24th European Conference on Artificial Intelligence - ECAI 2020
Santiago de Compostela, Spain

Dynamics in Abstract Argumentation Frameworks with Recursive Attack and Support Relations

Gianvincenzo Alfano [1] and **Andrea Cohen** [2] and **Sebastian Gottifredi** [2] and **Sergio Greco** [1] and **Francesco Parisi** [1] and **Guillermo Simari** [2]

Abstract. Argumentation is an important topic in the field of AI. There is a substantial amount of work about different aspects of Dung's abstract Argumentation Framework (AF). Two relevant aspects considered separately so far are extending the framework to account for recursive attacks and supports, and considering dynamics, *i.e.*, AFs evolving over time. In this paper, we jointly deal with these two aspects. We focus on Attack-Support Argumentation Frameworks (ASAFs) which allow for attack and support relations not only between arguments but also targeting attacks and supports at any level, and propose an approach for the incremental computation of extensions (sets of accepted arguments, attacks and supports) of updated ASAFs. Our approach assumes that an initial ASAF extension is given and uses it for first checking whether updates are irrelevant; for relevant updates, an extension of an updated ASAF is computed by translating the problem to the AF domain and leveraging on AF solvers. We experimentally show our incremental approach outperforms the direct computation of extensions for updated ASAFs.

1 Introduction

Argumentation has become an attractive and efficient paradigm for knowledge representation and reasoning within the field of Artificial Intelligence [17, 39, 43, 47]. In particular, Dung's abstract Argumentation Framework (AF) [32] provides a simple yet powerful formalism for modelling and reasoning with information expressed in terms of arguments and their conflicts. Following the work by Dung, there have been many extensions of AFs allowing for bipolar interactions [26], second-order attacks [20] or, more generally, recursive attacks [10]. In addition to these approaches extending Dung's AFs, the Attack-Support Argumentation Framework (ASAF) [37] allows for attacks and supports not only towards arguments but also targeting the attack and support relations at any level. This framework has the advantage of enabling a straightforward representation of reasoning situations (*e.g.* in the area of modelling decision processes), which are not easily accommodated within other frameworks such as Dung's AF or flat bipolar AFs. In particular, the support in ASAF is interpreted as *necessity* [45]: if argument $\mathcal{A}$ supports argument $\mathcal{B}$, then the acceptance of $\mathcal{A}$ is necessary to get the acceptance of $\mathcal{B}$; equivalently, the acceptance of $\mathcal{B}$ implies the acceptance of $\mathcal{A}$. However, a support can also target an attack or a support.

An ASAF can be represented by a graph as that in Figure $1(a)$, where arguments are denoted with calligraphic letters, attacks are denoted with single arrows ($\rightarrow$), and supports are denoted with double arrows ($\Rightarrow$). In particular, the ASAF in Figure $1(a)$ models the following scenario. 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 ($\mathcal{D}$) or make use of public transportation ($\mathcal{PT}$). Arguments $\mathcal{D}$ and $\mathcal{PT}$ 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 ($\mathcal{P}$). This is encoded by the attack α_4 from $\mathcal{P}$ 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 ($\mathcal{ST}$), but this is only required during winter, the current season in Bariloche ($\mathcal{W}$). Hence, $\mathcal{ST}$ supports $\mathcal{D}$ (support β_1) as the acceptance of $\mathcal{ST}$ is necessary for the acceptance of $\mathcal{D}$, and $\mathcal{W}$ supports β_1. In addition, rental car services ran out of such devices ($\mathcal{RN}$). In this context, John will end up deciding to use public transportation in Bariloche.

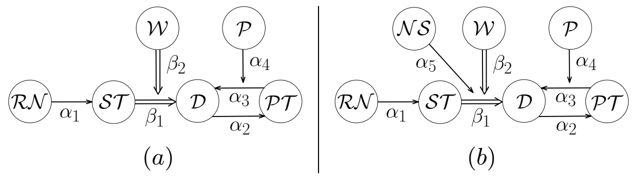

Figure 1: (a) Initial ASAF Δ_1 and (b) Updated ASAF

However, in practice, argumentation frameworks can be dynamic systems [4, 7, 12, 13, 18, 29, 35, 42]. In fact, typically an ASAF represents a temporary situation, and new arguments, attacks and supports can be added/removed to take into account new available knowledge. For instance, in 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 his stay, represented by an argument $\mathcal{NS}$. Then, when considering the previous arguments and attacks together with the new argument, a new conflict arises, needing to update John's knowledge: argument $\mathcal{NS}$ attacks β_1, the support from $\mathcal{ST}$ to $\mathcal{D}$, 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 $\mathcal{NS}$ and then, of the attack α_5, corresponds to the updated ASAF shown in Figure $1(b)$, according to which 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 Dung AFs [1, 3, 12, 19, 31, 38], Bipolar AFs and AFs with second order attacks [2]. Notwithstanding this, none of the developments regarding dynamics

[1] DIMES, University of Calabria, Italy, email: {g.alfano, greco, fparisi}@dimes.unical.it
[2] DCIC, Universidad Nacional del Sur, ICIC (CONICET-UNS), Argentina, email: {ac, sg, grs}@cs.uns.edu.ar

in argumentation has so far considered bipolar recursive frameworks like the ASAF. However, 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, the acceptance status of $\mathcal{RN}$ and $\mathcal{ST}$, as well as that of α_1, does not change after adding the attack α_5 from $\mathcal{NS}$ to the support from $\mathcal{ST}$ to $\mathcal{D}$.

With the aim of exploiting features like those above, in this paper we propose an incremental approach for efficiently computing extensions of an ASAF after performing an update. Specifically, we propose a technique addressing the following problem: given an ASAF Δ, an initial extension E_0 of Δ, and an update u consisting of the addition/removal of an attack/support, determine an extension E of the updated ASAF $u(\Delta)$.

Contributions. Our main contributions are as follows:

1. We identify and formally characterize *irrelevant updates* for an ASAF, for which an extension E of an updated ASAF $u(\Delta)$ can be directly obtained without requiring its overall computation. These results can be used to avoid wasted effort for any incremental algorithm, not just that proposed in this paper.
2. We characterize an ASAF in terms of an AF, and formally show that this AF yields *equivalent extensions* to those of the ASAF. Note that this AF can be used to compute ASAFs's extensions even in the static case, where updates are not considered.
3. We define an incremental algorithm for computing an extension E of an updated ASAF $u(\Delta)$, accounting for early termination conditions implied by irrelevant updates and leveraging on the incremental technique proposed in [1] for the computation on the (extensions-equivalent) AF for an updated ASAF $u(\Delta)$. The technique is able to incorporate any existing AF-solver to perform the incremental computation of ASAFs's extensions.
4. We perform an experimental analysis showing that our incremental approach outperforms the computation from scratch, where the fastest solvers from the International Competition on Computational Models of Argumentation (ICCMA) [3] are used as baselines.

As a side contribution, we develop a benchmark for testing ASAF solvers, and we briefly introduce labellings for the ASAF.

2 Essential Background

We briefly review the essential background for abstract Argumentation Frameworks (AFs) [32] and Attack-Support Argumentation Frameworks (ASAFs) (for a full presentation see [37]).

An AF consists of a set of arguments whose origin is left unspecified, and a set of conflicts between them [32].

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 [32] some semantic notions are defined leading to the characterization of collectively acceptable sets of arguments. Given an AF $\langle \mathsf{A}, \mathsf{R}\rangle$ and a set $\mathbf{S} \subseteq \mathsf{A}$ of arguments, we say that:

- $\mathbf{S}$ is *conflict-free* iff $\nexists \mathcal{A}, \mathcal{B} \in \mathbf{S}$ s.t. $(\mathcal{A}, \mathcal{B}) \in \mathsf{R}$.
- $\mathcal{A} \in \mathsf{A}$ is *acceptable w.r.t.* $\mathbf{S}$ iff $\forall \mathcal{B} \in \mathsf{A}$ s.t. $(\mathcal{B}, \mathcal{A}) \in \mathsf{R}$, $\exists \mathcal{C} \in \mathbf{S}$ s.t. $(\mathcal{C}, \mathcal{B}) \in \mathsf{R}$.
- $\mathbf{S}$ is *admissible* iff it is conflict-free and $\forall \mathcal{A} \in \mathbf{S}$, $\mathcal{A}$ is acceptable w.r.t. $\mathbf{S}$.

[3] http://argumentationcompetition.org

Then, by adding restrictions to the notion of admissibility, the complete (co), preferred (pr), stable (st), and grounded (gr) extensions of an AF are defined as follows. Given an AF $\langle \mathsf{A}, \mathsf{R}\rangle$ and a set of arguments $\mathbf{S} \subseteq \mathsf{A}$, we say that:

- $\mathbf{S}$ is a *complete extension* of AF iff it is admissible and $\forall \mathcal{A} \in \mathsf{A}$, if $\mathcal{A}$ is acceptable w.r.t. $\mathbf{S}$, then $\mathcal{A} \in \mathbf{S}$.
- $\mathbf{S}$ is a *preferred extension* of AF iff it is a maximal (w.r.t. $\subseteq$) complete extension of AF.
- $\mathbf{S}$ is a *stable extension* of AF iff it is a complete extension of AF and $\forall \mathcal{A} \in \mathsf{A}\backslash\mathbf{S}$, $\exists \mathcal{B} \in \mathbf{S}$ s.t. $(\mathcal{B}, \mathcal{A}) \in \mathsf{R}$.
- $\mathbf{S}$ is the *grounded extension* of AF iff it is the smallest (w.r.t. $\subseteq$) complete extension of AF.

