A Basic Framework for Explanations in Argumentation

AnneMarie Borg[1] and Floris Bex[1,2]

[1]Department of Information and Computing Sciences, Utrecht University
[2]Department of Law, Technology, Markets and Society, Tilburg University

{a.borg, f.j.bex}@uu.nl

Abstract

We discuss explanations for formal (abstract and structured) argumentation – the question whether and why a certain argument or claim can be accepted (or not) under various extension-based semantics. We introduce a flexible framework, which can act as the basis for many different types of explanations. For example, we can have simple or comprehensive explanations in terms of arguments for or against a claim, arguments that (indirectly) defend a claim, the evidence (knowledge base) that supports or is incompatible with a claim, and so on. We show how different types of explanations can be captured in our basic framework, discuss a real-life application and formally compare our framework to existing work.

1 Introduction

Recently, *explainable AI* (XAI) has received much attention, mostly directed at new techniques for explaining decisions of (subsymbolic) machine learning algorithms [12]. However, explanations also play an important role in (symbolic) knowledge-based systems [8]. Argumentation is one research area in symbolic AI that is frequently mentioned in relation to XAI. For example, arguments can be used to provide reasons for or against decisions [8, 9]. The focus can also be on the argumentation itself, where it is explained whether and why a certain argument or claim can be accepted under certain semantics for computational argumentation [5, 6, 7, 13]. It is the latter type of explanations we are interested in.

Two central concepts in argumentation are *abstract argumentation frameworks* [2] – sets of arguments and the attack relations between them – and *structured* or *logical argumentation frameworks* (e.g., [11]) – where arguments are constructed from a knowledge base and a set of rules and the attack relation is based on the individual elements in the arguments. For both abstract and structured argumentation frameworks we can determine extensions, sets of arguments that can collectively be considered as acceptable, under different semantics [2]. In XAI terms [4], this is a *global* explanation – what can we conclude from the model as a whole? However, as argumentation is being applied in real-life AI systems with lay-users, we would rather have simpler, more compact explanations for the acceptability of individual arguments – a *local* explanation for a particular decision or conclusion. We noticed the need for such explanations when deploying an argumentation system at the Dutch National Police, which assists citizens in filing online reports and complaints [1, 10].

We propose a basic framework for explanations in structured and abstract argumentation, with which explanations for (non-)accepted arguments and (sub-)conclusions can be generated. Though some work on explanations for argumentation-based conclusions exists in the literature ([5, 6, 7, 13], Section 5), our framework is *generic* in that the underlying argumentation framework does not have to be adjusted and the definitions are semantics-independent – for example, the explanations based on the new semantics of Fan and Toni [5] are a special case of our framework. The framework is also *flexible*, as the contents of explanations can be varied. For example, rather than returning all defending or attacking arguments, we can return only those that can defend themselves, or the ones that directly attack an argument. Furthermore, we are the first to use the structure of arguments for explanations: not just arguments for a conclusion, but also elements of these arguments (e.g., premises or rules) can be returned as an explanation.

2 Preliminaries

An *abstract argumentation framework* (AF) [2] is a pair $\mathcal{AF} = \langle \mathsf{Args}, \mathsf{Att} \rangle$, where Args is a set of *arguments* and $\mathsf{Att} \subseteq \mathsf{Args} \times \mathsf{Args}$ is an *attack relation* on these arguments. An AF can be viewed as a directed graph, in which the nodes represent arguments and the arrows represent attacks between arguments.

Example 1. Consider the AF $\mathcal{AF}_1 = \langle \mathsf{Args}_1, \mathsf{Att}_1 \rangle$ where $\mathsf{Args}_1 = \{A_1, A_2, A_3, A_4\}$ and $\mathsf{Att}_1 = \{(A_2, A_1), (A_3, A_2), (A_3, A_4), (A_4, A_3)\}$.

Given an AF $\mathcal{AF}$, Dung-style semantics [2] can be applied to it, to determine what combinations of arguments (called *extensions*) can collectively be accepted.

Definition 1. Let $\mathcal{AF} = \langle \mathsf{Args}, \mathsf{Att} \rangle$ be an AF, $\mathsf{S} \subseteq \mathsf{Args}$ a set of arguments and let $A \in \mathsf{Args}$. Then:

- S *attacks* A if there is an $A' \in \mathsf{S}$ such that $(A', A) \in \mathsf{Att}$, S^+ denotes the set of all arguments attacked by S;
- S *defends* A if S attacks every attacker of A;
- S is *conflict-free* if there are no $A_1, A_2 \in \mathsf{S}$ such that $(A_1, A_2) \in \mathsf{Att}$; and
- S is *admissible* if it is conflict-free and it defends all of its elements.

