (resp. support, argument).

Algorithm 5 HOAFN dataset generator.

Input: AF $\langle \mathsf{A}, \mathsf{R} \rangle$, fractions $P_{\mathbb{T}}$, $P_{\mathbb{R}} = 1 - P_{\mathbb{T}}$, $P_{\mathbb{RR}}$, $P_{\mathbb{RT}}$, $P_{\mathbb{TR}}$, $P_{\mathbb{TT}}$.
Output: HOAFN $\Delta = \langle \mathbb{A}, \mathbb{R}, \mathbb{T} \rangle$.
1: **if** $P_{\mathbb{RR}} + P_{\mathbb{RT}} > P_{\mathbb{R}}$ or $P_{\mathbb{TR}} + P_{\mathbb{TT}} > P_{\mathbb{T}}$ **then**
2: **return** $\langle \emptyset, \emptyset, \emptyset \rangle$;
3: $S =$ **randAcyclicSet**$(\mathsf{R}, \lfloor P_{\mathbb{T}} \cdot |\mathsf{R}| \rfloor)$;
4: $\mathbb{R} = \{\alpha_{ab} \mid (a, b) \in \mathsf{R} \setminus S\}$;
5: $\mathbb{T} = \{\beta_{ab} \mid (a, b) \in S\}$;
6: Let x, y be two new arguments and β_{xy} a new support;
7: $\Delta = \langle \mathbb{A} = \mathsf{A} \cup \{x, y\}, \mathbb{R}, \mathbb{T} \cup \{\beta_{xy}\} \rangle$;
8: $(S_{\mathbb{RR}}, S_{\mathbb{RT}}) =$ **randDisjointSets**$(\mathbb{R}, \lfloor P_{\mathbb{RR}} \cdot |\mathsf{R}| \rfloor, \lfloor P_{\mathbb{RT}} \cdot |\mathsf{R}| \rfloor)$;
9: $(S_{\mathbb{TR}}, S_{\mathbb{TT}}) =$ **randDisjointSets**$(\mathbb{T}, \lfloor P_{\mathbb{TR}} \cdot |\mathsf{R}| \rfloor, \lfloor P_{\mathbb{TT}} \cdot |\mathsf{R}| \rfloor)$;
10: **for** $\gamma \in S_{\mathbb{RR}} \cup S_{\mathbb{TR}}$ **do**
11: $\mathbf{t}(\gamma) =$ **randElement**$(\mathbb{R})$;
12: **for** $\alpha \in S_{\mathbb{RT}}$ **do**
13: $\mathbf{t}(\gamma) =$ **randElement**$(\mathbb{T})$;
14: **for** $\beta \in S_{\mathbb{TT}}$ **do**
15: $\mathbf{t}(\beta) =$ **randElement**$(\mathbb{T}, \beta_{xy})$;
16: **return** Δ;

Algorithm 5 works as follows. It starts by checking if the input parameters are well-formed (Line 1); if the well-formedness condition is not satisfied, then an empty HOAFN is returned (Line 2). Then, it selects a subset S of attacks in R which will be rewritten as supports (Line 3). The initial set of HOAFN attacks $\mathbb{R}$ (resp. supports $\mathbb{T}$) is computed at Line 4 (resp. Line 5), by assigning a unique label α_{ab} (resp. β_{ab}) to every pair (a, b) in $\mathsf{R} \setminus S$ (resp. S). Next the (meta) arguments x and y and the (meta) support β_{xy}, with $\mathbf{s}(\beta_{xy}) = x$ and $\mathbf{t}(\beta_{xy}) = y$, are introduced in the HOAFN denoted by Δ (Line 7).

After that, at Line 8, two disjoint subsets $S_{\mathbb{RR}}$ and $S_{\mathbb{RT}}$ of $\mathbb{R}$ are randomly selected. Similarly, at Line 9, two disjoint subsets $S_{\mathbb{TR}}$ and $S_{\mathbb{TT}}$ of $\mathbb{T}$ are determined. The target of the attacks and supports in $S_{\mathbb{RR}}$ and $S_{\mathbb{TR}}$ is updated to be an attack in Lines 10-11, by setting it to an attack randomly taken from $\mathbb{R}$ by means of the auxiliary function **randElement**. Analogously, the target of attacks in $S_{\mathbb{RT}}$ (resp. supports in $S_{\mathbb{TT}}$) is updated in Lines 12-13 (resp. 14-15) by setting it to be an attack (resp. support) randomly selected from $\mathbb{T}$ by means of function **randElement**. It is worth noting that Line 15 uses a different **randElement** function (with two parameters). This function assigns the support β_{xy} as the target of β whenever the assignment of any element in $\mathbb{T}$ introduces a cycle in $S_{\mathbb{T}\mathbb{T}}$.

Regarding function **randAcyclicSet**, it can be easily implemented by using any algorithm computing an approximate solution for the *Maximum Acyclic Subgraph (MAS)* problem, such as those proposed in [70,71], here referred to as AMAS algorithm. To this end, let S' be the set of attacks of the subgraph computed by the AMAS algorithm, if $|S'| > \lfloor P_{\mathbb{T}\mathbb{T}} \cdot |\mathsf{R}| \rfloor$, a number $|S'| - \lfloor P_{\mathbb{T}\mathbb{T}} \cdot |\mathsf{R}| \rfloor$ of edges are randomly selected and removed from S'. Moreover, if $|S'| < \lfloor P_{\mathbb{T}\mathbb{T}} \cdot |\mathsf{R}| \rfloor$, $\lfloor P_{\mathbb{T}\mathbb{T}} \cdot |\mathsf{R}| \rfloor - |S'|$ attacks are randomly taken from $\mathsf{R} \setminus S'$ and, after setting their target to be the (meta) argument x, are added to S'. It is worth noting that, let E be the set of edges of a directed graph, the AMAS algorithm ensures that the number of edges of the returned subgraph is at least $|E|/2$ and very often it is much larger than $|E|/2$. This means that the meta argument x is rarely used for setting the target of a support, and for $|S'| \leq |E|/2$ it is never used.

Before concluding, it is important to observe that the dataset generation process aims at preserving the number of AF's arguments and relationships, as well as the out-degree (*i.e.* the number of outcoming interactions) for every argument of the input AF. As for the complexity of Algorithm 5, each step is polynomial w.r.t. $|\mathsf{R}|$.

6.2. Empirical evaluation

We implemented a C++ prototype and, for each semantics $\mathcal{s}$, compared the performance of Algorithm 4 against the computation from scratch (Algorithm 2). For the computation from scratch, as well as in Algorithm 4, we have used the solvers that won the fourth edition of the ICCMA competition for the tasks DC-$\mathcal{s}$ (with $\mathcal{s} \in \{\mathtt{co}, \mathtt{pr}, \mathtt{st}\}$)[13] of computing the credulous acceptance of a goal element w.r.t. a given AF under semantics $\mathcal{s}$. That is, for any task DC-$\mathcal{s}$ (with $\mathcal{s} \in \{\mathtt{co}, \mathtt{pr}, \mathtt{st}\}$), we used *$\mu$-toksia* [72,46], A-FOLIO-DPDB [47] (with the same version of μ-toksia, that is, version 2021), and FUDGE [48]. As a matter of fact, A-FOLIO-DPDB [47] is equivalent to calling μ-toksia for all the tasks considered in this paper, thus we avoid to report the results for A-FOLIO-DPDB in what follows.

6.2.1. Dataset

The HOAFN dataset is obtained from the AF's datasets used as benchmarks in ICCMA'17, ICCMA'19 and ICCMA'21. As it will be cleared in the following, since our technique is sensible w.r.t. the percentage of arguments whose status potentially needs to be recomputed after an update, we have considered only AFs having such percentage lower than 80%. This percentage corresponds to the average reachability in the argumentation graph, which is defined as follows. For each AF, for each argument a, compute the number of arguments that are reachable from a; this gives the percentage of arguments that are reachable from a in the considered AF. Then, we keep an AF from the benchmarks dataset if the average percentage of arguments that are reachable from any of its arguments is less than 80%. Each considered AF $\langle \mathsf{A}, \mathsf{R} \rangle$ was processed through Algorithm 5 to obtain a HOAFN. The percentages used as input in Algorithm 5 are as follows:

- $P_{\mathbb{R}} = P_{\mathbb{T}} = 0.5$, prescribing that 50% of the attacks in R are transformed into first-level attacks of the resulting HOAFN, while the 50% of the (remaining) attacks in R are transformed into first-level supports;
- $P_{\mathbb{R}\mathbb{R}} = P_{\mathbb{R}\mathbb{T}} = P_{\mathbb{T}\mathbb{R}} = P_{\mathbb{T}\mathbb{T}} = 0.15$, prescribing that 60% of attacks in R are transformed into attacks or supports targeting an attack or a support (*i.e.*, 15% for each of the four kinds of relations); 40% of attacks in R are transformed into attacks (20%) or supports (20%) whose target is an argument.

The choice of these parameters was guided by a recent work building Bipolar AF benchmarks in real contexts by using argument-mining techniques [73,74], where the percentage of support relations obtained is on average 50.6%. That is, approximately half of the (first-level) interactions are supports and the remaining ones are attacks. Therefore, in line with results of that work, in our setting we set the number of attack and support relations equal to 50% in each HOAFN.

Starting from the above-mentioned AFs, we obtained the HOAFN dataset consisting of 541 HOAFNs with a number of arguments $|\mathbb{A}| \in [5, 50\text{K}]$ and a number of interactions $|\mathbb{R} \cup \mathbb{T}| \in [5, 214\text{K}]$.

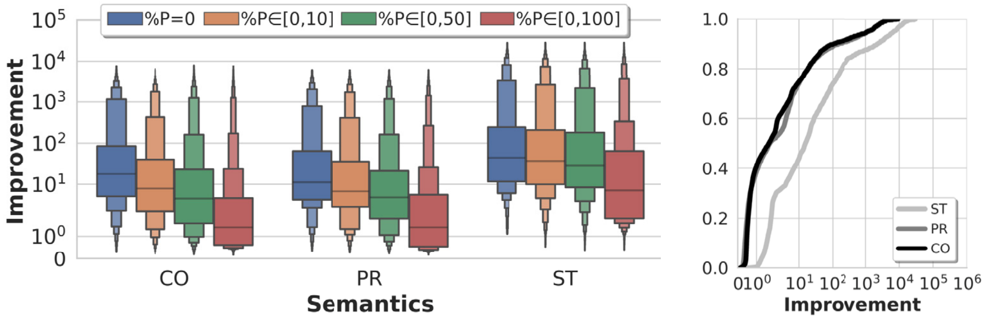
The set of the AFs for the ASAFs (resp., RAFNs), obtained by using Definitions 16 and 17, consists of 541 AFs $\langle \mathsf{A}, \mathsf{R} \rangle$ with $|\mathsf{A}| \in [23, 857\text{K}]$ (resp., $|\mathsf{A}| \in [29, 2\text{M}]$) and $|\mathsf{R}| \in [14, 1.5\text{M}]$ (resp., $|\mathsf{R}| \in [18, 3\text{M}]$). We observed that the size of AFs for HOAFNs is up to 50% smaller than what would be obtained without applying the optimization of the mappings given in Definitions 16 and 17.

6.2.2. Methodology

For each semantics $\mathcal{s}$ and HOAFN Δ in the dataset, we consider *i*) an initial $\mathcal{s}$-extension E of Δ,[14] *ii*) a randomly selected goal element $\varrho \in E$, and *iii*) an update u selected among one of the possible two types (deletion of an element or addition of a set of elements). Next, we compute an $\mathcal{s}$-extension E' of the updated HOAFN $u(\Delta)$ possibly containing the goal element by calling our incremental algorithm DYN-HOAFN-CA (Algorithm 4). Finally, we compute the *improvement* of DYN-HOAFN-CA over the computation from scratch (*i.e.*, Algorithm 2) as t_s/t_{A_4}, where t_s is the time needed by the computation from scratch and t_{A_4} is the time needed by Algorithm 4. Thus, the improvement tells us how many times Algorithm 4 is faster than the computation from scratch.

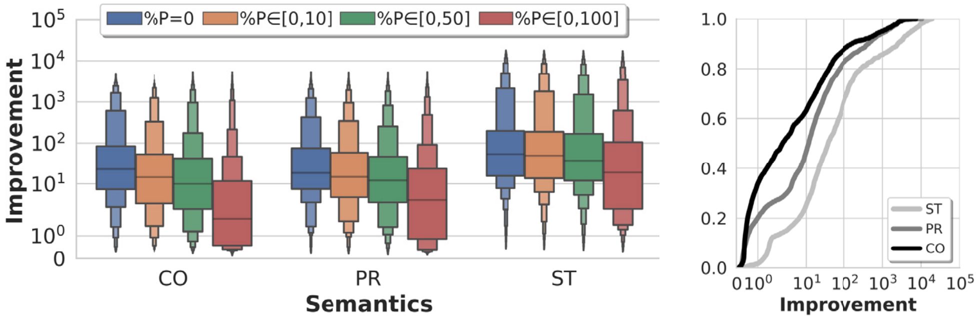
[13] As the grounded semantics was completely excluded from the competition because it is not considered as challenging enough, we avoid considering DC-$\mathcal{gr}$.

[14] To avoid timeout problems, we select the first extension returned by standard solvers.



(a) (b)

Fig. 14. Letter-value plots (a) and cumulative distribution (b) of improvements for the RAFN dataset.



(a) (b)

Fig. 15. Letter-value plots (a) and cumulative distribution (b) of improvements for the ASAF dataset.

The experiments have been carried out on an Intel Core i7-8700B CPU 3.2 GHz, 16 GB RAM running Ubuntu.