Attack-Support Argumentation Framework. The ASAF extends Dung's AF by incorporating bipolar higher-order interactions. In that way, the ASAF 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 support relation of the ASAF is interpreted as *necessity* [45]. That is, the necessary support relation in the ASAF imposes the following acceptability constraints on the elements it relates: the acceptance of $\mathcal{B}$ implies the acceptance of $\mathcal{A}$ or, equivalently, the non-acceptance of $\mathcal{A}$ implies the non-acceptance of $\mathcal{B}$.

Definition 2 (*ASAF*). *An* Attack-Support Argumentation Framework *(ASAF) is a tuple* $\langle \mathbb{A}, \mathbb{R}, \mathbb{S}\rangle$ *where* $\mathbb{A}$ *is a set of arguments,* $\mathbb{R} \subseteq W$ *is the attack relation, and* $\mathbb{S} \subseteq W$ *is the support relation, where* W *is the set iteratively defined as follows:* $W = \mathbb{A} \times \mathbb{A}$ *(basic step) and* $W = \mathbb{A} \times W$ *(iterative step). It is assumed that* $\mathbb{S}$ *is acyclic and* $\mathbb{R} \cap \mathbb{S} = \emptyset$.

As stated before, attacks and supports in an ASAF can also be attacked and supported. To simplify the notation, an attack $(\mathcal{A}, \mathcal{B}) \in \mathbb{R}$ will be denoted as $\alpha_1 = (\mathcal{A}, \mathcal{B})$; similarly, a support $(\mathcal{B}, \mathcal{C}) \in \mathbb{S}$ will be denoted as $\beta_1 = (\mathcal{B}, \mathcal{C})$. Then, for instance, an attack from $\mathcal{D}$ to α_1 will be denoted as $\alpha_2 = (\mathcal{D}, \alpha_1)$. Moreover, given an attack $\alpha = (\mathcal{A}, X) \in \mathbb{R}$, $\mathcal{A}$ is called the source of α, denoted $\mathsf{src}(\alpha) = \mathcal{A}$, and X is called the target of α, denoted $\mathtt{trg}(\alpha) = X$. Analogously, given a support $\beta = (\mathcal{B}, Y) \in \mathbb{S}$, $\mathcal{B}$ is called the source of β, denoted $\mathsf{src}(\beta) = \mathcal{B}$, and Y is called the target of β, denoted $\mathtt{trg}(\beta) = Y$.

An ASAF Δ can be represented following a graph-like notation by G_Δ: an argument $\mathcal{A} \in \mathbb{A}$ will be a node in G_Δ, an attack $\alpha = (\mathcal{A}, X) \in \mathbb{R}$ will be an edge $\mathcal{A} \xrightarrow{\alpha} X$ in G_Δ, and a support $\beta = (\mathcal{B}, Y) \in \mathbb{S}$ will be an edge $\mathcal{B} \overset{\beta}{\Longrightarrow} Y$ in G_Δ. Attacks and supports whose target is an argument are said to be *first-level* interactions, while attacks and supports whose target is an interaction of level i are said to be interactions of level $i+1$.

Example 1. *The initial example from the introduction can be represented by the ASAF* $\Delta_1 = \langle \mathbb{A}_1, \mathbb{R}_1, \mathbb{S}_1\rangle$ *in Figure 1(a), with* $\mathbb{A}_1 = \{\mathcal{D}, \mathcal{PT}, \mathcal{P}, \mathcal{ST}, \mathcal{W}, \mathcal{RN}\}$*,* $\mathbb{R}_1 = \{\alpha_1, \alpha_2, \alpha_3, \alpha_4\}$*, and* $\mathbb{S}_1 = \{\beta_1, \beta_2\}$*; where* $\alpha_1 = (\mathcal{RN}, \mathcal{ST})$*,* $\alpha_2 = (\mathcal{D}, \mathcal{PT})$ *and* $\alpha_3 = (\mathcal{PT}, \mathcal{D})$ *are first-level attacks,* $\beta_1 = (\mathcal{ST}, \mathcal{D})$ *is a first-level support, and the second-level interactions are the attack* $\alpha_4 = (\mathcal{P}, \alpha_3)$ *and the support* $\beta_2 = (\mathcal{W}, \beta_1)$.

In standard graphs, paths are defined among nodes. Here, for graphs denoting ASAFs, we consider paths whose target can also be an edge. In particular, we define a support path from $\mathcal{A}$ to X as a path $\mathcal{A} = \mathcal{A}_1 \Longrightarrow \mathcal{A}_2 \Longrightarrow \ldots \Longrightarrow \mathcal{A}_n = X$, where each $\mathcal{A}_i$ $(1 \leq i < n)$ is an argument and $\mathcal{A}_n = X$ is an argument, attack or support, whose set of edges $\mathbf{S}$ contains the support links in the path;

a support path is empty iff $\mathcal{A} = X$. Then, the conflicts between the elements of an ASAF, referred to as *defeats*, are defined as follows.

Definition 3 (Defeats)**.** *Let* $\Delta = \langle \mathbb{A}, \mathbb{R}, \mathbb{S} \rangle$ *be an ASAF,* $\alpha \in \mathbb{R}$*,* $X \in (\mathbb{A} \cup \mathbb{R} \cup \mathbb{S})$ *and* $\mathbf{S} \subseteq \mathbb{S}$*. We say that* α defeats X (given $\mathbf{S}$)*, denoted* α def X given $\mathbf{S}$ *(or simply* α def X*, whenever* $\mathbf{S} = \emptyset$*) iff:*

- *there exists a (possibly empty) support path from* $\mathsf{trg}(\alpha)$ *to* X*, whose set of edges is* $\mathbf{S}$*; or*
- $X \in \mathbb{R}$ *and there exists a (possibly empty) support path from* $\mathsf{trg}(\alpha)$ *to* $\mathsf{src}(X)$*, whose set of edges is* $\mathbf{S}$*.*

The preceding definition not only accounts for defeats originated by the attack relation, but also accounts for defeats arising from the coexistence of the attack and support relations in the ASAF.

Example 2. *Given the ASAF* Δ_1 *from Example 1, these defeats occur:* α_1 def $\mathcal{ST}$*,* α_2 def $\mathcal{PT}$*,* α_2 def α_3*,* α_3 def $\mathcal{D}$*,* α_3 def α_2*,* α_4 def α_3*,* α_1 def $\mathcal{D}$ given $\{\beta_1\}$ *and* α_1 def α_2 given $\{\beta_1\}$*.*

We now define the sets of elements (arguments, attacks and supports) of an ASAF that can be collectively accepted under some criteria, called *extensions*. When not explicitly stated otherwise, attacks in an ASAF will be denoted with α, supports will be denoted with β (both, possibly with subscripts), and elements that can either be arguments, attacks or supports will be denoted with capital letters such as X or Y. Similarly, sets of elements from an ASAF will be denoted with capital boldface letters such as $\mathbf{S}$ or $\mathbf{S}'$.

The notion of conflict-freeness establishes the minimum requirements a set of arguments, attacks, and supports should satisfy in order to be collectively accepted.

Definition 4 (Conflict-freeness)**.** *Let* $\Delta = \langle \mathbb{A}, \mathbb{R}, \mathbb{S} \rangle$ *be an ASAF and* $\mathbf{S} \subseteq (\mathbb{A} \cup \mathbb{R} \cup \mathbb{S})$*. We say that* $\mathbf{S}$ is conflict-free *iff* $\nexists \alpha, X \in \mathbf{S}$*,* $\nexists \mathbf{S}' \subseteq \mathbf{S}$ *s.t.* α def X given $\mathbf{S}'$*.*

The notion of acceptability characterizes the defense by a set of arguments, attacks and supports against the defeats on its elements.

Definition 5 (ASAF Acceptability)**.** *Let* $\Delta = \langle \mathbb{A}, \mathbb{R}, \mathbb{S} \rangle$ *be an ASAF,* $X \in (\mathbb{A} \cup \mathbb{R} \cup \mathbb{S})$ *and* $\mathbf{S} \subseteq (\mathbb{A} \cup \mathbb{R} \cup \mathbb{S})$*. We say that* X is acceptable w.r.t. $\mathbf{S}$ *iff* $\forall \alpha \in \mathbb{R}$*,* $\forall \mathbf{T} \subseteq \mathbb{S}$ *s.t.* α def X given $\mathbf{T}$*:* $\exists Y \in (\{\alpha\} \cup \mathbf{T})$*,* $\exists \alpha' \in \mathbf{S}$*,* $\exists \mathbf{S}' \subseteq \mathbf{S}$ *s.t.* α' def Y given $\mathbf{S}'$*.*

Then, admissible sets are defined by combining the notions of conflict-freeness and acceptability.