An admissible set that contains all the arguments that it defends is a *complete extension* (cmp).

- The *grounded extension* (grd) is the minimal (with respect to $\subseteq$) complete extension;
- A *preferred extension* (prf) is a maximal (with respect to $\subseteq$) complete extension; and
- A *semi-stable extension* (sstb) S is a complete extension where $\mathsf{S} \cup \mathsf{S}^+$ is maximal.

$\mathsf{Ext}_{\mathsf{sem}}(\mathcal{AF})$ denotes the set of all the extensions of $\mathcal{AF}$ under the semantics $\mathsf{sem} \in \{\mathsf{cmp}, \mathsf{grd}, \mathsf{prf}, \mathsf{sstb}\}$.

Where $\mathcal{AF} = \langle \mathsf{Args}, \mathsf{Att} \rangle$ is an AF, sem a semantics and $\mathsf{Ext}_{\mathsf{sem}}(\mathcal{AF}) \neq \emptyset$, it is said that $A \in \mathsf{Args}$ is *skeptically* [resp. *credulously*] *accepted* if $A \in \bigcap \mathsf{Ext}_{\mathsf{sem}}(\mathcal{AF})$ [resp. $A \in \bigcup \mathsf{Ext}_{\mathsf{sem}}(\mathcal{AF})$]. These acceptability strategies are denoted by $\cap$ [resp. $\cup$]. A is said to be *skeptically* [resp. *credulously*] *non-accepted* in $\mathcal{AF}$ if for some [resp. all] $\mathcal{E} \in \mathsf{Ext}_{\mathsf{sem}}(\mathcal{AF})$, $A \notin \mathcal{E}$. When these are arbitrary, result in the same or are clear from the context, we will refer to accepted respectively non-accepted arguments.

The notions of attack and defense can also be defined between arguments:

Definition 2. Let $\mathcal{AF} = \langle \mathsf{Args}, \mathsf{Att} \rangle$ be an AF, $A, B \in \mathsf{Args}$ and $\mathcal{E} \in \mathsf{Ext}_{\mathsf{sem}}(\mathcal{AF})$ for some sem. A can defend B directly or indirectly: A *directly defends* B if there is some $C \in \mathsf{Args}$ such that $(C, B) \in \mathsf{Att}$ and $(A, C) \in \mathsf{Att}$, and A *indirectly defends* B if A defends $C \in \mathsf{Args}$ and C defends B. It is said that A *defends* B *in* $\mathcal{E}$ if A defends B and $A \in \mathcal{E}$.

Similarly, A can attack B directly or indirectly: A *directly attacks* B if $(A, B) \in \mathsf{Att}$ and A *indirectly attacks* B if A attacks some $C \in \mathsf{Args}$ and C defends B.

Next we define two notions that will be used in the basic definitions of explanations. The first, used for acceptance explanations, denotes the set of arguments that defend the argument A, while the last, used for non-acceptance explanations, denotes the set of arguments that attack A and for which there is no defense in the given extension.

Definition 3. Let $\mathcal{AF} = \langle \mathsf{Args}, \mathsf{Att} \rangle$ be an AF, $A \in \mathsf{Args}$ and $\mathcal{E} \in \mathsf{Ext}_{\mathsf{sem}}(\mathcal{AF})$ an extension for some semantics sem.

- $\mathsf{DefBy}(A) = \{B \in \mathsf{Args} \mid B \text{ defends } A\}$;
- $\mathsf{DefBy}(A, \mathcal{E}) = \mathsf{DefBy}(A) \cap \mathcal{E}$ denotes the set of arguments that defend A in $\mathcal{E}$;
- $\mathsf{NotDef}(A, \mathcal{E}) = \{B \in \mathsf{Args} \mid B \text{ attacks } A \text{ and } \mathcal{E} \text{ does not attack } B\}$, denotes the set of all attackers of A for which no defense exists from $\mathcal{E}$.