Notice that the incremental task considered in this paper is different from the dynamic task of ICCMA 2021, which is as follows: given an AF Δ_0, a goal argument g, and a sequence of updates $U = (u_1, ..., u_n)$, compute the credulous acceptance of g w.r.t. $\Delta_i = u_i(\Delta_{i-1})$ with $i \in [1..n]$. The main difference between our incremental approach and the dynamic ICCMA 2021 problem is that, for each upcoming update, we compute the credulous acceptance of a goal element (as well as an $\mathscr{s}$-extension of the updated HOAFN containing the goal if it exists) *given an extension* of the initial HOAFN containing the goal, whereas in the dynamic ICCMA 2021 problem there is a *fixed* sequence of updates and the credulous acceptance is computed after each update in the sequence (herein all updates are known in advance). Furthermore, our incremental approach also differs from the problem addressed by the ICCMA 2023 dynamic track, which does not require the sequence of updates to be known beforehand but still does not include an initial extension as input at the beginning of the computation. Notwithstanding this, interestingly, the task considered in this paper may be relevant even for solving the dynamic tasks considered at ICCMA from the second update on.

6.2.3. Results

For each HOAFN we performed 20 runs, each with a different update (10 whose root is an attack or a support, and 10 whose root is an argument). For the sake of readability, we only report the results obtained by calling μ-toksia [46]. The results for FUDGE [48] are not reported here as the average improvement for each data point slightly differs ($< 1\%$) from that obtained w.r.t. μ-toksia.

By performing a correlation test, we observed that the improvement is inversely correlated (with Pearson, Kendall, and Spearman correlation values equal to -0.37, -0.63, -0.83, respectively) to the percentage of the elements involved in the incremental computation (hereafter denoted as P)—the lower the percentage P, the higher the improvement.

For this reason, we have grouped all the data points into four clusters in Fig. 14(a) and Fig. 15(a):

- For each semantics, a blue cluster groups all the data points where updates are irrelevant (that is, the percentage of elements whose status needs to be recomputed is $P = 0\%$);
- For each semantics, an orange (resp., green and red) cluster groups all the data points with $P \leq 10\%$ (resp. $P \leq 50\%$ and $P \leq 100\%$).

Fig. 14(a) and Fig. 15(a) report Letter-Value Plots (LVPs) [75], that is a variation of boxplots that replaces the whiskers with a variable number of letter values (the width of a box is proportional to the number of its points; a line representing the median is also shown in the central boxes).

RAFN dataset Fig. 14(a) shows the improvement for RAFN. The average improvement (that is, the mean of the improvements in the red-colored LVPs, which contains all data points) is 113 (resp., 120, and 747) under complete (resp., preferred, and stable) semantics. Considering all data points concerning all semantics, the average improvement is 166, meaning that the incremental computation takes on average 0.6% of the time needed by the computation from scratch.

Moreover, the median value of the improvement (*i.e.*, the median value of the improvements in the red-colored LVPs) is 1.34 (resp., 1.35, and 7.44) under complete (resp., preferred, and stable) semantics.

When focusing on updates whose root is (resp., is not) an argument, the average improvement is 4.3, 4.7, and 9.1 (resp., 162, 171, and 919) under complete, preferred and stable semantics, while the median value of the improvement is 1.1, 1.4, and 1.8 (resp., 1.6, 1.7, and 15). Considering all data points concerning all semantics, the average improvement is 5.4 (resp., 233), and thus the incremental computation takes on average 18.5% (resp., 0.43%) of the time needed by the computation from scratch. This can be explained by looking at the number of elements in the interaction set, which in the case of updates whose root is an argument is on average 64.6 times that of updates whose root is *not* an argument.

The percentage of updates that are irrelevant (*i.e.*, data points that are in the blue LVP over those in the red one) is 22% (resp., 27% and 52%) under complete (resp., preferred and stable) semantics.

Finally, Fig. 14(b) reports the cumulative distribution of the improvement under complete (resp., preferred and stable) semantics, that is, the percentages of runs whose improvement is less than or equal to a given value. It turns out that the improvement is less than 1 for 28.5% of the data points on average. However, in these cases the average improvement is 0.62, while for the remaining part (71.5% of the runs) it is 385.6, meaning that most of the times Algorithm 4 is much faster than the computation from scratch.

ASAF dataset Fig. 15(a) shows the improvement for ASAF. The results for ASAF are similar to those for RAFN. In particular, the average improvement is 101 (resp., 127, and 662) under complete (resp., preferred and stable) semantics. Overall, considering all data points concerning all semantics, we found that the average improvement is 156, and thus the incremental computation takes on average 0.64% of the time needed by the computation from scratch. Moreover, the median value of the improvement is 1.7 (resp., 4.3, and 20.1) under complete (resp., preferred and stable) semantics.

When focusing on updates whose root is (resp., is not) an argument, the average improvement is 6.3, 6.6, and 14.0 (resp., 143, 179, and 825) under complete, preferred and stable semantics, while the median value of the improvement is 1.2, 1.1, and 1.6 (resp., 2, 11, and 36). Considering all data points concerning all semantics, the average improvement is 7.9 (resp., 217), meaning that the incremental computation takes on average 12.7% (resp., 0.46%) of the time needed by the computation from scratch. Therefore the average improvement of the case of update whose root is *not* an argument is 27.5 times that of the case of update whose root is an argument. Again, this can be ascribed to the cardinality of the interaction set, which in the case of update whose root is an argument is 68 times that of the case of update whose root is *not* an argument, on average.

As for the percentage of updates that are irrelevant, it is 28% (resp., 46% and 65%) under complete (resp., preferred and stable) semantics.

Finally, Fig. 15(b) reports the cumulative distribution of the improvement under complete (resp., preferred and stable) semantics, The improvement is less than 1 for 22.2% of the data points on average. However, in these cases the average improvement is 0.55, while for the remaining cases (77.8% of the total) it is 338. As observed earlier, although there are cases in which the computation from scratch is slightly faster, most of the times the incremental approach is extremely faster than the computation from scratch.

7. Related work

Recent years have witnessed intensive formal study, development, and application of Dung's framework in various directions [5]. Several proposals have been made to extend the Dung's framework with the aim of better modeling the knowledge to be represented. These extensions include Bipolar AF [29,51], AF with recursive attacks and supports [59,27,28], AF with constraints [76–78], and AF with preferences [79,80,66]. Interestingly, the HOAFN framework can capture forms of preferences, other than representing higher-order interactions. In particular, two main approaches have been proposed in the literature for Preference-based Argumentation Frameworks (PAFs). The first approach considers preferences and attacks separately as they describe different pieces of knowledge [81,82]. However, this approach suffers from an increase in the computational complexity, which is essentially due to the fact that it is possible to express preferences on top of extensions of the underlying AF [66]. The second approach defines the PAF semantics in terms of an auxiliary AF that can be obtained by modifying the underlying AF (*e.g.*, by deleting or inverting attacks "contradicting" preferences) and can be built in polynomial time, thus guaranteeing that the complexity does not increase w.r.t. that of AF [83,81,82]. The advantage of using the HOAFN framework is that it is able to model the latter kind of preferences as well as other higher-order interactions (*e.g.*, preferences that, in addition to arguments, may also involve attacks and supports) without resulting in an increase of the computational complexity. The fact that the complexity remains the same as that of AF follows from the translation from a HOAFN to an AF provided in Section 3.

Regarding the use of high-order interactions for modeling preferences, we note the work of [80] in which second-order attacks are used to model preferences, resulting in attacks from an argument to a first-order attack. Similarly to our approach, such arguments encoding the preferences can also be attacked. Furthermore, in a HOAFN such arguments could also be supported, and the corresponding attacks and supports targeting them, as well as the high-order attacks encoding preferences can themselves be attacked or supported. This latter characteristic is partly shared by the AFRA [20], in which high-order attacks can be used to encode preferences. This feature, shared by the AFRA and the HOAFN, cannot be directly captured by PAF.

There have been significant efforts aimed at coping with dynamic aspects of Dung's abstract argumentation frameworks [84,85]. On the one hand, [86,87] have investigated the principles according to which the grounded extension of an AF does not change when the set of arguments/attacks is changed. In [35] a synthesis is presented concerning the characterization of changes based on the work presented in [88–91], where the evolution of the set of extensions after performing a change operation is studied; there, a change operation can be about adding or removing one interaction, or adding or removing one argument and a set of interactions.

Dynamic argumentation has been applied to decision-making of an autonomous agent in [92], where the authors study how the acceptability of arguments evolves when a new argument is added to the decision system. The division-based method, proposed by [38] and then refined in [33], divides the updated framework into two parts: *affected* and *unaffected*, where only the status of affected arguments is recomputed after updates. [93] investigated the efficient evaluation of the justification status of a subset of arguments in an AF (instead of the whole set of arguments), and proposed an approach based on answer-set programming for *local* computation. In [94], an AF is decomposed into a set of strongly connected components, yielding sub-AFs located in layers, which are then used for incrementally computing the semantics of the given AF by proceeding layer by layer. [95] introduced a matrix representation of argumentation frameworks and proposed a matrix reduction that, when applied to dynamic AFs, resembles the division-based method of [38].

Other relevant works on dynamic aspects of Dung's AFs include the following. [34] proposed an approach exploiting the concept of splitting of logic programs to deal with dynamic argumentation. The technique considers *weak expansions* of the initial AF, where added arguments never attack previous ones. [96] has investigated whether and how it is possible to modify a given AF so that a desired set of arguments becomes an extension, whereas [97] studied equivalence between two AFs when further information (another AF) is added to both AFs. [98] has focused on expansions where new arguments and attacks may be added but the attacks among the old arguments remain unchanged, while [99] characterized update and deletion equivalence, where adding/deleting arguments/attacks is allowed (deletions were not considered by [97,98]). The line of work by Baumann et al. was further pursued, more recently in [100], where the authors introduced a parametrized equivalence notion for AFs that subsumes the pre-existing notions of standard and strong equivalence as corner cases.

Bipolarity in argumentation is discussed in [101], where an overview of the use of bipolarity and a formal definition of a Bipolar Argumentation Framework (BAF) that extends Dung's AF by including supports is provided. A survey of different approaches to support in argumentation can be found in [102]. [103] considers a general support relation among arguments as a positive interaction, without giving additional constraints. [104] introduces the concept of evidential support which enables to distinguish between *prima facie* and standard arguments. Prima facie arguments represent the notion of evidence and do not require support from other arguments to stand, while standard arguments cannot be accepted unless they are supported by evidence. [105] provides a deductive interpretation of support. Deductive support is intended to capture the following intuition: if argument a supports argument b then the acceptance of a implies the acceptance of b and, as a consequence, the non-acceptance of b implies the non-acceptance of a. Necessary support relations have been introduced in [106,29,107] to enforce the following constraint: if argument a supports argument b it means that a is necessary for b. Thus, the acceptance of b implies the acceptance of a and, conversely, the non-acceptance of a implies the non-acceptance of b.

In Bipolar AFs, necessary support and deductive support are shown to be dual (*i.e.* we can transform a BAF with necessity into an equivalent BAF with deductive supports by simply reversing the direction of the support arrows) [26]. However, in the case of HOAFNs that are not BAFs, this duality no longer holds. This is because, in line with other approaches in the literature, the HOAFN requires the source of any higher-order support to be an argument (respectively, a set of arguments) [26]. Hence, if one were to reverse the direction of a higher-order support link in the HOAFN, it would result in a support having an attack or a support as its source which is, in fact, syntactically forbidden in the framework.

Changes in bipolar argumentation frameworks (BAFs) have been studied in [108], where it is shown how the addition of one argument together with one support involving it (and without any attack) impacts on the extensions of the updated BAF. The problem of incrementally computing extensions of dynamic BAFs, with a deductive interpretation of supports [19], has been first addressed in [109], and then extended in [44] to deal with second-order attacks [110]. The problem of efficiently and incrementally computing extensions for dynamic ASAFs has been investigated in [111]. Differently from that work, we considered RAFNs in addition to ASAFs, formally characterized updates and identified irrelevant updates for HOAFNs in general, and provided an incremental algorithm in a wider setting where the focus is on credulous acceptance (a HOAFN extension of the updated framework is returned as byproduct). Also, the characterizations of HOAFNs in terms of AFs are new for both the ASAF and the RAFN. Our incremental algorithm deals with the more general case of a HOAFN, which can either be an ASAF or a RAFN, as well as the new mappings into an AF. Finally, we defined the dataset generator algorithm and provided an experimental analysis for the credulous acceptance problem in the dynamic setting.

In relation to the translation schemes presented in Section 3, several works have been made sharing the same underlying idea. In [105] auxiliary arguments are used to encode a BAF into a standard AF. In [112], a flattening function converting Extended Explanatory Argumentation Frameworks into AFs was shown, providing equivalence in terms of complete extensions. However, none of the existing approaches deals with the incremental computation and the empirical evaluation of a general framework like the HOAFN, which generalizes ASAFs and RAFNs.

8. Conclusion and future work

In this paper we jointly tackled two relevant aspects that were so far considered separately by the community of argumentation: extending Dung's abstract argumentation framework with recursive attacks and supports, and considering dynamic aspects of computational argumentation. In order to do this, we started by proposing a general framework that encompasses the ASAF [27] and a restricted version of the RAFN [28], that we called *High-Order Argumentation Framework with Necessities (HOAFN)*. Based on this general framework, we provided alternative characterizations for the acceptability semantics of the ASAF and the RAFN, which were useful for developing our algorithmic solutions.

Taking the HOAFN as a starting point, we formally showed that some updates that may be performed over an ASAF or a RAFN are irrelevant, enabling to obtain an extension of the updated framework containing a given goal element with low computational effort. In particular, we proposed an algorithm that efficiently and incrementally computes the credulous acceptance of a given element (as well as an extension containing it, as byproduct) of an updated HOAFN. We empirically showed that our approach significantly outperforms the winners of the fourth ICCMA competition for the considered semantics in this dynamic setting.