Definition 6 (ASAF Admissibility)**.** *Let* $\Delta = \langle \mathbb{A}, \mathbb{R}, \mathbb{S} \rangle$ *be an ASAF and* $\mathbf{S} \subseteq (\mathbb{A} \cup \mathbb{R} \cup \mathbb{S})$*.* $\mathbf{S}$ is admissible *iff it is conflict-free and* $\forall X \in \mathbf{S}$*:* X *is acceptable w.r.t.* $\mathbf{S}$*.*

Example 3. *Given the ASAF* Δ_1 *from Example 1, for instance,* $\mathcal{D}$ *is not acceptable w.r.t.* $\{\alpha_4\}$ *because, even though* α_4 *defends it against the defeat by* α_3*, it does not defend it against the defeat by* α_1 *given* $\{\beta_1\}$*. In contrast,* $\mathcal{PT}$ *is acceptable w.r.t.* $\{\alpha_1, \beta_1\}$ *because this set defends it against the defeat by* α_2*. Consequently, for instance, the set* $\{\alpha_1, \beta_1, \mathcal{PT}\}$ *is admissible whereas the set* $\{\alpha_4, \mathcal{D}\}$ *is not.*

Finally, the complete (`co`), preferred (`pr`), stable (`st`), and grounded (`gr`) extensions are defined as follows.

Definition 7 (*ASAF* Extensions)**.** *Let* $\Delta = \langle \mathbb{A}, \mathbb{R}, \mathbb{S} \rangle$ *be an ASAF and* $\mathbf{S} \subseteq (\mathbb{A} \cup \mathbb{R} \cup \mathbb{S})$*:*

- $\mathbf{S}$ *is a* complete extension *of* Δ *iff it is admissible and* $\forall X \in (\mathbb{A} \cup \mathbb{R} \cup \mathbb{S})$*, if* X *is acceptable w.r.t.* $\mathbf{S}$*, then* $X \in \mathbf{S}$*.*
- $\mathbf{S}$ *is a* preferred extension *of* Δ *iff it is a maximal* (*w.r.t.* $\subseteq$) *complete extension of* Δ*.*
- $\mathbf{S}$ *is a* stable extension *of* Δ *iff it is a complete extension of* Δ *and* $\forall X \in (\mathbb{A} \cup \mathbb{R} \cup \mathbb{S}) \backslash \mathbf{S}$*,* $\exists\, \alpha \in \mathbf{S}$*,* $\exists\, \mathbf{S}' \subseteq \mathbf{S}$ *s.t.* α def X given $\mathbf{S}'$*.*
- $\mathbf{S}$ *is the* grounded extension *of* Δ *iff it is the smallest* (*w.r.t.* $\subseteq$) *complete extension of* Δ*.*

Example 4. *The grounded extension of the ASAF* Δ_1 *from Example 1 is* $\{\mathcal{RN}, \alpha_1, \beta_1, \mathcal{W}, \beta_2, \mathcal{PT}, \mathcal{P}, \alpha_4\}$ *(it is also the only preferred and stable extension, thus it is the unique complete extension).*

Similarly to the case of AFs (see [8] for an overview), extensions of an ASAF can also be expressed through labellings. A *labelling* for an ASAF $\Delta = \langle \mathbb{A}, \mathbb{R}, \mathbb{S} \rangle$ is a total function $\mathcal{L}ab : (\mathbb{A} \cup \mathbb{R} \cup \mathbb{S}) \mapsto \{\textsc{in}, \textsc{out}, \textsc{undec}\}$. Given a labelling $\mathcal{L}ab$, we define $\textsc{in}(\mathcal{L}ab) = \{X \mid \mathcal{L}ab(X) = \textsc{in}\}$, $\textsc{out}(\mathcal{L}ab) = \{X \mid \mathcal{L}ab(X) = \textsc{out}\}$, and $\textsc{undec}(\mathcal{L}ab) = \{X \mid \mathcal{L}ab(X) = \textsc{undec}\}$. Also, when convenient, a labelling $\mathcal{L}ab$ will be represented as the triple $(\textsc{in}(\mathcal{L}ab), \textsc{out}(\mathcal{L}ab), \textsc{undec}(\mathcal{L}ab))$.

Then, the *complete labellings* can be defined as follows. $\mathcal{L}ab$ is a *complete labelling* of an ASAF $\Delta = \langle \mathbb{A}, \mathbb{R}, \mathbb{S} \rangle$ iff for every $X \in (\mathbb{A} \cup \mathbb{R} \cup \mathbb{S})$ it holds that: (1) $\mathcal{L}ab(X) = \textsc{in}$ iff $\forall \alpha \in \mathbb{R}, \forall \mathbf{S} \subseteq \mathbb{S}$ s.t. α def X given $\mathbf{S}$, $\exists Y \in (\{\alpha\} \cup \mathbf{S})$ s.t. $\mathcal{L}ab(Y) = \textsc{out}$; and (2) $\mathcal{L}ab(X) = \textsc{out}$ iff $\exists \alpha \in \mathbb{R}, \exists \mathbf{S} \subseteq \mathbb{S}$ s.t. α def X given $\mathbf{S}$ and $\forall Y \in (\{\alpha\} \cup \mathbf{S})$, $\mathcal{L}ab(Y) = \textsc{in}$.

Hence, for X to be labelled as IN by a complete labelling of the ASAF we require that, for every set of elements originating a defeat on X, one of the elements in the set is labelled as OUT (*i.e.*, either the attack or one of the supports, if they exist). Analogously, for X to be labelled as OUT, we require that there exists a set of elements originating a defeat on X where every element in the set (*i.e.*, the attack and every support) is IN. Finally, if X is neither labelled as IN nor OUT, it is labelled as UNDEC.

As it holds for AFs (cf. [23]), a one-to-one correspondence between complete extensions and complete labellings of an ASAF can be established. Specifically, each complete extension E is in one-to-one correspondence with a complete labelling $L = (E, E^+, (\mathbb{A} \cup \mathbb{R} \cup \mathbb{S}) \backslash (E \cup E^+))$, where $E^+ = \{X \in (\mathbb{A} \cup \mathbb{R} \cup \mathbb{S}) \mid \exists \alpha \in E, \exists \mathbf{S} \subseteq E$ s.t. α def X given $\mathbf{S}\ \}$. In other words, the complete labelling L *corresponding* to a complete extension E of an ASAF is given by the triple $(\textsc{in}(L), \textsc{out}(L), \textsc{undec}(L))$, where $\textsc{in}(L) = E$, $\textsc{out}(L) = E^+$, and $\textsc{undec}(L) = (\mathbb{A} \cup \mathbb{R} \cup \mathbb{S}) \backslash (E \cup E^+))$.

Then, the preferred, stable and grounded labellings of an ASAF can be defined in terms of the complete labellings of the framework: $\mathcal{L}ab$ is a *preferred* (resp. *stable*, *grounded*) *labelling* of Δ iff it is a complete labelling s.t. $\textsc{in}(\mathcal{L}ab)$ is a preferred (resp. *stable*, *grounded*) extension of Δ.

Example 5. *Given the ASAF* Δ_1 *and its extensions listed in Example 4, the grounded labelling is* $(\{\mathcal{RN}, \alpha_1, \beta_1, \mathcal{W}, \beta_2, \mathcal{PT}, \mathcal{P}, \alpha_4\}, \{\mathcal{ST}, \mathcal{D}, \alpha_2, \alpha_3\}, \emptyset)$ *(it is also the only complete, preferred and stable labelling of* Δ_1*).*

3 Dynamics: Updates and Translation into AF

We start by defining the notion of *update* for an ASAF and identify updates that can be considered as *irrelevant*, since they do not require to compute the corresponding extension of the updated ASAF. Then, we formally characterize an ASAF in terms of an AF, whose extensions are shown to be equivalent to the extensions of the ASAF.

We start by defining the universal sets of attacks and supports, which account for every conceivable relationship that may appear in an ASAF. For a set of arguments $\mathbb{A}$, the set of *universal attacks* $\mathbb{R}_{\mathbb{U}}$ and the set of *universal supports* $\mathbb{S}_{\mathbb{U}}$ are such that $\mathbb{R}_{\mathbb{U}} \subseteq W$, $\mathbb{S}_{\mathbb{U}} \subseteq W$, and $\mathbb{R}_{\mathbb{U}} \cap \mathbb{S}_{\mathbb{U}} = \emptyset$ (where W is as defined in Definition 2).

An *update* consists of the addition (resp., removal) of an attack or a support not present (resp., present) in a given ASAF, as formalized in what follows.