Example 2. In $\mathcal{AF}_1$ (recall Example 1), example conflict-free sets are $\{A_1, A_3\}$ and $\{A_2, A_4\}$. $\mathsf{Ext}_{\mathsf{cmp}}(\mathcal{AF}_1) = \{\emptyset, \{A_1, A_3\}, \{A_2, A_4\}\}$, while $\mathsf{Ext}_{\mathsf{prf}}(\mathcal{AF}_1) = \mathsf{Ext}_{\mathsf{sstb}}(\mathcal{AF}_1) = \{\{A_1, A_3\}, \{A_2, A_4\}\}$ and $\mathsf{Ext}_{\mathsf{grd}}(\mathcal{AF}_1) = \{\emptyset\}$. None of the arguments in Args_1 is skeptically accepted, while all of them are credulously accepted for $\mathsf{sem} \in \{\mathsf{cmp}, \mathsf{prf}, \mathsf{sstb}\}$.

Argument A_3 directly attacks A_4, and attacks A_2 both directly and indirectly. A_3 defends A_1 directly against A_2 and indirectly against A_4. Moreover, $\mathsf{DefBy}(A_1) = \{A_3\}$, $\mathsf{DefBy}(A_1, \{A_1, A_3\}) = \{A_3\}$ and $\mathsf{NotDef}(A_3, \{A_2, A_4\}) = \{A_4\}$.

2.1 ASPIC$^+$

We investigate explanations for a well-known approach to structured argumentation: ASPIC$^+$ [11], which allows for two types of premises – *axioms* that cannot be questioned and *ordinary premises* that can be questioned – and two types of rules – *strict* rules that cannot be questioned and *defeasible rules*. We choose ASPIC$^+$ as the structured argumentation approach in this paper since it allows to vary the form of the explanations in many ways (see Section 4). The definitions in this section are based on [11].

Definition 4. An *argumentation system* is a tuple AS $= \langle \mathcal{L}, \mathcal{R}, n \rangle$, where:

- $\mathcal{L}$ is a propositional language closed under classical negation ($\neg$), we denote $\psi = -\phi$ if $\psi = \neg\phi$ or $\phi = \neg\psi$.
- $\mathcal{R} = \mathcal{R}_s \cup \mathcal{R}_d$ is a set of strict ($\mathcal{R}_s$) and defeasible ($\mathcal{R}_d$) inference rules of the form $\phi_1, \ldots, \phi_n \to \phi$ resp. $\phi_1, \ldots, \phi_n \Rightarrow \phi$, such that $\{\phi_1, \ldots, \phi_n, \phi\} \subseteq \mathcal{L}$ and $\mathcal{R}_s \cap \mathcal{R}_d = \emptyset$.

  Where $r \in \mathcal{R}$, $\mathsf{Ant}(r) = \{\phi_1, \ldots, \phi_n\}$ are the *antecedents* of the rule and $\mathsf{Cons}(r) = \phi$ is the *consequent* of the rule. Moreover, $\mathsf{Rules}(\mathcal{R}, \phi) = \{r \in \mathcal{R} \mid \mathsf{Cons}(r) = \phi\}$.
- $n : \mathcal{R}_d \to \mathcal{L}$ is a naming convention for defeasible rules.

A *knowledge base* in an argumentation system $\langle \mathcal{L}, \mathcal{R}, n \rangle$ is a set of formulas $\mathcal{K} \subseteq \mathcal{L}$ which contains two disjoint subsets: $\mathcal{K} = \mathcal{K}_p \cup \mathcal{K}_n$, the set of *axioms* $\mathcal{K}_n$ and the set of *ordinary premises* $\mathcal{K}_p$.

Arguments in ASPIC$^+$ are constructed in an argumentation system from a knowledge base.

Definition 5. An *argument* A on the basis of a knowledge base $\mathcal{K}$ in an argumentation system $\langle \mathcal{L}, \mathcal{R}, n \rangle$ is:

1. ϕ if $\phi \in \mathcal{K}$, where $\mathsf{Prem}(A) = \mathsf{Sub}(A) = \{\phi\}$, $\mathsf{Conc}(A) = \phi$, $\mathsf{Rules}(a) = \emptyset$ and $\mathsf{TopRule}(A) =$ undefined;
2. $A_1, \ldots, A_n \rightsquigarrow \psi$, where $\rightsquigarrow \in \{\to, \Rightarrow\}$, if $A_1, \ldots, A_n$ are arguments such that there exists a rule $\mathsf{Conc}(A_1), \ldots, \mathsf{Conc}(A_n) \rightsquigarrow \psi$ in $\mathcal{R}_s$ if $\rightsquigarrow = \to$ and in $\mathcal{R}_d$ if $\rightsquigarrow = \Rightarrow$.