To the best of our knowledge, this is the first paper addressing the issue of recomputing credulous acceptance in HOAFNs or other frameworks with high-order interactions in dynamic scenarios. Moreover, given the generality of the HOAFN, our technique can also be applied to restricted frameworks such as AFRAs [20], RAFs [56] and AFNs [29].

As part of future work, we plan to extend our technique to deal with other computational problems, such as enumerating extensions and deciding skeptical acceptance for dynamic HOAFNs. In addition, we will work on formalizing labeling-based semantics for the HOAFN, in particular, considering its instantiation with an ASAF or a RAFN. Also, related to the discussion of the duality between necessary support and deductive support, we are interested in studying this matter in the context of frameworks with higher-order interactions like the HOAFN. To this end, we will explore the possibility of having higher-order interactions whose source is an attack or a support in the context of a HOAFN, specially for the case of supports, and study how such support links should be interpreted. Finally, as additional future work, we plan to carry out a principle-based analysis of the semantics for HOAFNs and work on the development of a direct solver for HOAFNs, under RAFN and ASAF semantics.

CRediT authorship contribution statement

Gianvincenzo Alfano: Conceptualization, Data curation, Formal analysis, Funding acquisition, Investigation, Methodology, Project administration, Resources, Software, Supervision, Validation, Visualization, Writing – original draft, Writing – review & editing. **Andrea Cohen:** Conceptualization, Data curation, Formal analysis, Funding acquisition, Investigation, Methodology, Project administration, Resources, Software, Supervision, Validation, Visualization, Writing – original draft, Writing – review & editing. **Sebastian Gottifredi:** Conceptualization, Data curation, Formal analysis, Funding acquisition, Investigation, Methodology, Project administration, Resources, Software, Supervision, Validation, Visualization, Writing – original draft, Writing – review & editing. **Sergio Greco:** Conceptualization, Data curation, Formal analysis, Funding acquisition, Investigation, Methodology, Project administration, Resources, Software, Supervision, Validation, Visualization, Writing – original draft, Writing – review & editing. **Francesco Parisi:** Conceptualization, Data curation, Formal analysis, Funding acquisition, Investigation, Methodology, Project administration, Resources, Software, Supervision, Validation, Visualization, Writing – original draft, Writing – review & editing. **Guillermo R. Simari:** Conceptualization, Data curation, Formal analysis, Funding acquisition, Investigation, Methodology, Project administration, Resources, Software, Supervision, Validation, Visualization, Writing – original draft, Writing – review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

Acknowledgements

The authors wish to thank the anonymous referees for providing detailed comments and suggestions that helped to substantially improve the paper. This research was partially supported by PGI-UNS under grants 24/ZN37 and 24/N055. Moreover, the authors acknowledge the support of the PNRR project FAIR - Future AI Research (PE00000013), Spoke 9 - Green-aware AI, under the NRRP MUR program funded by the NextGenerationEU. Finally, this work was also funded by the Next Generation EU - Italian NRRP, Mission 4, Component 2, Investment 1.5, call for the creation and strengthening of 'Innovation Ecosystems', building 'Territorial R&D Leaders' (Directorial Decree n. 2021/3277) - project Tech4You - Technologies for climate change adaptation and quality of life improvement, n. ECS0000009. This work reflects only the authors' views and opinions, neither the Ministry for University and Research nor the European Commission can be considered responsible for them.

Appendix A. Summary of the main notations

Below we report a table containing the main notations used in the paper.

Table A.3
Summary of the main notations used in the paper, possibly with subscripts or superscripts. The last column reports the definitions where the symbols are introduced or the places of their first occurrences, if defined inline.

	Symbols	Meaning	Occurrence
Frameworks	$\langle \mathsf{A}, \mathsf{R} \rangle$ (or simply Λ)	abstract Argumentation Framework (AF)	Definition 1.
	$\langle \mathsf{A}, \mathsf{R}, \mathsf{T} \rangle$	Bipolar AF	Definition 4.
	$\langle \mathbb{A}, \mathbb{R}, \mathbb{T} \rangle$ (or simply Δ)	High-Order AFs with Necessities (HOAFN)	Definition 5.
	Δ_{AF}	AF for HOAFN Δ (also called AF derived from Δ)	Definition 12.
Constants	$\mathtt{in}, \mathtt{out}, \mathtt{undec}$	Argument statuses	Definition 11.
	$\mathtt{true}, \mathtt{false}, \mathtt{unknown}$	Truth values	Page 8.
	$\mathsf{a}, \mathsf{b}, \mathsf{c}, \mathsf{d}, \mathsf{rn}, \mathsf{ns}, \mathsf{w}, \mathsf{p}, \mathsf{d}, \mathsf{pt}$	Specific arguments	Page 2.
Variables	a, b, c, d, e, f	Generic arguments	Page 2.
	g	Goal argument (in AF context)	Page 2.
	ϱ	Goal element (in HOAFN context)	Page 3.
HOAFN Elements	α (with $\alpha \in \mathbb{R}$)	Attack name	Page 2.
	β (with $\beta \in \mathbb{T}$)	Support name	Page 2.
	γ (with $\gamma \in \mathbb{R} \cup \mathbb{T}$)	Either an attack or a support name	Page 5.
	ϑ (with $\vartheta \in \mathbb{A} \cup \mathbb{R} \cup \mathbb{T}$)	Generic element	Page 5.
	$\mathbf{s}(\gamma)$ (with $\mathbf{s}(\gamma) \in \mathbb{A}$)	Function returning the source argument of γ	Page 5.
	$\mathbf{t}(\gamma)$ (with $\mathbf{t}(\gamma) \in \mathbb{A} \cup \mathbb{R} \cup \mathbb{T}$)	Function returning the target element of γ	Page 5.
	$\mathbf{b}(\gamma)$ (with $\mathbf{b}(\gamma) \in \mathbb{A} \times \mathbb{A} \cup \mathbb{R} \cup \mathbb{T}$)	Function returning the pair (source,target) of γ	Page 5.
Semantics	$\mathcal{gr}, \mathcal{co}, \mathcal{pr}, \mathcal{st}$	Argumentation semantics	Page 2.
	$\mathcal{wf}, \mathcal{ps}, \mathcal{ms}, \mathcal{ts}$	Partial stable model semantics	Page 28.
	$\mathcal{s}$ (with $\mathcal{s} \in \{\mathcal{gr}, \mathcal{co}, \mathcal{pr}, \mathcal{st}\}$)	Generic argumentation semantics	Page 2.
Sets	S (with $S \subseteq \mathsf{A}$)	Generic subset of arguments (in AF context)	Page 4.
	S (with $S \subseteq \mathbb{A} \cup \mathbb{R} \cup \mathbb{T}$)	Generic subset of elements (in HOAFN context)	Page 5.
	$\mathbf{def}(S)$	Function returning defeated arguments/elements	Definition 2.
	$\mathbf{acc}(S)$	Function returning acceptable arguments/elements	Definition 2.
Extensions	E (with $E \subseteq \mathsf{A}$)	An $\mathcal{s}$-extension (in AF context)	Page 9.
	E (with $E \subseteq \mathbb{A} \cup \mathbb{R} \cup \mathbb{T}$)	An $\mathcal{s}$-extension (in HOAFN context)	Page 3.
	$\mathcal{E}_{\mathcal{s}}$ (with $\mathcal{s} \in \{\mathcal{gr}, \mathcal{co}, \mathcal{pr}, \mathcal{st}\}$)	Set of $\mathcal{s}$-extensions	Page 4.
	$\mathbf{v}_E(\vartheta)$ or simply $\mathbf{v}(\vartheta)$ if E is understood (with $\mathbf{v}_E(\vartheta) \in \{\mathtt{true}, \mathtt{false}, \mathtt{unknown}\}$)	Function evaluating element ϑ w.r.t. E	Page 8.
Interaction set	$C = \overline{\vartheta}_\Delta$ or simply $\overline{\vartheta}$ (with $\overline{\vartheta}_\Delta \subseteq \mathbb{A} \cup \mathbb{R} \cup \mathbb{T}$)	Interaction set of ϑ w.r.t. HOAFN Δ	Definition 18.
	$\mathcal{G}_C$	Interaction graph for C	Page 14.
	$\mathbf{r}(C)$ (with $\mathbf{r}(C) \subseteq \mathbb{A} \cup \mathbb{R} \cup \mathbb{T}$)	Function returning the root elements of C	Page 14.
Update	u	Generic update	Page 1.
	$u = +\vartheta$ (or simply $+\vartheta$)	Single positive update	Page 15.
	$u = +C$ (or simply $+C$)	Multiple positive updates	Page 14.
	$u = -\vartheta$ (or simply $-\vartheta$)	Negative update	Page 14.
	$u(\Delta) = \widehat{\Delta} = \langle \widehat{\mathbb{A}}, \widehat{\mathbb{R}}, \widehat{\mathbb{T}} \rangle$	updated HOAFN	Page 3.
Functions	$\mathbf{inf}(u, \Lambda, E)$	Influenced set of Λ w.r.t. u and E	Page 9.
	$\mathbf{red}(u, \Lambda, E)$	Reduced AF of Λ w.r.t. u and E	Page 9.
	$\mathbf{RAFNtoAF}(E)$, $\mathbf{AFtoRAFN}(E)$	Mapping RAFN to AF extensions and vice-versa	Definition 14.
	$\mathbf{ASAFtoAF}(E)$, $\mathbf{AFtoASAF}(E)$	Mapping ASAF to AF extensions and vice-versa	Definition 15.
	$\mathbf{randElement}, \mathbf{randAcyclicSet}, \mathbf{randDisjointSets}$	Functions returning random items	Algorithm 5.
AF Problems and Solvers	$\mathsf{CA}_{\mathcal{s}}$ (with $\mathcal{s} \in \{\mathcal{gr}, \mathcal{co}, \mathcal{pr}, \mathcal{st}\}$)	Credulous Acceptance Problems under $\mathcal{s}$	Page 2.
	$\mathsf{SA}_{\mathcal{s}}$ (with $\mathcal{s} \in \{\mathcal{gr}, \mathcal{co}, \mathcal{pr}, \mathcal{st}\}$)	Skeptical Acceptance Problem under $\mathcal{s}$	Page 4.
	$\mathsf{SE}_{\mathcal{s}}$ (with $\mathcal{s} \in \{\mathcal{gr}, \mathcal{co}, \mathcal{pr}, \mathcal{st}\}$)	Single Extension Problem under $\mathcal{s}$	Page 9.
	AFS	External AF solver for CA and SE problems	Algorithm 1.
	$\mathsf{AFS}_{\mathcal{s}}^{\mathsf{CA}}(\Lambda, g)$	Function solving $\mathsf{CA}_{\mathcal{s}}$ for input AF Λ and goal g	Algorithm 2.
	$\mathsf{AFS}_{\mathcal{s}}^{\mathsf{SE}}(\Lambda)$	Function solving $\mathsf{SE}_{\mathcal{s}}$ for input AF Λ	Algorithm 1.

Appendix B. Proofs

To prove the results stated in the paper, we use some results linking argumentation semantics to Partial Stable Models (PSMs) of logic programs [54], that are recalled next. We briefly summarize the basic concepts underlying the notion of partial stable models [113].

A (normal, logic) program is a set of rules r of the form $a \leftarrow b_1 \wedge \cdots \wedge b_n$, with $n \geq 0$, where a is an atom, called head and denoted by $head(r)$, and $b_1 \wedge \cdots \wedge b_n$ is a conjunction of literals, called body and denoted by $body(r)$. With a little abuse of notation

$body(r)$ also denotes the set of literals in the body of r, whereas $body^+(r)$ (resp. $body^-(r)$) denotes the set of positive (resp. negative) literals in $body(r)$. We consider programs without function symbols. Given a program P, $ground(P)$ denotes the set of all ground instances of the rules in P. The Herbrand Base of a program P, *i.e.* the set of all ground atoms which can be constructed using predicate and constant symbols occurring in P, is denoted by B_P, whereas $\neg B_P$ denotes the set $\{\neg a \mid a \in B_P\}$. Analogously, for any set $S \subseteq B_P \cup \neg B_P$, $\neg S$ denotes the set $\{\neg a \mid a \in S\}$, where $\neg\neg a = a$. Given $I \subseteq B_P \cup \neg B_P$, $pos(I)$ (resp., $neg(I)$) stands for $I \cap B_P$ (resp., $\neg I \cap B_P$). I is *consistent* if $pos(I) \cap \neg neg(I) = \emptyset$, otherwise I is *inconsistent*.

A set $I \subseteq B_P \cup \neg B_P$ is an *interpretation* of P if I is consistent; I is *total* if $pos(I) \cup neg(I) = B_P$, *partial* otherwise. For any partial interpretation I of a program P, atoms in $pos(I)$ (resp. in $neg(I)$, $B_P \setminus (pos(I) \cup neg(I))$) are said to be *true* (resp. *false*, *undefined*) w.r.t. I. The truth value of an atom a w.r.t. an interpretation I is denoted by $v_I(a)$. We assume the truth value ordering $\mathtt{false} < \mathtt{undec} < \mathtt{true}$.

A partial interpretation M of a program P is a *partial model* of P if for each rule $r \in ground(P)$, $v_M(head(r)) \geq v_M(body(r))$.

Given a program P and a partial model M, the reduct of P w.r.t. M, denoted by P^M, is obtained from $ground(P)$ by replacing each negated literal with its truth value in M. Clearly, rules having the truth value $\mathtt{false}$ in the body can be deleted, and the truth value $\mathtt{true}$ can be deleted from the body of rules.