Definition 8 (Update for ASAF)**.** *Let $\mathbb{A}$ be a set of arguments, $\mathbb{R}_{\mathbb{U}}$ and $\mathbb{S}_{\mathbb{U}}$ the universal sets of attacks and supports, and $\Delta = \langle \mathbb{A}, \mathbb{R}, \mathbb{S} \rangle$ an ASAF, where $\mathbb{R} \subseteq \mathbb{R}_{\mathbb{U}}$ and $\mathbb{S} \subseteq \mathbb{S}_{\mathbb{U}}$. An* update *$u$ over Δ belongs to any of the sets characterized below, yielding the updated ASAF $\Delta' = \langle \mathbb{A}, \mathbb{R}', \mathbb{S}' \rangle$:*

- $u \in \{+X \mid X \in (\mathbb{R}_{\mathbb{U}} \cup \mathbb{S}_{\mathbb{U}}) \backslash (\mathbb{R} \cup \mathbb{S}), \mathsf{src}(X) \in \mathbb{A}, \mathsf{trg}(X) \in (\mathbb{A} \cup \mathbb{R} \cup \mathbb{S})\}$. *If $X \in \mathbb{R}_{\mathbb{U}}$, then $\mathbb{R}' = \mathbb{R} \cup \{X\}$ and $\mathbb{S}' = \mathbb{S}$; if $X \in \mathbb{S}_{\mathbb{U}}$, then $\mathbb{R}' = \mathbb{R}$, $\mathbb{S}' = \mathbb{S} \cup \{X\}$ and $\mathbb{S}'$ is acyclic.*
- $u \in \{-X \mid X \in \mathbb{R} \cup \mathbb{S}, \nexists X' \in \mathbb{R} \cup \mathbb{S}$ *s.t.* $\mathsf{trg}(X') = X\}$. *Thus, $\mathbb{R}' = \mathbb{R} \backslash \{X\}$ and $\mathbb{S}' = \mathbb{S} \backslash \{X\}$.*

The ASAF Δ' obtained by applying an update u over an ASAF Δ will also be denoted as $u(\Delta)$. Furthermore, for simplicity, we write $\pm X$ for the addition or removal of an attack or a support $X = (\mathsf{src}(X), \mathsf{trg}(X))$. Then, for an update $u = +X$, the interaction X must not belong to the attack and support relations of the ASAF it will be applied on, and the source and target of X must belong to the ASAF; moreover, the support relation of the updated ASAF must remain acyclic. Contrastingly, for an update $u = -X$, the interaction X cannot be targeted by any other interaction in the ASAF. For instance, in the scenario described in the introduction, if we first extend the ASAF Δ_1 to also include argument $\mathcal{NS}$, an update over this extended framework could be $u = +(\mathcal{NS}, \beta_1)$.

We only focus on updates that involve the addition or deletion of an attack or a support, as the effect of adding/removing arguments to/from an ASAF has a trivial effect on its extensions. In fact, in such cases, an updated extension E of the updated ASAF $u(\Delta)$ can be trivially obtained by adding/removing the added/removed argument to/from an initial extension E_0 of Δ.

It should be noted that not every update has a large impact on the acceptability of the elements of the ASAF it is applied on. That is, there are some situations in which an update is *irrelevant* for the acceptability calculus, meaning an extension of the updated ASAF can be easily obtained, without requiring its overall computation. More in detail, given an extension E_0 of an ASAF Δ and an irrelevant update $u = +X$, an extension E of the updated ASAF $u(\Delta)$ will at most differ from E_0 in that it includes X. Similarly, given an irrelevant update $u = -X$, the updated extension will differ from the initial extension in that it does not include X.

Tables 1–4 illustrate all cases of irrelevant updates for an ASAF under the four considered semantics. Irrelevant updates are identified by looking at the labelling L_0 corresponding to an extension E_0 of the initial ASAF, under the semantics $\mathcal{S} \in \{\mathtt{co}, \mathtt{pr}, \mathtt{st}, \mathtt{gr}\}$. We use $\mathcal{E}_{\mathcal{S}}(\Delta)$ (resp., $\mathcal{E}_{\mathcal{S}}(u(\Delta))$) to denote the set of $\mathcal{S}$-extensions of the initial ASAF Δ (resp., of the updated ASAF $u(\Delta)$). For each irrelevant update, the caption of Tables 1–4 states how to obtain the updated extension in $\mathcal{E}_{\mathcal{S}}(u(\Delta))$ starting from the initial one in $\mathcal{E}_{\mathcal{S}}(\Delta)$. N/A in a cell means not applicable (*e.g.* in Table 4 it cannot be the case that $L_0(\mathsf{src}(\beta)) = \text{OUT}$ and $L_0(\mathsf{trg}(\beta)) = \text{IN}$).

The following theorem tells us for each of the four cases of update (addition/deletion of an attack/support) and for each combination of the labelling of the source and target of the updated interaction, and for each semantics, if the update is irrelevant or not.

Theorem 1. *Let $\Delta = \langle \mathbb{A}, \mathbb{R}, \mathbb{S} \rangle$ be an ASAF, $u = \pm X$ an update for Δ, and $u(\Delta)$ the updated ASAF. Also, let E_0 be an $\mathcal{S}$-extension of Δ and L_0 the labelling corresponding to E_0, where $\mathcal{S} \in \{\mathtt{co}, \mathtt{pr}, \mathtt{st}, \mathtt{gr}\}$. Then, for each type of update considered in Tables 1–4, a semantics $\mathcal{S}$ occurs in the cell $\langle L_0(\mathsf{src}(X)), L_0(\mathsf{trg}(X)) \rangle$ of Table i ($i \in [1, 4]$) iff u is irrelevant for Δ w.r.t. $\mathcal{S}$ and an $\mathcal{S}$-extension E of $u(\Delta)$ can be obtained directly from E_0 as described in the caption of Table i.*

Proof. (Sketch) The proof considers each kind of update separately and then moves to considering, for each of the four semantics appearing in a given cell, the initial labelling (L_0) of the source and target of the updated interaction. For each case of update that is not irrelevant, a counter-example showing that the initial labelling changes after performing an update is given. As an example, the proof corresponding to the top-middle cell in Table 3 follows. Consider the update $u = +\beta$, with $\beta \in (\mathbb{S}_{\mathbb{U}} \backslash \mathbb{S})$. Also, let L_0 be an $\mathcal{S}$-labelling of Δ, with $\mathcal{S} \in \{\mathtt{co}, \mathtt{pr}, \mathtt{gr}\}$, s.t. $L_0(\mathsf{src}(\beta)) = \text{IN}$ and $L_0(\mathsf{trg}(\beta)) = \text{UNDEC}$. We have to show that there exists an $\mathcal{S}$-labelling L of $u(\Delta)$ s.t. $\text{IN}(L_0) = \text{IN}(L) \cup \{\beta\}$. First, since no interaction in $u(\Delta)$ can target β, no defeats on β will exist in $u(\Delta)$ and β will be IN in every co, pr and gr-labelling of $u(\Delta)$. Then, we have to show that $\text{IN}(L_0) \subseteq \text{IN}(L)$. The only way in which the addition of β can change the labelling of other elements in $u(\Delta)$ is by generating new defeats towards them, in which β will be involved. Since $L_0(\mathsf{src}(\beta)) = \text{IN}$, it holds that $\forall \alpha_1 \in \mathbb{R}$, $\forall \mathbf{S}' \subseteq \mathbb{S}$ s.t. α_1 def $\mathsf{src}(\beta)$ given $\mathbf{S}'$, $\exists Y \in (\{\alpha_1\} \cup \mathbf{S}')$ s.t. $L_0(Y) = \text{OUT}$. Then, since every new defeat existing in $u(\Delta)$ but not in Δ also has α_1 and the supports in $\mathbf{S}'$ among its originating elements, the new defeats do not affect the labelling of the arguments, attacks or supports they defeat. Therefore, $\forall Z \in (\mathbb{A} \cup \mathbb{R} \cup \mathbb{S})$, Z maintains in $u(\Delta)$ the labelling assigned by L_0 in Δ. Consequently, there exists an $\mathcal{S}$-labelling L of $u(\Delta)$ corresponding to an $\mathcal{S}$-extension E of $u(\Delta)$ s.t. $E = E_0 \cup \{\beta\}$. $\square$

Tables 1 and 2 generalize similar results for AFs [1] to ASAFs where the target of an interaction can be also an attack or support.

Table 1: Cases in which the update $u = +\alpha$, with $\alpha \in (\mathbb{R}_{\mathbb{U}} \backslash \mathbb{R})$ is irrelevant. If $L_0(\mathsf{src}(\alpha)) = \text{IN}$, then $E_0 \cup \{\alpha\} \in \mathcal{E}_{\mathcal{S}}(u(\Delta))$; otherwise, $E_0 \in \mathcal{E}_{\mathcal{S}}(u(\Delta))$.