As P^M is a positive program, the minimal Herbrand model of P can be obtained as the least fixpoint of its immediate consequence operator $\mathcal{T}_{P^M}$, denoted by $\mathcal{T}^\omega_{P^M}(\emptyset)$, containing true and undefined atoms. Let Ψ_{P^M} be the set of atoms which are either true or false w.r.t. $\mathcal{T}^\omega_{P^M}(\emptyset)$ (false atoms are those in B_P that do not occur in $\mathcal{T}^\omega_{P^M}(\emptyset)$ neither as true atoms nor as undefined atoms), then Ψ_{P^M} is minimal w.r.t. $pos(\Psi_{P^M})$ and maximal w.r.t. $neg(\Psi_{P^M})$.

Let P be a program and M a partial model for P. Then M is a *Partial Stable Model (PSM)* of P iff $M = \Psi_{P^M}$.

The set of partial stable models of a logic program P, denoted by $\mathcal{ps}(P)$, defines a meet semi-lattice. The *well-founded* model (denoted by $\mathcal{wf}(P)$) and the *maximal-stable* models $\mathcal{ms}(P)$[15] are defined by considering $\subseteq$-minimal and $\subseteq$-maximal elements. The set of (total) *stable* models (denoted by $\mathcal{ts}(P)$) is obtained by considering the maximal-stable models which are total.

The set of partial stable (resp., well-founded, maximal-stable, (total) stable) models of P will be denoted by $\mathcal{ps}(P)$ (resp., $\mathcal{wf}(P)$, $\mathcal{ms}(P)$, $\mathcal{ts}(P)$).

Example 22. Consider the program $P = \{\mathtt{a} \leftarrow \neg\mathtt{b};\ \mathtt{b} \leftarrow \neg\mathtt{a};\ \mathtt{c} \leftarrow \neg\mathtt{a} \wedge \neg\mathtt{b} \wedge \neg\mathtt{d};\ \mathtt{d} \leftarrow \neg\mathtt{c}\}$. The set of PSMs of P is $\mathcal{ps}(P) = \{\emptyset, \{\neg\mathtt{c}, \mathtt{d}\}, \{\mathtt{a}, \neg\mathtt{b}, \neg\mathtt{c}, \mathtt{d}\}, \{\neg\mathtt{a}, \mathtt{b}, \neg\mathtt{c}, \mathtt{d}\}\}$. Then, $\mathcal{wf}(P) = \{\emptyset\}$, and $\mathcal{ms}(P) = \mathcal{ts}(P) = \{\{\mathtt{a}, \neg\mathtt{b}, \neg\mathtt{c}, \mathtt{d}\}, \{\neg\mathtt{a}, \mathtt{b}, \neg\mathtt{c}, \mathtt{d}\}\}$.

Propositional programs Given a set of symbols $\Lambda = \{a_1, ..., a_n\}$, a *(propositional) program* over Λ is a set of $|\Lambda|$ rules $a_i \leftarrow body_i$ ($1 \leq i \leq n$), where every $body_i$ is a propositional formula defined over Λ. The semantics of a propositional program P, defined over a given alphabet Λ, is given in terms of the set $\mathcal{ps}(P)$ of its PSMs that are obtained as follows: *i*) P is first rewritten into a set of standard (ground) logic rules P', whose bodies contain conjunction of literals (even by adding fresh symbols to the alphabet)[16]; *ii*) next, the set of PSMs of P' is computed; *iii*) finally, fresh literals added to Λ in the first step are deleted from the models. It is worth noting that for propositional programs we can assume as Herbrand Base the set of (ground) atoms occurring in the program.

Theorem 1. *Let $\Delta = \langle \mathbb{A}, \mathbb{R}, \mathbb{T} \rangle$ be a HOAFN, $\Delta_{AF} = \langle \mathsf{A}, \mathsf{R} \rangle$ its AF, $E \subseteq (\mathbb{A} \cup \mathbb{R} \cup \mathbb{T})$, $E_{AF} \subseteq \mathsf{A}$, and $\mathcal{s} \in \{co, pr, st, gr\}$ a semantics. It holds that E is an $\mathcal{s}$-extension of Δ iff* $\mathbf{HOAFNtoAF}_\Delta(E)$ *is an $\mathcal{s}$-extension of Δ_{AF}. Equivalently, it holds that E_{AF} is an $\mathcal{s}$-extension of Δ_{AF} iff* $\mathbf{AFtoHOAFN}_\Delta(E_{AF})$ *is an $\mathcal{s}$-extension of Δ.*

Proof. Recently, the semantics of RAFN has been given in terms of Partial Stable Models (PSMs) of a corresponding (normal, ground) logic program [54]. Particularly, given a RAFN $\Delta = \langle \mathbb{A}, \mathbb{R}, \mathbb{T} \rangle$, the propositional program derived from Δ (*i.e.*, P_Δ) contains, for each $\vartheta \in (\mathbb{A} \cup \mathbb{R} \cup \mathbb{T})$, a rule:

$$\vartheta \leftarrow \bigwedge_{\alpha \in \mathbb{R} \wedge \mathbf{t}(\alpha) = \vartheta} (\neg\alpha \vee \neg\mathbf{s}(\alpha)) \wedge \bigwedge_{\beta \in \mathbb{T} \wedge \mathbf{t}(\beta) = \vartheta} (\neg\beta \vee \mathbf{s}(\beta)). \tag{B.1}$$

Then, it was shown that the set of fulfillments of complete extensions[17] of Δ coincides with the set of PSMs of P_Δ, that is $\widetilde{\mathcal{E}_{co}}(\Delta) = \mathcal{ps}(P_\Delta)$. The same result holds for the case of AFs [115], where the propositional program derived from an AF $\langle \mathsf{A}, \mathsf{R} \rangle$ contains, for each argument $a \in \mathsf{A}$ a rule:

$$a \leftarrow \bigwedge_{(b,a) \in \mathsf{R}} \neg b. \tag{B.2}$$

We now show that P_Δ can be rewritten into a program by replacing positive literals and disjunctions in rules' body of Equation (B.1) with negated literals and conjunctions, respectively, as in Equation (B.2).

Any attack $\alpha \in \mathbb{R}$ in Δ corresponds to five attacks in Δ_{AF}, that are (α, α^-), $(\alpha^-, \alpha^\vee)$, $(\alpha^\vee, \mathbf{t}(\alpha))$, $(\mathbf{s}(\alpha), \mathbf{s}(\alpha)^-)$, $(\mathbf{s}(\alpha)^-, \alpha^\vee)$, which in turn correspond to the following set of rules of $P_{\Delta_{AF}}$:

[15] Corresponding to the *preferred extensions* of [114].

[16] A rule $a \leftarrow (b \vee c) \wedge (d \vee e)$ is rewritten as $a \leftarrow \neg a_1 \wedge \neg a_2$, $a_1 \leftarrow \neg b \wedge \neg c$ and $a_2 \leftarrow \neg d \wedge \neg e$.

[17] Given an $\mathcal{s}$-extension E, $\widetilde{E} = E \cup \{\neg a \mid a \in \mathbf{def}(E)\}$ denotes the fulfillment of E.

$r_1 : \mathbf{t}(\alpha) \leftarrow \neg\alpha^{\vee}$
$r_2 : \alpha^{\vee} \leftarrow \neg\mathbf{s}(\alpha)^{-} \wedge \neg\alpha^{-}$
$r_3 : \alpha^{-} \leftarrow \neg\alpha$
$r_4 : \mathbf{s}(\alpha)^{-} \leftarrow \neg\mathbf{s}(\alpha)$

As each literal in $\{\mathbf{s}(\alpha)^{-}, \alpha^{\vee}, \alpha^{-}\}$ is defined through one (fixed) rule, and appears in exactly one rule's body, the rules given above can be directly substituted, by means of equivalence steps, with other rules. Particularly, r_1, r_2, r_3, and r_4 can be replaced with the equivalent rule $r : \mathbf{t}(\alpha) \leftarrow \neg\mathbf{s}(\alpha) \vee \neg\alpha$. Note that, whenever $\mathbf{t}(\alpha)$ is attacked by another attack α', we obtain a rule $r' : \mathbf{t}(\alpha) \leftarrow (\neg\mathbf{s}(\alpha) \vee \neg\alpha) \wedge (\neg\mathbf{s}(\alpha') \vee \neg\alpha')$. The same reasoning can be applied to support relations. As a result we obtain an equivalent program $P_{\Delta'_{AF}}$ which is equal to P_Δ, and thus $\mathcal{ps}(P_\Delta) = \mathcal{ps}(P_{\Delta'_{AF}}) = \widetilde{\mathcal{E}_{co}}(\Delta) = \widetilde{\mathcal{E}_{co}}(\Delta'_{AF})$. As preferred, stable, semi-stable, and grounded, extensions are defined by selecting a subset of the complete extensions satisfying a given criterion (see Section 2), and the maximal, total, least-undefined, and well-founded, (partial) stable models are obtained by selecting a subset of the PSMs satisfying criteria coinciding with those used to restrict the set of complete extensions, then the result also holds for any semantics $\mathcal{s}$ (see Proposition 1 of [54]).

Given $E \in \mathcal{E}_{co}(\Delta)$ and let $E' = \mathbf{RAFNtoAF}(E)$, we now show that $E' \in \mathcal{E}_{co}(\Delta_{AF})$. Note that $E' = \mathbf{RAFNtoAF}(E)$ can be computed by extending the PSM $\widetilde{E}$ of P_Δ to the rewritten program $P_{\Delta_{AF}}$ containing additional literals. Then, α^{-} (resp. $\mathbf{s}(\alpha)^{-}$) will be in E' whenever $\neg\alpha \in \widetilde{E}$ (resp. $\neg\mathbf{s}(\alpha) \in \widetilde{E}$), meaning that $\alpha \in \mathbf{def}(E)$ (resp. $\mathbf{s}(\alpha) \in \mathbf{def}(E)$). As r_2 can be rewritten into $r'_2 : \alpha^{\vee} \leftarrow \mathbf{s}(\alpha) \wedge \alpha$, then $\alpha^{\vee} \in E'$ whenever both $\mathbf{s}(\alpha)$ and α are in E. By reasoning analogously for $\beta^{\vee}$ and β^{-}, we obtain exactly the condition identified in the **RAFNtoAF** function.

Let $E' \in \mathcal{E}_{co}(\Delta_{AF})$ and $E = \mathbf{AFtoRAFN}(E')$, we now show that $E \in \mathcal{E}_{co}(\Delta)$. As observed above $P_{\Delta'_{AF}} = P_\Delta$ and then $\mathcal{ps}(P_\Delta) = \mathcal{ps}(P_{\Delta'_{AF}}) = \widetilde{\mathcal{E}_{co}}(\Delta) = \widetilde{\mathcal{E}_{co}}(\Delta'_{AF})$. Thus, let $\widetilde{E} = (E' \cap \widetilde{(\mathbb{A} \cup \mathbb{R}} \cup \mathbb{T}))$, we have that $\widetilde{E} \in \mathcal{ps}(P_{\Delta'_{AF}})$ and thus also $\widetilde{E} \in \mathcal{ps}(P_\Delta)$, obtaining that $E \in \mathcal{E}_{co}(\Delta)$.

Consider now the case of the HOAFN Δ being an ASAF. The proof follows by reasoning analogously to the case of RAFN but replacing Equation (B.1) with:

$$\vartheta \leftarrow \varphi(\vartheta) \wedge \bigwedge_{\alpha \in \mathbb{R} \wedge \mathbf{t}(\alpha)=\vartheta} \neg\alpha \wedge \bigwedge_{\beta \in \mathbb{T} \wedge \mathbf{t}(\beta)=\vartheta} (\neg\beta \vee \mathbf{s}(\beta)) \text{ where } \varphi(\vartheta) = \begin{cases} \mathbf{s}(\vartheta) \text{ if } \vartheta \in \mathbb{R} \\ \texttt{true} \text{ otherwise} \end{cases} \quad \square \tag{B.3}$$

Proposition 1. *Let $\Delta = \langle \mathbb{A}, \mathbb{R}, \mathbb{T} \rangle$ be a HOAFN, $\Delta_{AF} = \langle \mathsf{A}, \mathsf{R} \rangle$ be the AF for Δ obtained from Definition 12 or 13, and $\Delta'_{AF} = \langle \mathsf{A}', \mathsf{R}' \rangle$ be the AF for Δ obtained from Definition 16 or 17, respectively. For any semantics $\mathcal{s} \in \{co, pr, st, gr\}$, it holds that:*

- $\mathcal{E}_{\mathcal{s}}(\Delta_{AF}) = \{\mathbf{HOAFNtoAF}(\mathbf{AFtoHOAFN}(E'_{AF})) \mid E'_{AF} \in \mathcal{E}_{\mathcal{s}}(\Delta'_{AF})\}$*; and*
- $\mathcal{E}_{\mathcal{s}}(\Delta'_{AF}) = \{E_{AF} \cap \mathsf{A}' \mid E_{AF} \in \mathcal{E}_{\mathcal{s}}(\Delta_{AF})\}$.

Proof. We start with the case of untargeted supports β in RAFNs, where differences in Δ_{AF} and Δ'_{AF} regard arguments β^{-}. Particularly, β^{-} no longer appears in Δ'_{AF} whenever β is neither attacked nor supported. Note that β^{-} is `out` in every complete extension of Δ_{AF} (β is always `in` as it is unattacked). Thus, the removal of argument β^{-} will not affect the label of $\beta^{\vee}$ and $\mathbf{t}(\beta)$. Observe that this also holds for ASAFs as the mapping of supports is equal (see Fig. 5).

For the case of untargeted attacks α in RAFNs we can reason analogously as before. Particularly, the differences between Δ_{AF} and Δ'_{AF} concern arguments $\mathbf{s}(\alpha)$, $\alpha^{\vee}$ and α^{-}. Those arguments are not appearing in Δ'_{AF} and are replaced by an attack from $\mathbf{s}(\alpha)$ to $\mathbf{t}(\alpha)$ whenever α is untargeted. Note that, α^{-} is `out` in every complete extension of Δ_{AF} (as α is always `in` because it is unattacked). Thus, the removal of argument α^{-} will not affect the label of $\alpha^{\vee}$. Moreover, the label of $\alpha^{\vee}$ is equal to that of $\mathbf{s}(\alpha)$ in every complete extension of Δ_{AF} as the label of α^{-} is always `out` and $\mathbf{s}(\alpha)$ and $\alpha^{\vee}$ are not attacked by other arguments. Thus, the attack from $\mathbf{s}(\alpha)$ to $\mathbf{t}(\alpha)$ will have the same effect on the complete extensions of the RAFN as that from $\alpha^{\vee}$ to $\mathbf{t}(\alpha)$.

Thus, the elements of Δ continue to have the same status in every complete extension, prescribing the same set of complete extensions of Δ w.r.t. AFs Δ_{AF} and Δ'_{AF} (*i.e.*, they can be computed by applying either $\mathbf{AFtoHOAFN}(E_{AF})$ or $\mathbf{AFtoHOAFN}(E'_{AF})$). As the set of complete extensions of the HOAFN does not change, the result also holds for grounded, preferred, and stable semantics, that are refinements of the complete semantics. $\square$

Proposition 2. *For any interaction set C we have that:*

1. *if $\mathcal{G}_C$ contains an argument a, then $\mathbf{r}(C) = \{a\}$,*
2. *otherwise, (i) if $\mathcal{G}_C$ is acyclic, then $\mathbf{r}(C)$ is unique, (ii) if $\mathcal{G}_C$ is a (reversed) tree, then $\mathbf{r}(C)$ coincides with the root of $\mathcal{G}_C$, and (iii) if $\mathcal{G}_C$ is cyclic, then $\mathcal{G}_C$ contains only one cycle and $\mathbf{r}(C)$ contains all elements in the cycle.*

Proof. Case 1 follows from Definition 18 as, by construction, any interaction set C cannot contain more than one argument. As for Case 2.i, we first we recall that, to let C be an interaction set, there must exists an element $\vartheta \in C$ s.t. $\overline{\vartheta} = C$. Thus, $\mathcal{G}_C$ is connected (i.e., it has one connected component only) and thus $\mathbf{r}(C)$ is the element of C having out-degree 0 in $\mathcal{G}_C$ as $\mathcal{G}_C$ is acyclic. We now consider Case 2.ii. If $\mathcal{G}_C$ is a (reversed) tree, then it is acyclic and, reasoning as in Case 2.i, $\mathbf{r}(C)$ is the unique element of C having out-degree 0 in $\mathcal{G}_C$. Finally, consider Case 2.iii. Using Case 1 we have that $\mathcal{G}_C$ cannot contain arguments (otherwise $\mathbf{r}(C)$ would be a singleton). Moreover, to let C be an interaction set, there must exists an element $\vartheta \in C$ s.t. $\overline{\vartheta} = C$. Thus, $\mathcal{G}_C$ has a unique connected component. Now, as $\mathcal{G}_C$ has one component only and does not contain arguments, if there are two cycles then there must exist (at least) one attack or support element ϑ having two outgoing edges in $\mathcal{G}_C$, which is an absurd (as the target of ϑ is unique). Thus $\mathcal{G}_C$ contains only one cycle. Now assume by contradiction that $\mathbf{r}(C)$ does not contain all elements of the cycle. Then either (a) $\mathbf{r}(C)$ does not contain all elements of the cycle and contains at least one element or (b) $\mathbf{r}(C)$ does not contain any element of the cycle. In the first case, observe that if at least one element of the cycle (say ϑ) appears in $\mathbf{r}(C)$ then this implies by definition of $\mathbf{r}(C)$ that all elements in the cycle also appear in $\mathbf{r}(C)$, contradiction. As for case (b), recall that $\mathcal{G}_C$ is connected and thus there must exist an element in the cycle having two outgoing edges, contradiction. Therefore, $\mathcal{G}_C$ contains only one cycle and $\mathbf{r}(C)$ contains every element in the cycle. □

Theorem 2. *Given a HOAFN $\Delta = \langle \mathbb{A}, \mathbb{R}, \mathbb{T} \rangle$, a semantics $\mathcal{s} \in \{co, pr, st, gr\}$, an $\mathcal{s}$-extension E of Δ, a goal $\varrho \in E$, and a cyclic update $u = -\gamma$ (resp. $u = +C$) such that $\gamma \notin \mathbb{A}$ (resp. $\mathbf{r}(C) \cap \mathbb{A}_{\mathbb{U}} = \emptyset$) and $\varrho \notin \overline{\gamma}$. Then u is irrelevant w.r.t. E and ϱ iff (i) u is negative or $\mathcal{s} \in \{co, pr, gr\}$, or (ii) u is positive, $\mathcal{s} = st$, and there exists either $\alpha \in (C \cap \mathbb{R}_U)$ s.t. $\mathbf{v}_E(\mathbf{s}(\alpha)) = \texttt{false}$ or $\beta \in (C \cap \mathbb{T}_U)$ s.t. $\mathbf{v}_E(\mathbf{s}(\alpha)) = \texttt{true}$.*

Proof. As for case (i), if u is negative then $E \setminus \overline{\gamma}$ is an $\mathcal{s}$-extension of $u(\Delta)$, otherwise (*i.e.* u is positive and $\mathcal{s} \neq st$), there always exists at least one $\mathcal{s}$-extension $E' \supseteq E$ for $u(\Delta)$ (constructed as in Theorem 4). For case (ii), the limitations imposed over the update u allow us to expand the stable extension E. Particularly, there always exists at least one stable extension $E' \supseteq E$ as elements in the unique cycle in $\mathcal{G}_C$ (see Proposition 2) assume the truth value `true`/`false` according to that of their respective source argument. More formally, assume by contradiction that E cannot be expanded, or equivalently that E is a preferred extension for $u(\Delta)$. If this holds then all elements in C are labeled undecided w.r.t. E and $u(\Delta)$. Then, all attacks $\alpha \in C$ and supports $\beta \in C$ assume `undec` as their status w.r.t. E and $u(\Delta)$. This would imply that $\mathbf{v}_E(\mathbf{s}(\alpha)) \neq \texttt{false}$ and $\mathbf{v}_E(\mathbf{s}(\beta)) \neq \texttt{true}$, which is a contradiction as the statement requires that the status of the source of at least one attack (or support) is false (resp. true). □

Theorem 3. *Given a HOAFN $\Delta = \langle \mathbb{A}, \mathbb{R}, \mathbb{T} \rangle$, a semantics $\mathcal{s} \in \{co, pr, st, gr\}$, an $\mathcal{s}$-extension E of Δ, a goal $\varrho \in E$, a cyclic irrelevant update $u = -\gamma$ (resp. $u = +C$) w.r.t. E, and ϱ such that $\gamma \notin \mathbb{A}$ (resp. $(\mathbf{r}(C) \cap \mathbb{A}_{\mathbb{U}}) = \emptyset$) and $\varrho \notin \overline{\gamma}$. Then, $\widehat{E} = E \setminus \overline{\gamma}$ if $u = -\gamma$ (resp. $\widehat{E} = E \cup$ **AFtoHOAFN**(E') if $u = +C$, where E' is an $\mathcal{s}$-extension of Λ_u), is an $\mathcal{s}$-extension of $u(\Delta)$.*

Proof. Consider the case where u is a negative update (*i.e.*, $u = -\gamma$). As u is irrelevant w.r.t. extension E and the goal ϱ, there exists an $\mathcal{s}$-extension $\widehat{E}$ for $u(\Delta)$ such that $\varrho \in \widehat{E}$ and $L_E(\vartheta) = L_{\widehat{E}}(\vartheta)$ for every $\vartheta \in \left((\mathbb{A} \cap \widehat{\mathbb{A}}) \cup (\mathbb{R} \cap \widehat{\mathbb{R}}) \cup (\mathbb{T} \cap \widehat{\mathbb{T}})\right)$, where the updated HOAFN is derived from Δ by deleting all elements in $\overline{\gamma}$, from which the statement follows.

As for the addition case, the result follows from the fact that Λ_u is the *reduced AF* [49] w.r.t Λ (the AF derived from Δ) and **HOAFNtoAF**(E) for the set of updates corresponding to elements in C. Thus, E' is an $\mathcal{s}$-extension for the reduced AF and can be merged with **HOAFNtoAF**(E) to obtain an updated extension for $u(\Lambda)$ (Theorem 1 in [49]), and thus, using Theorem 1 (in this paper) $E \cup$ **AFtoHOAFN**(E') is an $\mathcal{s}$-extension for $u(\Delta)$. □

Theorem 4. *Given a HOAFN $\Delta = \langle \mathbb{A}, \mathbb{R}, \mathbb{T} \rangle$, a semantics $\mathcal{s} \in \{co, pr, st, gr\}$, an $\mathcal{s}$-extension E of Δ, a goal $\varrho \in E$, and a negative (resp. positive) update $u = -\vartheta$ (resp. $u = +C$) which is acyclic and irrelevant w.r.t. E and ϱ, and $\varrho \notin \overline{\vartheta}$. Let $u(\Delta) = \langle \widehat{\mathbb{A}}, \widehat{\mathbb{R}}, \widehat{\mathbb{T}} \rangle$ be the updated HOAFN. Then, $\widehat{E} = E \cap (\widehat{\mathbb{A}} \cup \widehat{\mathbb{R}} \cup \widehat{\mathbb{T}})$ (resp. $\widehat{E} = \{\varphi \mid \varphi \in (\widehat{\mathbb{A}} \cup \widehat{\mathbb{R}} \cup \widehat{\mathbb{T}}) \wedge \mathbf{v}'(\varphi) = \texttt{true}\}$, where $\mathbf{v}'$ is the temporary valuation for $u(\Delta)$ w.r.t. E) is an $\mathcal{s}$-extension of $u(\Delta)$ containing ϱ.*

Proof. Consider first the case where u is negative, *i.e.*, $u = -\vartheta$. As u is irrelevant for Δ w.r.t. E, from Definition 20, we have that $L_{\widehat{E}}(\varphi) = L_E(\varphi)$ for every $\varphi \in \left((\mathbb{A} \cap \widehat{\mathbb{A}}) \cup (\mathbb{R} \cap \widehat{\mathbb{R}}) \cup (\mathbb{T} \cap \widehat{\mathbb{T}})\right)$. As $\widehat{\mathbb{A}} \subseteq \mathbb{A}$, $\widehat{\mathbb{R}} \subseteq \mathbb{R}$, and $\widehat{\mathbb{T}} \subseteq \mathbb{T}$ then $\texttt{in}(L_{\widehat{E}}) = \texttt{in}(L_E) \setminus \overline{\vartheta}$, from which the result holds as $\texttt{in}(L_{\widehat{E}}) = \widehat{E}$ and $\texttt{in}(L_E) = E$.

Consider now the addition case. Let P_Δ (resp. $P_{\widehat{\Delta}}$) be the propositional program derived from Δ (resp. $\widehat{\Delta} = u(\Delta)$) [54]. Recall that $\widetilde{\mathcal{E}}_{co}(\Delta) = ps(P_\Delta)$ and $\widetilde{\mathcal{E}}_{co(\widehat{\Delta})} = ps(P_{\widehat{\Delta}})$ hold. Let M be a PSM of P_Δ (i.e., $M \in ps(P_\Delta)$). Observe that, as u is irrelevant, there must exists an $\mathcal{s}$-extension $\widehat{E}$ of $\widehat{\Delta}$ s.t. $\widehat{E} = E \cup E'$. Thus, there exists a PSM $\widehat{M}$ for $P_{\widehat{\Delta}}$ s.t. $\widehat{M} = M \cup M'$.

Let P^M_Δ (resp., $P^M_{\widehat{\Delta}}$) be the propositional program obtained from P_Δ (resp. $P_{\widehat{\Delta}}$) in the following way: for any rule r of P_Δ (resp. $P_{\widehat{\Delta}}$) s.t. $head(r) \in (\mathbb{A} \cup \mathbb{R} \cup \mathbb{T})$ replace any atom $a \in (\mathbb{A} \cup \mathbb{R} \cup \mathbb{T})$ in the body of r with `true` (resp. `false`, `unknown`) if $a \in pos(M)$ (resp. $a \in neg(M)$, $a \notin pos(M) \cup neg(M)$), and then replace $body(r)$ with its 3-valued valuation.

We have that $P^M_\Delta \subseteq P^M_{\widehat{\Delta}}$ and, as the only PSM of P^M_Δ is M and it is trivially determined (as bodies are truth values), each PSM in $ps(P^M_{\widehat{\Delta}})$ has the form $M \cup M'$ with M' being a PSM of $P^M_{\widehat{\Delta}} \setminus P^M_\Delta$.