Update $u = +\alpha$, with $\alpha \in (\mathbb{R}_{\mathbb{U}} \backslash \mathbb{R})$		$L_0(\mathsf{trg}(\alpha))$		
		IN	UNDEC	OUT
$L_0(\mathsf{src}(\alpha))$	IN			co,pr,st,gr
	UNDEC		co,gr	co,pr,gr
	OUT	co,pr,st	co,pr,gr	co,pr,st,gr

Table 2: Cases in which the update $u = -\alpha$, with $\alpha \in \mathbb{R}$, is irrelevant. In all cases, $E_0 \backslash \{\alpha\} \in \mathcal{E}_{\mathcal{S}}(u(\Delta))$.

Update $u = -\alpha$, with $\alpha \in \mathbb{R}$		$L_0(\mathsf{trg}(\alpha))$		
		IN	UNDEC	OUT
$L_0(\mathsf{src}(\alpha))$	IN	N/A	N/A	
	UNDEC	N/A		co,pr,gr
	OUT	co,pr,st,gr	co,pr,gr	co,pr,st,gr

AF for ASAF. We now introduce the notion of *AF for an ASAF*, corresponding to an AF that encodes every argument, attack, and support of the ASAF. Then, we show that there exists a one-to-one

Table 3: Cases in which the update $u = +\beta$, with $\beta \in (\mathbb{S}_{\mathbb{U}} \backslash \mathbb{S})$ is irrelevant. In all cases, $E_0 \cup \{\beta\} \in \mathcal{E}_{\mathcal{S}}(u(\Delta))$.

Update $u = +\beta$, with $\beta \in (\mathbb{S}_{\mathbb{U}} \backslash \mathbb{S})$		$L_0(\mathsf{trg}(\beta))$ IN	UNDEC	OUT
$L_0(\mathsf{src}(\beta))$	IN	co,pr,st,gr	co,pr,gr	co,pr,st,gr
	UNDEC		co,gr	co,pr,gr
	OUT			co,pr,st,gr

Table 4: Cases in which the update $u = -\beta$, with $\beta \in \mathbb{S}$, is irrelevant. In all cases, $E_0 \backslash \{\beta\} \in \mathcal{E}_{\mathcal{S}}(u(\Delta))$.

Update $u = -\beta$, with $\beta \in \mathbb{S}$		$L_0(\mathsf{trg}(\beta))$ IN	UNDEC	OUT
$L_0(\mathsf{src}(\beta))$	IN	co,pr,st,gr	co,pr,gr	co,pr,st,gr
	UNDEC	N/A		co,pr,gr
	OUT	N/A	N/A	

correspondence between the $\mathcal{S}$-extensions of an ASAF and the $\mathcal{S}$-extensions of its AF, where $\mathcal{S} \in \{\texttt{co}, \texttt{pr}, \texttt{st}, \texttt{gr}\}$.

Briefly, the set of arguments of the AF consists of the arguments of Δ plus a pair of arguments, α and α^*, for each attack α in Δ and a pair of arguments, β and β^*, for each support β in Δ. Arguments α and α^* determine whether α is accepted or not, and are used to propagate defeats on the source of α to the attack itself. Argument β represents the support itself and is used to determine whether it is accepted or not, whereas argument β^* is used to propagate defeats on the source of β to its target. Then, the attacks of the AF are as follows. For each attack α in Δ, the AF contains a chain of 3 attacks starting in the source of α and ending in its target, with intermediate arguments α and α^*; moreover, if the target of α is a support β, then an attack between α and β^* is added to the AF. For each support β in Δ, the AF contains a chain of 2 attacks starting in the source of β and ending in its target, with intermediate argument β^*; moreover, if the target of β is a support β_1, an attack between β^* and β_1^* is added.

Definition 9 (AF for ASAF). *Let $\Delta = \langle \mathbb{A}, \mathbb{R}, \mathbb{S} \rangle$ be an ASAF. The* AF for Δ *is $\Delta_{AF} = \langle \mathsf{A}, \mathsf{R} \rangle$, where:*

- $\mathsf{A} = \mathbb{A} \cup \{\alpha, \alpha^* \mid \alpha \in \mathbb{R}\} \cup \{\beta, \beta^* \mid \beta \in \mathbb{S}\}$.
- $\mathsf{R} = \{(\mathsf{src}(\alpha), \alpha^*), (\alpha^*, \alpha), (\alpha, \mathsf{trg}(\alpha)) \mid \alpha \in \mathbb{R}\} \cup \{(\alpha, \mathsf{trg}(\alpha)^*) \mid \alpha \in \mathbb{R}, \mathsf{trg}(\alpha) \in \mathbb{S}\} \cup \{(\mathsf{src}(\beta), \beta^*), (\beta^*, \mathsf{trg}(\beta)) \mid \beta \in \mathbb{S}\} \cup \{(\beta^*, \mathsf{trg}(\beta)^*) \mid \beta \in \mathbb{S}, \mathsf{trg}(\beta) \in \mathbb{S}\}$.

Example 6. *The AF for the ASAF Δ_1 from Example 1 is Δ_{AF}, shown in Figure 2. For instance, the attack $\alpha_3 = (\mathcal{PT}, \mathcal{D})$ corresponds to the chain of 3 attacks from $\mathcal{PT}$ to $\mathcal{D}$. The attack $\alpha_4 = (\mathcal{P}, \alpha_3)$ corresponds to the chain of 3 attacks from $\mathcal{P}$ to α_3. The support $\beta_1 = (\mathcal{ST}, \mathcal{D})$ corresponds to the attacks*

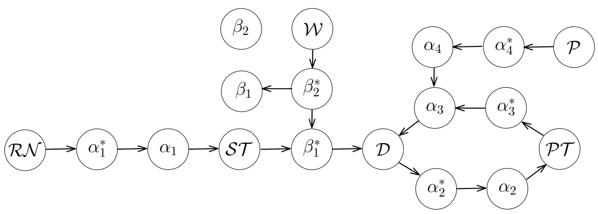

Figure 2: AF for the ASAF Δ_1

$(\mathcal{ST}, \beta_1^*), (\beta_1^*, \mathcal{D})$, while the attack $\beta_2 = (\mathcal{W}, \beta_1)$ corresponds to the attacks $(\mathcal{W}, \beta_2^*), (\beta_2^*, \beta_1)$, plus (β_2^*, β_1^*).

Before showing the equivalence between $\mathcal{S}$-extensions of an ASAF and $\mathcal{S}$-extensions of its AF, we introduce functions mapping extensions of Δ into extensions of Δ_{AF} and vice-versa.

Definition 10 (`ASAFtoAF` and `AFtoASAF` Functions). *Let $\Delta = \langle \mathbb{A}, \mathbb{R}, \mathbb{S} \rangle$ be an ASAF and $\Delta_{AF} = \langle \mathsf{A}, \mathsf{R} \rangle$ its AF. We define the functions* `ASAFtoAF` *and* `AFtoASAF`*, where E is a complete extension of Δ and E' is a complete extension of Δ_{AF}:*

- $\texttt{ASAFtoAF}(E) = \{X \mid X \in E\} \cup \{\alpha^* \mid \alpha \in \mathbb{R} \backslash E, \mathsf{src}(\alpha) \notin E$ *and* $\exists \alpha_1 \in E, \exists \mathbf{S} \subseteq E$ *s.t.* α_1 def $\mathsf{src}(\alpha)$ given $\mathbf{S}\ \} \cup \{\beta^* \mid \beta \in \mathbb{S} \cap E$, *and* $\exists \alpha_1 \in E, \exists \mathbf{S} \subseteq E$ *s.t.* α_1 def $\mathsf{src}(\beta)$ given $\mathbf{S}\ \}$
- $\texttt{AFtoASAF}(E') = E' \backslash \{\alpha^*, \beta^* \mid \alpha \in \mathbb{R}, \beta \in \mathbb{S}\}$

The following theorem states that the AF for a given ASAF yields extensions which are equivalent to those of the ASAF.

Theorem 2. *Let $\Delta = \langle \mathbb{A}, \mathbb{R}, \mathbb{S} \rangle$ be an ASAF, $\Delta_{AF} = \langle \mathsf{A}, \mathsf{R} \rangle$ its AF, $E \subseteq (\mathbb{A} \cup \mathbb{R} \cup \mathbb{S})$, $E' \subseteq \mathsf{A}$ and $\mathcal{S} \in \{\texttt{co}, \texttt{pr}, \texttt{st}, \texttt{gr}\}$. It holds that E is an $\mathcal{S}$-extension of Δ iff $\texttt{ASAFtoAF}(E)$ is an $\mathcal{S}$-extension of Δ_{AF}. Equivalently, it holds that E' is an $\mathcal{S}$-extension of Δ_{AF} iff $\texttt{AFtoASAF}(E')$ is an $\mathcal{S}$-extension of Δ.*

Then, given an ASAF and an update, we can obtain its AF and perform the update on this AF. For this purpose, we redefine an update u over an ASAF in terms of a set of updates u' over its AF, and say that u' is the set of updates *corresponding* to u. Briefly, each update identified in Definition 8 will be expressed as a set of updates in terms of the elements of the AF, following the translation given in Definition 9. In case of an addition update $+X$, the corresponding arguments X and X^* will be added first. Then, the set of updates on the AF can be directly obtained by applying Definition 9: it is the set of attacks belonging to the AF for the updated ASAF but not belonging to the AF for the initial ASAF. On the other hand, the set of updates on the AF corresponding to the removal of an attack or a support in the ASAF are those identified in Definition 9 by the attacks corresponding to the mapped attack or support.