Note that M' can be obtained in the same way that the temporary valuation $\mathbf{v}'$ for $\widehat{\Delta}$ w.r.t. E is obtained, as Equations (1)-(2) coincide with rules of $P^M_{\widehat{\Delta}} \setminus P^M_\Delta$. As $(M \cup M') \in ps(P^M_{\widehat{\Delta}}) \subseteq ps(P_{\widehat{\Delta}})$, we have that $\left(pos(M) \cup pos(M')\right) = (E \cup E') = \left(\{\vartheta \mid \mathbf{v}_E(\vartheta) = \texttt{true}\} \cup \{\vartheta \mid \mathbf{v}'(\vartheta) = \texttt{true}\}\right) \in \mathcal{E}_{co}(\widehat{\Delta})$, from which the result holds for $\mathcal{s} = co$. As stable extension exists (otherwise u is not irrelevant), the result holds for any $\mathcal{s}$ semantics. □

Theorem 5. *Let Δ be a HOAFN, $\mathcal{s} \in \{co, pr, st, gr\}$ a semantics, E an $\mathcal{s}$-extension of Δ, $\varrho \in E$ a goal, and $u = -\gamma$ (resp. $u = +C$) an acyclic update for Δ such that $\gamma \notin \mathbb{A}$ (resp. $\mathbf{r}(C) \notin \mathbb{A}_{\mathbb{U}}$) and $\varrho \notin \overline{\gamma}$. Also, let $\mathbf{v}$ be the valuation for Δ w.r.t. E, and $\mathbf{v}'$ be the temporary valuation for $u(\Delta)$ w.r.t. E and u. Then, u is irrelevant for Δ w.r.t. the $\mathcal{s}$-extension E and the goal ϱ if the semantics $\mathcal{s}$ occurs in the cell identified by row $\mathbf{V_s}(\gamma)$ and column $\mathbf{V_t}(\gamma)$ of Table 1 (resp. Table 2).*

Proof. Consider first the case where $u = -\gamma$ with $\gamma \in (\mathbb{R} \cup \mathbb{T})$ is a negative update. Let E be an $\mathcal{s}$-extension of Δ, Δ_{AF} be the AF for Δ and $E_{AF} = \mathbf{HOAFNtoAF}_{\Delta}(E)$ the $\mathcal{s}$-extension for the AF Δ_{AF} obtained from E. Assume by contradiction that the result does not hold, *i.e.*, $\mathcal{s}$ is in the cell $(\mathbf{V_s}(\gamma), \mathbf{V_t}(\gamma))$ of Table 1, but u is not irrelevant. If this holds then it might be the case that the update $u_{AF} = -(a, b)$ is not irrelevant for Δ_{AF} w.r.t. E_{AF}, with:

- $a = \gamma$ and $b = \mathbf{t}(\gamma)$, if Δ is an ASAF and $\gamma \in \mathbb{R}$;
- $a = \gamma^{\vee}$ and $b = \mathbf{t}(\gamma)$, otherwise.

Note that, $\mathbf{v}_{E_{AF}}(a) = \mathbf{V_s}(\gamma)$ and $\mathbf{v}_{E_{AF}}(b) = \mathbf{V_t}(\gamma)$, and thus this would result in an absurd, as u_{AF} has been proved to be irrelevant for Δ_{AF} w.r.t. E_{AF} [116].

Consider now the case of positive update $u = +\gamma$ with $\gamma \in ((\mathbb{R}_{\mathbb{U}} \cup \mathbb{T}_{\mathbb{U}}) \backslash (\mathbb{R} \cup \mathbb{T}))$. Assume by contradiction that the result does not hold, *i.e.*, $\mathcal{s}$ is in the cell $(\mathbf{V_s}(\gamma), \mathbf{V_t}(\gamma))$ of Table 2, but u is not irrelevant. If this holds then it might be the case that the update $u_{AF} = +(c, d)$ is not irrelevant for Δ_{AF} w.r.t. E_{AF}, with:

- $u_{AF} = +(c, d)$ with $(c, d) = (\gamma, \mathbf{t}(\gamma))$ (resp. $(\gamma^{\vee}, \mathbf{t}(\gamma))$) if Δ is an ASAF and $\gamma \in \mathbb{R}_{\mathbb{U}}$ (resp. otherwise);
- $\Delta_{AF} = \langle \mathsf{A} = \mathsf{A}^*, \mathsf{R} = \mathsf{R}^* \setminus \{(c, d)\} \rangle$ where $\Lambda^* = \langle \mathsf{A}^*, \mathsf{R}^* \rangle$ is the AF for $u(\Delta)$.
- $E_{AF} = \mathbf{HOAFNtoAF}_{u(\Delta)}(\{\vartheta \mid \mathbf{v}'(\vartheta) = \texttt{true}\})$ where $\mathbf{v}'$ is the temporary valuation for $u(\Delta)$ under the semantics $\mathcal{s}$ w.r.t. E and u (as prescribed by Theorem 4).

Note that, $\mathbf{v}_{E_{AF}}(c) = \mathbf{V_s}(\gamma)$ and $\mathbf{v}_{E_{AF}}(d) = \mathbf{V_t}(\gamma)$, and thus this would result in an absurd, as u_{AF} has been proved to be irrelevant for Δ_{AF} w.r.t. E_{AF} [116]. □

Theorem 6. *Let $\Delta = \langle \mathbb{A}, \mathbb{R}, \mathbb{T} \rangle$ be a HOAFN, $\mathcal{s} \in \{co, pr, st, gr\}$ a semantics, E an $\mathcal{s}$-extension of Δ, $\varrho \in E$ a goal, and $u = -a$ with $a \in \mathbb{A}$ an acyclic update for Δ such that $\varrho \notin \overline{a}$. Then, u is irrelevant for Δ w.r.t. E and ϱ if, for every $\gamma \in \overline{a}$ such that $\mathbf{s}(\gamma) = a$, it holds that the update $-\gamma$ is irrelevant for Δ w.r.t. E and ϱ as per Theorem 5.*

Proof. Let $\Gamma = \{\gamma \mid \gamma \in (\mathbb{R} \cup \mathbb{T}) \wedge \mathbf{s}(\gamma) = a\}$. As any update $-\gamma$ with $\gamma \in \Gamma$ is irrelevant for Δ w.r.t. E, for Theorem 4 we have that, let $\Delta^* = \langle \mathbb{A}, (\mathbb{R} \setminus \overline{\Gamma}), (\mathbb{T} \setminus \overline{\Gamma}) \rangle$ the HOAFN obtained by applying updates $-\gamma$, $E^* = E \setminus \overline{\Gamma}$ is an $\mathcal{s}$-extension of Δ^*. Thus, $L_E(\vartheta) = L_{E^*}(\vartheta)$ for any $\vartheta \in \Big((\mathbb{A} \cap \mathbb{A}^*) \cup (\mathbb{R} \cap \mathbb{R}^*) \cup (\mathbb{T} \cap \mathbb{T}^*)\Big)$.

Let $P_{\widehat{\Delta}}$ and P_{Δ^*} be the programs derived from $\widehat{\Delta} = u(\Delta)$ and Δ^*, respectively. Observe that $P_{\widehat{\Delta}} \subseteq P_{\Delta^*}$. Let M be the PSM of P_{Δ^*} corresponding to E^*, we denote by $P^M_{\Delta^*}$ (resp., $P^M_{\widehat{\Delta}}$) the propositional program obtained from P_{Δ^*} (resp. $P_{\widehat{\Delta}}$) in the following way: for any rule r of P_{Δ^*} (resp. $P_{\widehat{\Delta}}$) s.t. $head(r) \in \Big((\widehat{\mathbb{A}} \cap \mathbb{A}^*) \cup (\widehat{\mathbb{R}} \cap \mathbb{R}^*) \cup (\widehat{\mathbb{T}} \cap \mathbb{T}^*)\Big)$ replace any atom $a \in \Big((\widehat{\mathbb{A}} \cap \mathbb{A}^*) \cup (\widehat{\mathbb{R}} \cap \mathbb{R}^*) \cup (\widehat{\mathbb{T}} \cap \mathbb{T}^*)\Big)$ in the body of r with `true` (resp. `false`, `unknown`) if $a \in pos(M)$ (resp. $a \in neg(M)$, $a \notin pos(M) \cup neg(M)$), and then replace $body(r)$ with its 3-valued valuation. Again, observe that $P^M_{\widehat{\Delta}} \subseteq P^M_{\Delta^*}$.

Now it is sufficient to prove that $p\mathcal{s}(P^M_{\widehat{\Delta}}) \subseteq p\mathcal{s}(P^M_{\Delta^*})$ as the existence of a PSM M^* for $P^M_{\Delta^*}$ s.t. $M^* \neq \emptyset$ implies that $L_E(\vartheta) = L_{pos(M^*)}(\vartheta)$ for any $\vartheta \in \Big((\widehat{\mathbb{A}} \cap \mathbb{A}^*) \cup (\widehat{\mathbb{R}} \cap \mathbb{R}^*) \cup (\widehat{\mathbb{T}} \cap \mathbb{T}^*)\Big)$, that is $u^* = -\overline{a}_{\Delta^*}$ is irrelevant for Δ^* w.r.t. E^*. Such M^* always exists for any semantics except from the total stable, as it might prescribe no extensions in presence of cycles. However, as we know that $-a$ is acyclic, we also know that the interaction graph $\overline{a}_{\Delta^*}$ is acyclic, and this also holds for $P^M_{\Delta^*}$. Thus, as the program $P^M_{\Delta^*}$ is acyclic, there exists at least a stable extension for it. □

Corollary 1. *Let $\Delta = \langle \mathbb{A}, \mathbb{R}, \mathbb{T} \rangle$ be a HOAFN, $\mathcal{s} \in \{co, pr, st, gr\}$ a semantics, E an $\mathcal{s}$-extension of Δ, $\varrho \in E$ a goal, $u = -a$ with $a \in \mathbb{A}$ an acyclic update for Δ such that $\varrho \notin \overline{a}$, and $\Gamma = \{\gamma \mid \gamma \in \overline{a} \wedge \mathbf{s}(\gamma) = a\}$. Then, u is irrelevant for Δ w.r.t. E and ϱ if either the condition (i) $\mathbf{v}_E(a) = \texttt{false}$ and $\Gamma \subseteq \mathbb{R}$, or (ii) $\mathbf{v}_E(a) = \texttt{true}$ and $\Gamma \subseteq \mathbb{T}$ holds.*

Proof. If either case i) or ii) holds, then updates $-\gamma$ for any $\gamma \in \Gamma$ are irrelevant for Theorem 5 (as $\mathbf{V_s} = \texttt{false}$ in Table 1), and thus Theorem 7 applies. □

Theorem 7. *Let $\Delta = \langle \mathbb{A}, \mathbb{R}, \mathbb{T} \rangle$ be a HOAFN, $\mathcal{s} \in \{co, pr, st, gr\}$ a semantics, E an $\mathcal{s}$-extension of Δ, $\varrho \in E$ a goal, and $u = +C$ with $\mathbf{r}(C) \in \mathbb{A}_{\mathbb{U}}$ an acyclic update for Δ. Let $C' = \bigcup_{\gamma \in C \wedge \mathbf{s}(\gamma) = \mathbf{r}(C)} \overline{\gamma}_{u(\Delta)}$, $u' = +(C \setminus C')$, $\Delta' = u'(\Delta) = \langle \mathbb{A}', \mathbb{R}', \mathbb{T}' \rangle$, $\mathbf{v}'$ be the temporary valuation for Δ' w.r.t. E and u', and $E' = \{\vartheta \mid \vartheta \in (\mathbb{A}' \cup \mathbb{R}' \cup \mathbb{T}') \wedge \mathbf{v}'(\vartheta) = \texttt{true}\}$ be the $\mathcal{s}$-extension of Δ' expanding E (obtained as in Theorem 4). Then, u is irrelevant for Δ w.r.t. E and ϱ if for every $\gamma \in C'$ the update $+\overline{\gamma}_{u(\Delta)}$ is irrelevant for Δ' w.r.t. E' and ϱ.*

Proof. By Lemma 1 we have that u' is irrelevant for Δ w.r.t. E and, by Theorem 4, that $E' \in \mathcal{E}_s(\Delta')$. Then, by Theorem 5 we have that each update $+\overline{\gamma}_{u(\Delta)}$ is irrelevant for Δ', and by transitivity (as $E \subseteq E' \subseteq \widehat{E}$) that u is irrelevant for Δ w.r.t. E.

Lemma 1. *Let Δ be an HOAFN and E an s-extension of Δ, and $g \in E$ a goal. Then, for any acyclic update $u^* = +C$ for Δ, obtaining the updated HOAFN $\Delta^* = u^*(\Delta)$, s.t. $\mathbf{r}(C) \in \mathbb{A}$ and $\bigcup_{\gamma \in C \wedge \mathbf{s}(\gamma)=r(C)} \overline{\gamma}_{u(\Delta)} = \emptyset$, then u^* is irrelevant for Δ w.r.t. E.*

Proof. Let P_Δ and P_{Δ^*} be the programs derived from Δ and Δ^*, respectively. Observe that $P_\Delta \subseteq P_{\Delta^*}$. Let M be the PSM of P_Δ corresponding to E, we denote by P^M_Δ (resp., $P^M_{\Delta^*}$) the propositional program obtained from P_Δ (resp. P_{Δ^*}) in the following way: for any rule r of P_Δ (resp. P_{Δ^*}) s.t. $head(r) \in \Big((\mathbb{A} \cap \mathbb{A}^*) \cup (\mathbb{R} \cap \mathbb{R}^*) \cup (\mathbb{T} \cap \mathbb{T}^*)\Big)$ replace any atom $a \in \Big((\mathbb{A} \cap \mathbb{A}^*) \cup (\mathbb{R} \cap \mathbb{R}^*) \cup (\mathbb{T} \cap \mathbb{T}^*)\Big)$ in the body of r with `true` (resp. `false`, `unknown`) if $a \in pos(M)$ (resp. $a \in neg(M)$, $a \notin pos(M) \cup neg(M)$), and then replace $body(r)$ with its 3-valued valuation. Observe that $P^M_\Delta \subseteq P^M_{\Delta^*}$.

Now it is sufficient to prove that $ps(P^M_\Delta) \subseteq ps(P^M_{\Delta^*})$ as the existence of a PSM M^* for $P^M_{\Delta^*}$ s.t. $M^* \neq \emptyset$ implies that $L_E(\vartheta) = L_{pos(M^*)}(\vartheta)$ for any $\vartheta \in \Big((\mathbb{A} \cap \mathbb{A}^*) \cup (\mathbb{R} \cap \mathbb{R}^*) \cup (\mathbb{T} \cap \mathbb{T}^*)\Big)$, that is u^* is irrelevant for Δ w.r.t. E. Such M^* always exists for any semantics apart the total stable, as it might prescribe no extensions in presence of cycles. However, as we know that u^* is acyclic, then the interaction graph $\overline{C}$ is acyclic, and this is also the case for $P^M_{\Delta^*}$, guaranteeing that at least one stable extension exists. □

Corollary 2. *Let $\Delta = \langle \mathbb{A}, \mathbb{R}, \mathbb{T} \rangle$ be a HOAFN, $s \in \{co, pr, st, gr\}$ a semantics, E an s-extension of Δ, $\varrho \in E$ a goal, and $u = +C$ with $\mathbf{r}(C) \in \mathbb{A}$ an acyclic update for Δ. Then, u is irrelevant for Δ w.r.t. E and ϱ if $\mathbf{v}_E(\mathbf{t}(\gamma)) =$ `false` for every $\gamma \in C$ such that $\mathbf{s}(\gamma) = \mathbf{r}(C)$.*

Proof. Same proof strategy adopted in Corollary 2 can be applied to prove the statement. □

Proposition 3. *Let Δ be a HOAFN, $s \in \{co, pr, st, gr\}$ a semantics, u an irrelevant update for Δ w.r.t. a given s-extension E, and $\varrho \in E$ a goal. Then, for u acyclic or negative, or $s \in \{co, gr\}$, the computation of an s-extension $\widehat{E}$ of $u(\Delta)$ such that $\varrho \in \widehat{E}$ can be done in polynomial time.*

Proof. In the case of removal update the result is trivial as it corresponds to an intersection operation.

In the case of positive irrelevant acyclic updates $u = +C$, it suffices to prove that the computation of the extension $\widehat{E}$ of Theorem 4, and thus of the temporary valuation $\mathbf{v}'$ for $u(\Delta)$ w.r.t. E can be done in polynomial-time.

As u is acyclic, the interaction graph of C is acyclic, and can be computed in polynomial time. Let $\langle \vartheta_1 \dots \vartheta_n \rangle$ be any topological sorting of the nodes in the graph. The topological sorting can be computed in polynomial time, and we have that L' is computed by following the order in $\langle \vartheta_1 \dots \vartheta_n \rangle$, so that there is no recursion when computing $L'(\vartheta_i)$ for any $i \in [1, n]$. Then, the temporary valuation $\mathbf{v}'$ can be obtained from L' in linear time w.r.t. $|\mathtt{in}(L') \cup \mathtt{out}(L') \cup \mathtt{undec}(L')|$.

Consider now the case where $u = +C$ is a positive, cyclic update and $s \in \{co, gr\}$. Let P_Δ and $P_{\widehat{\Delta}}$ be the programs derived from Δ and $\widehat{\Delta} = u(\Delta)$, respectively. Let M (resp. $\widehat{M}$) be the PSM of P_Δ (resp. $P_{\widehat{\Delta}}$) corresponding to E (resp. $\widehat{E}$). Observe that $pos(M) \subseteq pos(\widehat{M})$, $neg(M) \subseteq neg(\widehat{M})$ as we assumed that $E \subseteq \widehat{E}$. We denote by $P^M_{\widehat{\Delta}}$ the propositional program obtained from $P_{\widehat{\Delta}}$ in the following way: for any rule r of $P_{\widehat{\Delta}}$ s.t. $head(r) \in (\mathbb{A} \cup \mathbb{R} \cup \mathbb{T}) \cap (pos(M) \cup neg(M))$ replace the body of r with `true` (resp. `false`) if $head(r) \in pos(M)$ (resp. $head(r) \in neg(M)$). Let $M^* = \mathcal{T}^\omega_{P^M_{\widehat{\Delta}}}(\emptyset)$ be the result (obtained in PTIME) of the application of the least fixpoint of the immediate consequence operator $\mathcal{T}_{P^M_{\widehat{\Delta}}}$ of $P^M_{\widehat{\Delta}}$. Observe that $pos(M) \subseteq pos(M^*)$, and $neg(M) \subseteq neg(M^*)$ by construction. Moreover, M^* is a PSM of $P^M_{\widehat{\Delta}}$ and $P_{\widehat{\Delta}}$, and thus $\widehat{M} = M^*$ can be returned in polynomial time. □

Theorem 8. *For any HOAFN $\Delta = \langle \mathbb{A}, \mathbb{R}, \mathbb{T} \rangle$, s-extension $E \in \mathcal{E}_s(\Delta)$ with $s \in \{co, pr, st, gr\}$, update u, and goal $\varrho \in \mathbb{A} \cup \mathbb{R} \cup \mathbb{T}$, Algorithm 4 computes $\widehat{E} \in \mathcal{E}_s(u(\Delta))$ with $\varrho \in \widehat{E}$ if ϱ is credulously accepted in $u(\Delta)$ w.r.t. semantics s, otherwise it returns $\perp$.*

Proof. It follows from Theorems 1-7, and soundness and correctness of Incr-AF-SE [49,50]. □

References

[1] T.J.M. Bench-Capon, P.E. Dunne, Argumentation in artificial intelligence, Artif. Intell. 171 (2007) 619–641.

[2] A. Hunter, Towards a framework for computational persuasion with applications in behaviour change, Argument Comput. 9 (1) (2018) 15–40.

[3] S. Modgil, F. Toni, F. Bex, I. Bratko, C.I. Chesñevar, W. Dvorák, M.A. Falappa, X. Fan, S.A. Gaggl, A.J. Garcia, M.P. González, T.F. Gordon, J. Leite, M. Mouzina, C. Reed, G.R. Simari, S. Szeider, P. Torroni, S. Woltran, Agreement Technologies, in: Ch. 21: The added value of argumentation: examples and challenges, in: Law, Governance and Technology, vol. 8, Springer, 2013, pp. 357–404.

[4] I. Rahwan, G.R. Simari, Argumentation in Artificial Intelligence, Springer, 2009.

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

[6] K. Atkinson, T.J.M. Bench-Capon, Argumentation schemes in AI and law, Argument Comput. 12 (3) (2021) 417–434.

[7] L. Amgoud, H. Prade, Using arguments for making and explaining decisions, Artif. Intell. 173 (3–4) (2009) 413–436.

[8] T.J.M. Bench-Capon, K. Atkinson, A.Z. Wyner, Using argumentation to structure e-participation in policy making, Trans. Large-Scale Data Knowl.- Centered Syst. 18 (2015) 1–29.

[9] M. Snaith, R.Ø. Nielsen, S.R. Kotnis, A. Pease, Ethical challenges in argumentation and dialogue in a healthcare context, Argument Comput. 12 (2) (2021) 249–264.
[10] N. Kökciyan, I. Sassoon, E. Sklar, S. Modgil, S. Parsons, Applying metalevel argumentation frameworks to support medical decision making, IEEE Intell. Syst. 36 (2) (2021) 64–71.
[11] A. Pazienza, D. Grossi, F. Grasso, R. Palmieri, M. Zito, S. Ferilli, An abstract argumentation approach for the prediction of analysts' recommendations following earnings conference calls, Intell. Artif. 13 (2) (2019) 173–188.
[12] M.E.B. Brarda, L.H. Tamargo, A.J. García, Using argumentation to obtain and explain results in a decision support system, IEEE Intell. Syst. 36 (2) (2021) 36–42.
[13] G. Alfano, M. Calautti, S. Greco, F. Parisi, I. Trubitsyna, Explainable acceptance in probabilistic abstract argumentation: complexity and approximation, in: Proc. of KR, 2020, pp. 33–43.
[14] G. Alfano, M. Calautti, S. Greco, F. Parisi, I. Trubitsyna, Explainable acceptance in probabilistic and incomplete abstract argumentation frameworks, Artif. Intell. 323 (2023) 103967.
[15] N. Kökciyan, N. Yaglikci, P. Yolum, An argumentation approach for resolving privacy disputes in online social networks, ACM Trans. Internet Technol. 17 (3) (2017) 27:1–27:22.
[16] P.M. Dung, On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games, Artif. Intell. 77 (2) (1995) 321–358.
[17] W. Dvorák, P.E. Dunne, Computational problems in formal argumentation and their complexity, in: Handbook of Formal Argumentation (HOFA), vol. 1, College Publications, 2018, Ch. 13.
[18] C. Cayrol, M.-C. Lagasquie-Schiex, On the acceptability of arguments in bipolar argumentation frameworks, in: Proc. of ECSQARU, 2005, pp. 378–389.
[19] G. Boella, D.M. Gabbay, L.W.N. van der Torre, S. Villata, Support in abstract argumentation, in: Proc. of COMMA, 2010, pp. 111–122.
[20] P. Baroni, F. Cerutti, M. Giacomin, G. Guida, AFRA: argumentation framework with recursive attacks, Int. J. Approx. Reason. 52 (1) (2011) 19–37.
[21] S. Polberg, A. Hunter, Empirical evaluation of abstract argumentation: supporting the need for bipolar and probabilistic approaches, Int. J. Approx. Reason. 93 (2018) 487–543.
[22] R. Walton, C. Gierl, H. Mistry, M. Vessey, J. Fox, Evaluation of computer support for prescribing (capsule) using simulated cases, Br. Med. J. 315 (1997) 791–795.
[23] A.S. Coulson, D.W. Glasspool, J. Fox, J. Emery, RAGs: a novel approach to computerized genetic risk assessment and decision support from pedigrees, Methods Inf. Med. 40 (4) (2001) 315–322.
[24] S. Gottifredi, L.H. Tamargo, A.J. Garcia, G.R. Simari, Arguing about informant credibility in open multi-agent systems, Artif. Intell. 259 (2018) 91–109.
[25] C. Cayrol, L.F. del Cerro, M.-C. Lagasquie-Schiex, A logical vision of abstract argumentation systems with bipolar and recursive interactions, Research report, Institut de Recherche en Informatique de Toulouse (IRIT), 2016.
[26] C. Cayrol, A. Cohen, M.-C. Lagasquie-Schiex, Higher-order interactions (bipolar or not) in abstract argumentation: a state of the art, in: Handbook of Formal Argumentation (HOFA), vol. 2, College Publications, 2021, Ch. 1.
[27] S. Gottifredi, A. Cohen, A.J. Garcia, G.R. Simari, Characterizing acceptability semantics of argumentation frameworks with recursive attack and support relations, Artif. Intell. 262 (2018) 336–368.
[28] C. Cayrol, J. Fandinno, L.F. del Cerro, M. Lagasquie-Schiex, Structure-based semantics of argumentation frameworks with higher-order attacks and supports, in: Proc. of COMMA, 2018, pp. 29–36.
[29] F. Nouioua, V. Risch, Argumentation frameworks with necessities, in: Proc. of SUM, 2011, pp. 163–176.
[30] G. Alfano, S. Greco, F. Parisi, G.I. Simari, G.R. Simari, An incremental approach to structured argumentation over dynamic knowledge bases, in: Proc. of KR, 2018, pp. 78–87.
[31] G. Alfano, S. Greco, F. Parisi, G.I. Simari, G.R. Simari, Incremental computation for structured argumentation over dynamic delp knowledge bases, Artif. Intell. 300 (2021) 103553.
[32] P. Baroni, G. Boella, F. Cerutti, M. Giacomin, L.W.N. van der Torre, S. Villata, On the input/output behavior of argumentation frameworks, Artif. Intell. 217 (2014) 144–197.
[33] P. Baroni, M. Giacomin, B. Liao, On topology-related properties of abstract argumentation semantics. a correction and extension to dynamics of argumentation systems: a division-based method, Artif. Intell. 212 (2014) 104–115.
[34] R. Baumann, Splitting an argumentation framework, in: Proc. of LPNMR, 2011, pp. 40–53.
[35] P. Bisquert, C. Cayrol, F.D. de Saint-Cyr, M.-C. Lagasquie-Schiex, Characterizing change in abstract argumentation systems, in: Trends in Belief Revision and Argumentation Dynamics, vol. 48, 2013, pp. 75–102.
[36] G. Charwat, W. Dvorák, S.A. Gaggl, J.P. Wallner, S. Woltran, Methods for solving reasoning problems in abstract argumentation - a survey, Artif. Intell. 220 (2015) 28–63.
[37] M.A. Falappa, A.J. Garcia, G. Kern-Isberner, G.R. Simari, On the evolving relation between belief revision and argumentation, Knowl. Eng. Rev. 26 (1) (2011) 35–43.
[38] B. Liao, L. Jin, R.C. Koons, Dynamics of argumentation systems: a division-based method, Artif. Intell. 175 (11) (2011) 1790–1814.
[39] G. Alfano, S. Greco, F. Parisi, Incremental computation in dynamic argumentation frameworks, IEEE Intell. Syst. 36 (6) (2021) 80–86.
[40] G. Alfano, S. Greco, Incremental skeptical preferred acceptance in dynamic argumentation frameworks, IEEE Intell. Syst. 36 (2) (2021) 6–12.
[41] S. Bistarelli, F. Faloci, F. Santini, C. Taticchi, Studying dynamics in argumentation with Rob, in: Proc. of COMMA, 2018, pp. 451–452.
[42] S. Doutre, J. Mailly, Constraints and changes: a survey of abstract argumentation dynamics, Argument Comput. 9 (3) (2018) 223–248.
[43] S. Greco, F. Parisi, Incremental computation of deterministic extensions for dynamic argumentation frameworks, in: Proc. of JELIA, 2016, pp. 288–304.
[44] G. Alfano, S. Greco, F. Parisi, A meta-argumentation approach for the efficient computation of stable and preferred extensions in dynamic bipolar argumentation frameworks, Intell. Artif. 12 (2) (2018) 193–211.
[45] G. Alfano, S. Greco, F. Parisi, Computing extensions of dynamic abstract argumentation frameworks with second-order attacks, in: IDEAS, 2018, pp. 183–192.
[46] A. Niskanen, M. Järvisalo, μ-toksia: an efficient abstract argumentation reasoner, in: Proc. of 17th International Conference on Principles of Knowledge Representation and Reasoning (KR), 2020, pp. 800–804.
[47] J. Lagniez, E. Lonca, J. Mailly, J. Rossit, Design and results of ICCMA 2021, CoRR, arXiv:2109.08884 [abs].
[48] M. Thimm, F. Cerutti, M. Vallati, FUDGE: a light-weight solver for abstract argumentation based on SAT reductions, Fourth ICCMA, 2021.
[49] G. Alfano, S. Greco, F. Parisi, Efficient computation of extensions for dynamic abstract argumentation frameworks: an incremental approach, in: Proc. of IJCAI, 2017, pp. 49–55.
[50] G. Alfano, S. Greco, F. Parisi, An efficient algorithm for skeptical preferred acceptance in dynamic argumentation frameworks, in: Proc. of IJCAI, 2019, pp. 18–24.
[51] S. Villata, G. Boella, D.M. Gabbay, L.W.N. van der Torre, Modelling defeasible and prioritized support in bipolar argumentation, Ann. Math. Artif. Intell. 66 (1–4) (2012) 163–197.
[52] I. Boudhar, F. Nouioua, V. Risch, Handling preferences in argumentation frameworks with necessities, in: Proceedings of the 4th International Conference on Agents and Artificial Intelligence ICAART, 2012, pp. 340–345.
[53] F. Nouioua, S. Boutouhami, Argumentation frameworks with necessities and their relationship with logic programs, Argument Comput. 14 (1) (2023) 17–58.