4 Computing Extensions of Updated ASAFs

Given an ASAF Δ_0, a semantics $\mathcal{S} \in \{\texttt{co}, \texttt{pr}, \texttt{st}, \texttt{gr}\}$, an extension $E_0 \in \mathcal{E}_{\mathcal{S}}(\Delta_0)$, and an update $u = \pm X$, we define an incremental algorithm (Algorithm 1) for computing an $\mathcal{S}$-extension E of the updated ASAF $u(\Delta_0)$, if it exists.[4]

Algorithm 1 first checks if the update is irrelevant at Line 1, where $checkIrrelevantUpdate(\Delta_0, u, E_0, \mathcal{S})$ is a function returning *true* iff some condition of Theorem 1 holds. If this is the case, Algorithm 1 returns the updated extension directly obtained from E_0 as given in Theorem 1. Otherwise, it computes the AF Δ_{AF} for Δ_0 (Line 4) according to Definition 9, and, if the update is an addition, augments Δ_{AF} with the auxiliary arguments X and X^*, obtaining the AF Δ'_{AF}. Then, the set u' of updates over Δ'_{AF} corresponding to u is built and an $\mathcal{S}$-extension E'_0 of Δ'_{AF} corresponding to E_0 is computed (Lines 6–7). Next, function $\mathsf{Incr\text{-}Alg}$ encoding the incremental algorithm proposed in [1] for Dung's AFs is invoked. $\mathsf{Incr\text{-}Alg}$ takes as input the parameters $\Delta'_{AF}, u', \mathcal{S}, E'_0$, and an external solver $\mathsf{Solver}_{\mathcal{S}}$ that computes an $\mathcal{S}$-extension for the input AF. $\mathsf{Incr\text{-}Alg}$ returns an $\mathcal{S}$-extension for $u'(\Delta'_{AF})$ by first identifying a suitable sub-graph of the updated AF, then computing an $\mathcal{S}$-extension

[4] The set of stable extensions $\mathcal{E}_{\texttt{st}}(u(\Delta_0))$ of the updated ASAF may be empty; in this case, the algorithm returns $\perp$.

Algorithm 1 *DynamicASAF*($\Delta_0, u, \mathcal{S}, E_0$, $\mathsf{Solver}_\mathcal{S}$)

Input: ASAF $\Delta_0 = \langle \mathbb{A}, \mathbb{R}, \mathbb{S} \rangle$, update $u = \pm X$ (X is either an attack or a support), semantics $\mathcal{S} \in \{\mathtt{co}, \mathtt{pr}, \mathtt{st}, \mathtt{gr}\}$, initial $\mathcal{S}$-extension E_0 of Δ_0, function $\mathsf{Solver}_\mathcal{S}(\langle \mathsf{A}, \mathsf{R} \rangle)$ returning an $\mathcal{S}$-extension for an AF $\langle \mathsf{A}, \mathsf{R} \rangle$ if it exists, $\perp$ otherwise.
Output: $\mathcal{S}$-extension E for $u(\Delta_0)$ if it exists, $\perp$ otherwise.

1: **if** $checkIrrelevantUpdate(\Delta_0, u, E_0, \mathcal{S})$ **then**
2: Obtain E from E_0 as per Theorem 1;
3: **return** E**;**
4: Let $\Delta_{AF} = \langle \mathsf{A}, \mathsf{R} \rangle$ be the AF for Δ_0;
5: Let Δ'_{AF}=$\langle \mathsf{A} \cup \{X, X^*\}, \mathsf{R} \rangle$ if u=$+X$, otherwise Δ'_{AF}=Δ_{AF};
6: Let u' be the set of updates over Δ'_{AF} corresponding to u;
7: Let $E'_0 = \mathtt{ASAFtoAF}(E_0) \cup E_X$, where $E_X = \{X, X^*\}$ if $u = +X$, otherwise $E_X = \emptyset$;
8: Let $E' = \mathsf{Incr\text{-}Alg}(\Delta'_{AF}, u', \mathcal{S}, E'_0, \mathsf{Solver}_\mathcal{S})$;
9: **if** $(E' \neq \perp)$ **then**
10: **return** $E = \mathtt{AFtoASAF}(E')$;
11: **else**
12: **return** $\perp$;

on this part only by using $\mathsf{Solver}_\mathcal{S}$, and finally merging it with E'_0 to get an extension E' of the updated AF. Finally, the extension of the updated ASAF (if any) is obtained by mapping the extension E' returned by $\mathsf{Incr\text{-}Alg}$ to the corresponding ASAF extension using the result of Theorem 2 (Line 10).

As stated next, Algorithm 1 is sound and complete.

Theorem 3. *For any ASAF Δ_0, extension $E_0 \in \mathcal{E}_\mathcal{S}(\Delta_0)$, update u, and semantics $\mathcal{S} \in \{\mathtt{co}, \mathtt{pr}, \mathtt{st}, \mathtt{gr}\}$, if $\mathsf{Solver}_\mathcal{S}$ is sound and complete then Algorithm 1 computes $E \in \mathcal{E}_\mathcal{S}(u(\Delta_0))$ if $\mathcal{E}_\mathcal{S}(u(\Delta_0))$ is not empty, otherwise it returns $\perp$.*

5 Empirical Evaluation

We implemented a C++ prototype and, for each semantics $\mathcal{S} \in \{\mathtt{co}, \mathtt{pr}, \mathtt{st}, \mathtt{gr}\}$, compared the performance of Algorithm 1—where $\mathsf{Solver}_\mathcal{S}$ is a solver that won the last or second-to-last edition of the ICCMA competition for the task of determining an $\mathcal{S}$-extension—against the computation from scratch, that is, the computation of an extension of the updated ASAF by running $\mathsf{Solver}_\mathcal{S}$ directly on the corresponding updated AF. The latter is the best competitor we can compare with as, to our knowledge, there is no other available ASAF solver. As for $\mathsf{Solver}_\mathcal{S}$, it is either *$\mu$-toksia* [44] that won the ICCMA'19 competition for all the considered semantics, or *ArgSemSAT* [28] for $\mathcal{S} = \mathtt{pr}$, *goDIAMOND* [48] for $\mathcal{S} = \mathtt{st}$, *heureka* [36] for $\mathcal{S} = \mathtt{gr}$, and *cegartix* [34] for $\mathcal{S} = \mathtt{co}$, the winners of ICCMA'17.

Dataset. Although there are several benchmark generators and solvers for Dung's AFs [49], no benchmark is available for ASAFs. Thus, we generated a set of benchmark ASAFs by starting from AFs used as benchmarks at ICCMA for the tracks SE-$\mathtt{pr}$ and SE-$\mathtt{st}$ of determining an $\mathcal{S}$-extension. Specifically, we use the AF datasets named $B1$ and $B2$ both consisting of 50 AFs, and $B3$ consisting of 100 AFs, and given a benchmark AF $\langle \mathsf{A}, \mathsf{R} \rangle$, we generate an ASAF as follows: 30% of attacks in R are transformed into first-level supports; 12% (resp. 3%) of attacks in R are transformed into second-level supports towards a support (resp. an attack); 3% (resp. 2%) of attacks in R are transformed into third-level supports supports towards a support (resp. an attack); 12% (resp. 3%) of attacks in R are transformed into second-level attacks towards an attack (resp. a support); 2% (resp. 3%) of attacks in R are transformed into third-level attacks towards an attack (resp. a support); the remaining 30% of attacks in R are kept as first-level attacks of the resulting ASAF. This benchmark generation process aimed at preserving AFs' topology as much as possible. However, the process of generating ASAF benchmarks starting from AF benchmarks is challenging because we require specific amounts of different kind of attacks and supports, and we also need to check that the sub-graph induced by first-level supports is acyclic. Hence, to make it feasible, for each dataset, we generated an ASAF Δ if the number of arguments $|\mathsf{A}|$ of the associated AF Δ_{AF} does not exceed the number of arguments of the biggest AF in the original dataset. Therefore, starting from the AF datasets $B1$, $B2$, and $B3$, we obtained an ASAF dataset B_Δ consisting of 139 ASAFs $\langle \mathbb{A}, \mathbb{R}, \mathbb{S} \rangle$ with a number of arguments $|\mathbb{A}| \in [35, 2.2K]$ and a number of interactions $|\mathbb{R} \cup \mathbb{S}| \in [48, 271K]$. The associated set of AFs consists of 139 AFs $\langle \mathsf{A}, \mathsf{R} \rangle$ with $|\mathsf{A}| \in [138, 251K]$ and $|\mathsf{R}| \in [128, 338K]$.