[54] G. Alfano, S. Greco, F. Parisi, I. Trubitsyna, On the semantics of abstract argumentation frameworks: a logic programming approach, Theory Pract. Log. Program. 20 (5) (2020) 703–718.
[55] A. Cohen, S. Gottifredi, A.J. Garcia, G.R. Simari, A survey of different approaches to support in argumentation systems, Knowl. Eng. Rev. 29 (2014) 513–550.
[56] C. Cayrol, J. Fandinno, L.F. del Cerro, M. Lagasquie-Schiex, Valid attacks in argumentation frameworks with recursive attacks, in: Proc. of COMMONSENSE, 2017.
[57] M. Lagasquie-Schiex, Handling support cycles and collective interactions in the logical encoding of higher-order bipolar argumentation frameworks, in: Proceedings of the 4th International Conference on Logic and Argumentation, in: Lecture Notes in Computer Science, vol. 13040, 2021, pp. 244–265.
[58] M. Lagasquie-Schiex, Handling support cycles and collective interactions in the logical encoding of higher-order bipolar argumentation frameworks, J. Log. Comput. 33 (2) (2023) 289–318.
[59] A. Cohen, S. Gottifredi, A.J. Garcia, G.R. Simari, An approach to abstract argumentation with recursive attack and support, J. Appl. Log. 13 (4) (2015) 509–533.
[60] P. Baroni, M. Caminada, M. Giacomin, An introduction to argumentation semantics, Knowl. Eng. Rev. 26 (4) (2011) 365–410.
[61] S. Doutre, M. Lafages, M. Lagasquie-Schiex, Argumentation frameworks with higher-order attacks: labellings and complexity, in: Proc. of International Conference on Tools with Artificial Intelligence (ICTAI), IEEE, 2020, pp. 1210–1217.
[62] S. Bistarelli, L. Kotthoff, F. Santini, C. Taticchi, Summary report for the third international competition on computational models of argumentation, AI Mag. 42 (3) (2021) 70–73.
[63] M. Järvisalo, T. Lehtonen, A. Niskanen, Solver and benchmark descriptions of ICCMA 2023: 5th international competition on computational models of argumentation, 2023.
[64] M. Järvisalo, T. Lehtonen, A. Niskanen, Design of ICCMA 2023, 5th international competition on computational models of argumentation: a preliminary report (invited paper), in: Proceedings of the First International Workshop on Argumentation and Applications (Arg&App), vol. 3472, 2023, pp. 4–10.
[65] W. Dvorák, P.E. Dunne, Computational problems in formal argumentation and their complexity, FLAP 4 (8).
[66] G. Alfano, S. Greco, F. Parisi, I. Trubitsyna, On preferences and priority rules in abstract argumentation, in: Proc. of IJCAI, 2022, pp. 2517–2524.
[67] G. Alfano, S. Greco, F. Parisi, I. Trubitsyna, Preferences and constraints in abstract argumentation, in: Proc. of IJCAI, 2023, pp. 3095–3103.
[68] Gerhard, S. Ellmauthaler, H. Strass, J.P. Wallner, S. Woltran, Abstract dialectical frameworks. An overview, FLAP 4 (8).
[69] M. Thimm, S. Villata, The first international competition on computational models of argumentation: results and analysis, Artif. Intell. 252 (2017) 267–294.
[70] B. Berger, P.W. Shor, Approximation algorithms for the maximum acyclic subgraph problem, in: Proceedings of the First Annual ACM-SIAM Symposium on Discrete Algorithms, 22-24 January 1990, San Francisco, California, USA, SIAM, 1990, pp. 236–243.
[71] R. Hassin, S. Rubinstein, Approximations for the maximum acyclic subgraph problem, Inf. Process. Lett. 51 (3) (1994) 133–140.
[72] A. Niskanen, M. Järvisalo, μ-toksia participating in ICCMA 2019, Third ICCMA, 2019.
[73] O. Cocarascu, E. Cabrio, S. Villata, F. Toni, Dataset independent baselines for relation prediction in argument mining, in: Proc. of COMMA, 2020, pp. 45–52.
[74] E. Cabrio, S. Villata, Node: a benchmark of natural language arguments, in: Proc. of COMMA, 2014, pp. 449–450.
[75] H. Hofmann, H. Wickham, K. Kafadar, Letter-value plots: boxplots for large data, J. Comput. Graph. Stat. 26 (3) (2017) 469–477.
[76] S. Coste-Marquis, C. Devred, P. Marquis, Constrained argumentation frameworks, in: Proc. of KR, 2006, pp. 112–122.
[77] O. Arieli, Conflict-free and conflict-tolerant semantics for constrained argumentation frameworks, J. Appl. Log. 13 (4) (2015) 582–604.
[78] G. Alfano, S. Greco, F. Parisi, I. Trubitsyna, Argumentation frameworks with strong and weak constraints: semantics and complexity, in: Proc. of AAAI, 2021, pp. 6175–6184.
[79] L. Amgoud, C. Cayrol, On the acceptability of arguments in preference-based argumentation, in: Proc. of UAI, 1998, pp. 1–7.
[80] S. Modgil, Reasoning about preferences in argumentation frameworks, Artif. Intell. 173 (9–10) (2009) 901–934.
[81] L. Amgoud, S. Vesic, Rich preference-based argumentation frameworks, Int. J. Approx. Reason. 55 (2) (2014) 585–606.
[82] S. Kaci, L.W.N. van der Torre, S. Villata, Preference in abstract argumentation, in: Proc. of Computational Models of Argument (COMMA), 2018, pp. 405–412.
[83] L. Amgoud, C. Cayrol, A reasoning model based on the production of acceptable arguments, Ann. Math. Artif. Intell. 34 (1–3) (2002) 197–215.
[84] P. Baroni, M. Giacomin, B. Liao, Locality and modularity in abstract argumentation, in: Handbook of Formal Argumentation, vol. 1, College Publications, 2018, Ch. 19.
[85] G. Alfano, S. Greco, F. Parisi, G.I. Simari, G.R. Simari, On the incremental computation of semantics in dynamic argumentation, FLAP 8 (6) (2021) 1749–1792.
[86] G. Boella, S. Kaci, L.W.N. van der Torre, Dynamics in argumentation with single extensions: attack refinement and the grounded extension (extended version), in: Proc. of ArgMAS, 2009.
[87] G. Boella, S. Kaci, L.W.N. van der Torre, Dynamics in argumentation with single extensions: abstraction principles and the grounded extension, in: Proc. of ECSQARU, 2009, pp. 107–118.
[88] C. Cayrol, F.D. de Saint-Cyr, M.-C. Lagasquie-Schiex, Revision of an argumentation system, in: Proc. of KR, 2008, pp. 124–134.
[89] C. Cayrol, F.D. de Saint-Cyr, M.-C. Lagasquie-Schiex, Change in abstract argumentation frameworks: adding an argument, J. Artif. Intell. Res. 38 (2010) 49–84.
[90] P. Bisquert, C. Cayrol, F.D. de Saint-Cyr, M. Lagasquie-Schiex, Change in argumentation systems: exploring the interest of removing an argument, in: Proc. of SUM, 2011, pp. 275–288.
[91] P. Bisquert, C. Cayrol, F.D. de Saint-Cyr, M. Lagasquie-Schiex, Changement dans un système d'argumentation: suppression d'un argument, Rev. Intell. Artif. 26 (3) (2012) 225–253.
[92] L. Amgoud, S. Vesic, Revising option status in argument-based decision systems, J. Log. Comput. 22 (5) (2012) 1019–1058.
[93] B. Liao, H. Huang, Partial semantics of argumentation: basic properties and empirical results, J. Log. Comput. 23 (3) (2013) 541–562.
[94] B. Liao, Toward incremental computation of argumentation semantics: a decomposition-based approach, Ann. Math. Artif. Intell. 67 (3–4) (2013) 319–358.
[95] Y. Xu, C. Cayrol, The matrix approach for abstract argumentation frameworks, in: Proc. of TAFA, 2015, pp. 243–259.
[96] R. Baumann, G. Brewka, Expanding argumentation frameworks: enforcing and monotonicity results, in: Proc. of COMMA, 2010, pp. 75–86.
[97] E. Oikarinen, S. Woltran, Characterizing strong equivalence for argumentation frameworks, Artif. Intell. 175 (14–15) (2011) 1985–2009.
[98] R. Baumann, Normal and strong expansion equivalence for argumentation frameworks, Artif. Intell. 193 (2012) 18–44.
[99] R. Baumann, Context-free and context-sensitive kernels: update and deletion equivalence in abstract argumentation, in: Proc. of ECAI, 2014, pp. 63–68.
[100] R. Baumann, W. Dvorák, T. Linsbichler, S. Woltran, A general notion of equivalence for abstract argumentation, Artif. Intell. 275 (2019) 379–410.
[101] L. Amgoud, C. Cayrol, M.-C. Lagasquie-Schiex, On the bipolarity in argumentation frameworks, in: Proc. of NMR, 2004, pp. 1–9.
[102] A. Cohen, S. Gottifredi, A.J. Garcia, G.R. Simari, A survey of different approaches to support in argumentation systems, Knowl. Eng. Rev. 29 (5) (2014) 513–550.
[103] C. Cayrol, M.-C. Lagasquie-Schiex, On the acceptability of arguments in bipolar argumentation frameworks, in: 8th European Conference on Symbolic and Quantitative Approaches to Reasoning with Uncertainty, Barcelona, Spain, in: Lecture Notes in Artificial Intelligence, vol. 3571, Springer, Germany, 2005, pp. 378–389.
[104] N. Oren, T.J. Norman, Semantics for evidence-based argumentation, in: P. Besnard, S. Doutre, A. Hunter (Eds.), 2nd International Conference on Computational Models of Argument, Toulouse, France, Frontiers in Artificial Intelligence and Applications, IOS Press, Amsterdam, NL, 2008, pp. 276–284.
[105] G. Boella, D.M. Gabbay, L.W.N. van der Torre, S. Villata, Support in abstract argumentation, in: COMMA, 2010, pp. 111–122.
[106] F. Nouioua, V. Risch, Bipolar argumentation frameworks with specialized supports, in: 22th IEEE International Conference on Tools with Artificial Intelligence, Arras, France, IEEE Press, Washington DC, USA, 2010, pp. 215–218.
[107] F. Nouioua, Afs with necessities: further semantics and labelling characterization, in: Proc. of Int. Conf. on Scalable Uncertainty Management (SUM), 2013, pp. 120–133.

[108] C. Cayrol, M.-C. Lagasquie-Schiex, Change in abstract bipolar argumentation systems, in: Proc. of SUM, 2015, pp. 314–329.
[109] G. Alfano, S. Greco, F. Parisi, Computing stable and preferred extensions of dynamic bipolar argumentation frameworks, in: Proc. of AI^3 Workshop, 2017, pp. 28–42.
[110] P. Baroni, F. Cerutti, M. Giacomin, G. Guida, Encompassing attacks to attacks in abstract argumentation frameworks, in: Proc. of ECSQARU, 2009, pp. 83–94.
[111] G. Alfano, A. Cohen, S. Gottifredi, S. Greco, F. Parisi, G.R. Simari, Dynamics in abstract argumentation frameworks with recursive attack and support relations, in: Proc. of ECAI, 2020, pp. 577–584.
[112] J. Dauphin, M. Cramer, Extended explanatory argumentation frameworks, in: Proc. of 4th International Workshop on Theory and Applications of Formal Argumentation (TAFA), in: Lecture Notes in Computer Science, vol. 10757, 2017, pp. 86–101.
[113] D. Saccà, C. Zaniolo, Stable models and non-determinism in logic programs with negation, in: PODS, 1990, pp. 205–217.
[114] P.M. Dung, Negations as hypotheses: an abductive foundation for logic programming, in: ICLP, 1991, pp. 3–17.
[115] M. Caminada, S. Sá, J.F.L. Alcântara, W. Dvorák, On the equivalence between logic programming semantics and argumentation semantics, Int. J. Approx. Reason. 58 (2015) 87–111.
[116] T. Rienstra, C. Sakama, L. van der Torre, B. Liao, A principle-based robustness analysis of admissibility-based argumentation semantics, Argument Comput. 11 (3) (2020) 305–339.