Methodology. For each semantics $\mathcal{S} \in \{\mathtt{co}, \mathtt{pr}, \mathtt{st}, \mathtt{gr}\}$ and ASAF Δ_0 in the datasets, we consider *i)* a randomly selected $\mathcal{S}$-extension E_0 of Δ_0 (initial extension), and *ii)* an update u selected among one of the possible 12 types (addition/deletion of an attack/support towards an argument/attack/support). Next, we compute an $\mathcal{S}$-extension E of the updated ASAF $u(\Delta_0)$ by calling our incremental algorithm *DynamicASAF*. Finally, the average run time of *DynamicASAF* to compute an $\mathcal{S}$-extension is compared with the average run time required by the competitor, which computes an $\mathcal{S}$-extension of the updated ASAF by directly computing an $\mathcal{S}$-extension of the updated AF for $u(\Delta_0)$. The experiments have been carried out on an Intel Core i7-3770K CPU 3.5GHz, 12GB RAM, running Ubuntu.

Results. Figure 3 reports the average run times (log scale) of the incremental computation (*DynamicASAF*) and the computation from scratch over B_Δ for the grounded (top), preferred (center) and stable (bottom) semantics. Each data point reported in the figure is the average over 12 runs, each of them corresponding to a different update. For the sake of readability, Figure 3 also shows the lines obtained by linear regression, and the results obtained for the complete semantics are not shown as they were analogous to those obtained for the grounded semantics. Moreover, the diagrams for the stable and grounded semantics contain less data points than those for the preferred one. As for the stable semantics, the missing points correspond to generated benchmark ASAFs having no stable extension (it is not possible to run Algorithm 1 without the input parameter E_0). As for the grounded semantics, the missing points correspond to ASAFs whose grounded extension is empty, for which we found that running the incremental algorithm gives worthless improvement—in fact, for the grounded semantics, one could run the incremental approach only if $E_0 \neq \emptyset$. This is due to the fact that $\mathsf{Incr\text{-}Alg}$ uses the initial extension E_0 to compute a sub-AF (called *reduced* AF) that is used for the computation of an extension of the updated AF. But if E_0 is empty and the considered semantics is polynomial, as for the grounded one [33], the overhead of computing the reduced AF may not pay off. Finally, the diagrams on the right-hand side do not contain data points for a large number of ASAF interactions because the solver ran into memory capacity saturation.

The results in Figure 3 show that, on average, the incremental algorithm outperforms the competitors that compute the extensions from scratch. However, the improvement obtained for the preferred and stable semantics—for which *DynamicASAF* is between one and two orders of magnitude faster than the competitors—is better than that obtained for the grounded semantics. This happens because the harder the computation from scratch is for a given semantics, the larger the improvements are for $\mathsf{Incr\text{-}Alg}$ [1]. In fact, the improve-

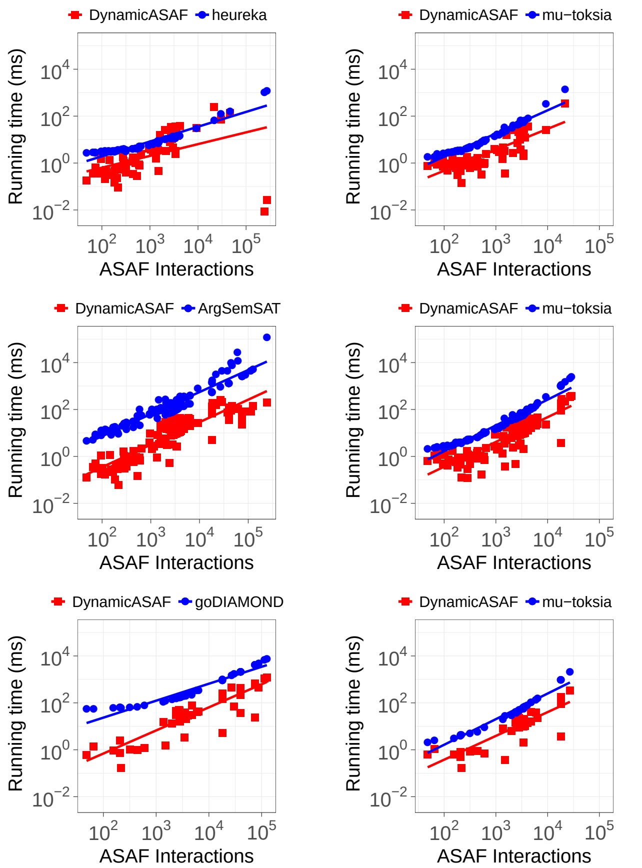

Figure 3: Run times (ms) of ICCMA solvers and *DynamicASAF* versus the number of interactions (attacks and supports) for the ASAFs in $B_\triangle$ and the grounded (top), preferred (center), and stable (bottom) semantics

ment of Incr-Alg is proportional to the size of the restriction (arguments and attacks of the input AF not occurring in the reduced AF). Thus, for the semantics whose computation is exponential in the size of the input AF, the improvement derived from considering the reduced AF is exponential in the size of the restriction. However, for faster solvers (*e.g.* μ*-toksia* vs *ArgSemSAT*) the improvement slightly decreases, though it remains greater than one order of magnitude.

As for the grounded semantics, the median of the improvements w.r.t. the ICMMA'17 solver is 4.3, and the average improvement without considering the two outlier ASAFs having the largest size is 5.6 (the average considering these two ASAFs would be skewed by the extremely large improvements obtained for those ASAFs). Also, the average improvement w.r.t. the ICMMA'19 solver is 4.1.

Finally, the experiments also showed that the identification of irrelevant updates is useful as they allowed to directly obtain the updated extension of the input ASAF for 42.4% of the updates performed on average. In particular, the average improvement for the case where only relevant updates are considered is almost half of the improvement for the case where all updates are considered. However, the improvements remain of the same order of magnitude.

6 Related Work

There have been significant efforts coping with dynamic aspects of Dung's abstract argumentation framework. [21, 22] have investigated the principles according to which the grounded extension of an AF does not change when the set of arguments/attacks are changed. [24, 25] addressed the problem of revising the set of extensions of an AF, and studied how the extensions can evolve when a new argument is considered. [18] have studied the evolution of the set of extensions after performing a change operation (addition/removal of arguments/interactions). Dynamic argumentation has been applied to decision-making of an autonomous agent by [6], where it is studied how the acceptability of arguments evolves when a new argument is added to the decision system. The division-based method, proposed by [42] and then refined by [12], divides the updated framework into two parts: *affected* and *unaffected*, where only the status of affected arguments is recomputed after updates. [41] 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 [40], 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. [50] introduced a matrix representation of argumentation frameworks and proposed a matrix reduction that, when applied to dynamic AFs, resembles the division-based method in [42].

Relevant work on dynamic aspects of Dung's AFs also includes the following. [13] have 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. [16] have investigated whether and how it is possible to modify a given AF so that a desired set of arguments becomes an extension, whereas [46] have studied equivalence between two AFs when further information (another AF) is added to both AFs. [14] have focused on expansions where new arguments and attacks may be added but the attacks among the old arguments remain unchanged, while [15] characterized update and deletion equivalence, where adding/deleting arguments/attacks is allowed. (deletions were not considered by [46, 14]).

Bipolarity in argumentation is discussed in [5], where a survey of the use of bipolarity and a formal definition of 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 [30]. Changes in bipolar argumentation frameworks (BAFs) have been studied in [27], where it is shown how the addition of one argument together with one support involving it (and without any attack) impacts the extensions of the updated BAF. The problem of incrementally computing extensions of dynamic BAFs, with deductive interpretation of support [20], has been first addressed in [1], and then extended in [2] to deal with second-order attacks [9]. However, none of the existing approaches deals with the incremental computation and the empirical evaluation of a general framework as the ASAF framework.

Finally, it is worth noting that our translation from an ASAF to an AF improves that of [37] from two standpoints: *i)* it is direct, not requiring the two-step process of [37] which first obtains an Argumentation Framework with Necessities (AFN) [45] and then an AF; *ii)* the size of the resulting AF is smaller than that of [37].

7 Conclusions and Future Work

To the best of our knowledge, this is the first paper addressing the problem of efficiently and incrementally computing extensions for dynamic ASAFs. Given the generality of the ASAF, our technique can be also applied to restricted frameworks such as Argumentation

Frameworks with Recursive Attacks (AFRAs) [11] and AFNs [45] (and, of course, Dung's AFs). Moreover, although for the sake of the presentation we have focused on single attack/support updates, the technique also applies to the case of multiple ASAF updates consisting of the addition/removal of a set of attacks and supports.

Future work will be devoted to extending our technique to deal with other computational problems, such as enumerating extensions and deciding credulous/sceptical acceptance in dynamic ASAFs.

REFERENCES

[1] G. Alfano, S. Greco, and F. Parisi, 'Efficient computation of extensions for dynamic abstract argumentation frameworks: An incremental approach', in *Proc. of IJCAI*, pp. 49–55, (2017).

[2] G. Alfano, S. Greco, and F. Parisi, 'A meta-argumentation approach for the efficient computation of stable and preferred extensions in dynamic bipolar argumentation frameworks', *Intelligenza Artificiale*, **12**(2), 193–211, (2018).

[3] G. Alfano, S. Greco, and F. Parisi, 'An efficient algorithm for skeptical preferred acceptance in dynamic argumentation frameworks', in *Proc. of IJCAI*, pp. 18–24, (2019).

[4] G. Alfano, S. Greco, F. Parisi, G. I. Simari, and G. R. Simari, 'An incremental approach to structured argumentation over dynamic knowledge bases', in *Proc. of KR*, pp. 78–87, (2018).

[5] L. Amgoud, C. Cayrol, and M.-C. Lagasquie-Schiex, 'On the bipolarity in argumentation frameworks', in *Proc. of NMR*, pp. 1–9, (2004).

[6] L. Amgoud and S. Vesic, 'Revising option status in argument-based decision systems', *J. Log. Comp.*, **22**(5), 1019–1058, (2012).

[7] P. Baroni, G. Boella, F. Cerutti, M. Giacomin, L. W. N. van der Torre, and S. Villata, 'On the input/output behavior of argumentation frameworks', *Artif. Intell.*, **217**, 144–197, (2014).

[8] P. Baroni, M. Caminada, and M. Giacomin, 'An introduction to argumentation semantics', *The Knowl. Eng. Rev.*, **26**(4), 365–410, (2011).

[9] P. Baroni, F. Cerutti, M. Giacomin, and G. Guida, 'Encompassing attacks to attacks in abstract argumentation frameworks', in *Proc. of ECSQARU*, pp. 83–94, (2009).

[10] P. Baroni, F. Cerutti, M. Giacomin, and G. Guida, 'AFRA: Argumentation Framework with Recursive Attacks', *Int. J. Approx. Reasoning*, **52**(1), 19–37, (2011).

[11] P. Baroni, F. Cerutti, M. Giacomin, and G. Guida, 'AFRA: Argumentation Framework with Recursive Attacks', *Int. J. Approx. Reasoning*, **52**(1), 19–37, (2011).

[12] P. Baroni, M. Giacomin, and 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**, 104–115, (2014).

[13] R. Baumann, 'Splitting an argumentation framework', in *Proc. of LPNMR*, pp. 40–53, (2011).

[14] R. Baumann, 'Normal and strong expansion equivalence for argumentation frameworks', *Artif. Intell.*, **193**, 18–44, (2012).

[15] R. Baumann, 'Context-free and context-sensitive kernels: Update and deletion equivalence in abstract argumentation', in *Proc. of ECAI*, pp. 63–68, (2014).

[16] R. Baumann and G. Brewka, 'Expanding argumentation frameworks: Enforcing and monotonicity results', in *COMMA*, pp. 75–86, (2010).

[17] T. J. M. Bench-Capon and P. E. Dunne, 'Argumentation in artificial intelligence', *Artif. Intell.*, **171**, 619 – 641, (2007).

[18] P. Bisquert, C. Cayrol, F. Dupin de Saint-Cyr, and M.-C. Lagasquie-Schiex, 'Characterizing change in abstract argumentation systems', in *Trends in Belief Revision and Argumentation Dynamics*, volume 48, 75–102, (2013).

[19] S. Bistarelli, F. Faloci, F. Santini, and C. Taticchi, 'Studying dynamics in argumentation with Rob', in *Proc. of COMMA*, pp. 451–452, (2018).

[20] G. Boella, D. M. Gabbay, L. W. N. van der Torre, and S. Villata, 'Support in abstract argumentation', in *COMMA*, pp. 111–122, (2010).

[21] G. Boella, S. Kaci, and L. W. N. van der Torre, 'Dynamics in argumentation with single extensions: Attack refinement and the grounded extension (extended version)', in *ArgMAS 2009*.

[22] G. Boella, S. Kaci, and L. W. N. van der Torre, 'Dynamics in argumentation with single extensions: Abstraction principles and the grounded extension', in *Proc. of ECSQARU*, pp. 107–118, (2009).

[23] M. W. A. Caminada and D. M. Gabbay, 'A logical account of formal argumentation', *Studia Logica*, **93**(2-3), 109–145, (2009).

[24] C. Cayrol, F. Dupin de Saint-Cyr, and M.-C. Lagasquie-Schiex, 'Revision of an argumentation system', in *Proc. of KR*, pp. 124–134, (2008).

[25] C. Cayrol, F. Dupin de Saint-Cyr, and M.-C. Lagasquie-Schiex, 'Change in abstract argumentation frameworks: Adding an argument', *Journal of Art. Int. Res.*, **38**, 49–84, (2010).

[26] C. Cayrol and M.-C. Lagasquie-Schiex, 'On the acceptability of arguments in bipolar argumentation frameworks', in *Proc. of ECSQARU*, pp. 378–389, (2005).

[27] C. Cayrol and M.-C. Lagasquie-Schiex, 'Change in abstract bipolar argumentation systems', in *Proc. of SUM*, pp. 314–329, (2015).

[28] F. Cerutti, M. Vallati, M. Giacomin, and T. Zanetti, 'Argsemsat-2017', *Second ICCMA*, (2017).

[29] G. Charwat, W. Dvorák, S. A. Gaggl, J. P. Wallner, and S. Woltran, 'Methods for solving reasoning problems in abstract argumentation - A survey', *AIJ*, **220**, 28–63, (2015).

[30] A. Cohen, S. Gottifredi, A. J. Garcia, and G. R. Simari, 'A survey of different approaches to support in argumentation systems', *The Knowl. Eng. Rev.*, **29**(5), 513–550, (2014).

[31] S. Doutre and J.-G. Mailly, 'Constraints and changes: A survey of abstract argumentation dynamics', *Argument & Computation*, **9**(3), 223–248, (2018).

[32] 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), 321–358, (1995).

[33] P. E. Dunne and M. Wooldridge, 'Complexity of abstract argumentation', in *Argumentation in Artificial Intelligence*, 85–104, (2009).

[34] W. Dvorák, M. Järvisalo, and J. P Wallner, 'Cegartix v2017-3-13: A sat-based counter-example guided argumentation reasoning tool', *Second ICCMA*, (2017).

[35] M. A. Falappa, A. J. Garcia, G. Kern-Isberner, and G. R. Simari, 'On the evolving relation between belief revision and argumentation', *The Knowl. Eng. Rev.*, **26**(1), 35–43, (2011).

[36] N. Geilen and M. Thimm, 'Heureka: A general heuristic backtracking solver for abstract argumentation', in *TAFA*, pp. 143–149, (2017).

[37] S. Gottifredi, A. Cohen, A. J. Garcia, and G. R. Simari, 'Characterizing acceptability semantics of argumentation frameworks with recursive attack and support relations', *Artif. Intell.*, **262**, 336–368, (2018).

[38] S. Greco and F. Parisi, 'Incremental computation of deterministic extensions for dynamic argumentation frameworks', in *Proc. of European Conf. on Logics in Artif. Intell. (JELIA)*, pp. 288–304, (2016).

[39] A. Hunter, 'Towards a framework for computational persuasion with applications in behaviour change', *Argument & Computation*, **9**(1), 15–40, (2018).

[40] B. Liao, 'Toward incremental computation of argumentation semantics: A decomposition-based approach', *Ann. Math. Artif. Intell.*, **67**(3-4), 319–358, (2013).

[41] B. Liao and H. Huang, 'Partial semantics of argumentation: basic properties and empirical results', *J. Log. Comp.*, **23**(3), 541–562, (2013).

[42] B. Liao, L. J., and R. C. Koons, 'Dynamics of argumentation systems: A division-based method', *Artif. Intell.*, **175**(11), 1790–1814, (2011).

[43] 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, and S. Woltran, *Agreement Technologies*, volume 8 of *Law, Governance and Technology*, chapter 21: The Added Value of Argumentation: Examples and Challenges, 357–404, Springer, 2013.

[44] A. Niskanen and M. Järvisalo, 'μ-toksia participating in iccma 2019', *Third ICCMA*, (2019).

[45] F. Nouioua and V. Risch, 'Argumentation frameworks with necessities', in *Proc. of SUM*, pp. 163–176, (2011).

[46] E. Oikarinen and S. Woltran, 'Characterizing strong equivalence for argumentation frameworks', *Artif. Intell.*, **175**(14-15), 1985–2009, (2011).

[47] I. Rahwan and G. R. Simari, *Argumentation in Artificial Intelligence*, Springer, 2009.

[48] H. Strass and S. Ellmauthaler, 'godiamond 0.6. 6–iccma 2017 system description, 2017', *Second ICCMA*, (2017).

[49] M. Thimm and S. Villata, 'The first international competition on computational models of argumentation: Results and analysis', *Artif. Intell.*, **252**, 267–294, (2017).

[50] Y. Xu and C. Cayrol, 'The matrix approach for abstract argumentation frameworks', in *Proc. of TAFA*, pp. 243–259, (2015).