   $\mathsf{Prem}(A) = \mathsf{Prem}(A_1) \cup \ldots \cup \mathsf{Prem}(A_n)$; $\mathsf{Conc}(A) = \psi$; $\mathsf{Sub}(A) = \mathsf{Sub}(A_1) \cup \ldots \cup \mathsf{Sub}(A_n) \cup \{A\}$; $\mathsf{Rules}(A) = \mathsf{Rules}(A_1) \cup \ldots \cup \mathsf{Rules}(A_n) \cup \{\mathsf{Conc}(A_1), \ldots, \mathsf{Conc}(A_n) \rightsquigarrow \psi\}$; $\mathsf{DefRules}(A) = \{r \in \mathcal{R}_d \mid r \in \mathsf{Rules}(A)\}$; $\mathsf{TopRule}(A) = \mathsf{Conc}(A_1), \ldots, \mathsf{Conc}(A_n) \rightsquigarrow \psi$.

The above notation can be generalized to sets. For example, where S is a set of arguments $\mathsf{Prem}(\mathsf{S}) = \bigcup\{\mathsf{Prem}(A) \mid A \in \mathsf{S}\}$, $\mathsf{Conc}(\mathsf{S}) = \{\mathsf{Conc}(A) \mid A \in \mathsf{S}\}$ and $\mathsf{DefRules}(\mathsf{S}) = \bigcup\{\mathsf{DefRules}(A) \mid A \in \mathsf{S}\}$.

Example 3. $\mathrm{AS}_2 = \langle \mathcal{L}_2, \mathcal{R}_2, n \rangle$ is an argumentation system where $\mathcal{R}_2 = \mathcal{R}_s^2 \cup \mathcal{R}_d^2$ such that $\mathcal{R}_s^2 = \emptyset$, $\mathcal{R}_d^2 = \{d_1, \ldots, d_5\}$ (the application of these rules is shown in the arguments below), let $\mathcal{K}_2 = \mathcal{K}_n^2 \cup \mathcal{K}_p^2$ where $\mathcal{K}_n^2 = \{t\}$ and $\mathcal{K}_p^2 = \{r\}$. The following arguments can be constructed:

$$\begin{array}{ll}
A_1 : t & B_1 : \ r \\
A_2 : A_1 \overset{d_3}{\Rightarrow} \neg r & B_2 : \ B_1 \overset{d_2}{\Rightarrow} p \\
A_3 : A_1, A_2 \overset{d_4}{\Rightarrow} q & B_3 : \ B_1 \overset{d_5}{\Rightarrow} \neg q \\
A_4 : A_3 \overset{d_1}{\Rightarrow} p &
\end{array}$$

We denote the set of arguments constructed from AS_2 and $\mathcal{K}_2$ by Args_2. For A_4 we have that $\mathsf{Prem}(A_4) = \{t\}$, $\mathsf{Conc}(A_4) = p$, $\mathsf{Sub}(A_4) = \{A_1, A_2, A_3, A_4\}$ and $\mathsf{Rules}(A_4) = \{d_1, d_3, d_4\}$. Furthermore, $\mathsf{Rules}(\mathcal{R}_2, p) = \{d_1, d_2\}$.

Attacks on an argument are based on the rules and premises applied in the construction of that argument.

Definition 6. An argument A *attacks* an argument B iff A *undercuts*, *rebuts* or *undermines* B, where:

- A *undercuts* B (on B') iff $\mathsf{Conc}(A) = -n(r)$ for some $B' \in \mathsf{Sub}(B)$ such that B''s top rule r is defeasible, it denies a rule;
- A *rebuts* B (on B') iff $\mathsf{Conc}(A) = -\phi$ for some $B' \in \mathsf{Sub}(B)$ of the form $B_1'', \ldots, B_n'' \Rightarrow \phi$, it denies a conclusion;
- A *undermines* B (on ϕ) iff $\mathsf{Conc}(A) = -\phi$ for some $\phi \in \mathsf{Prem}(B) \setminus \mathcal{K}_n$, it denies a premise.

Argumentation theories and their corresponding Dung-style argumentation frameworks can now be defined.

Definition 7. An *argumentation theory* is a pair $\mathrm{AT} = \langle \mathrm{AS}, \mathcal{K} \rangle$, where AS is an argumentation system and $\mathcal{K}$ is a knowledge base.

A *structured argumentation framework* (SAF) defined by an argumentation theory AT is a pair $\mathcal{AF}(\mathrm{AT}) = \langle \mathsf{Args}, \mathsf{Att} \rangle$, where Args is the set of all arguments constructed from AT and $(A, B) \in \mathsf{Att}$ iff A attacks B according to Definition 6.

Dung-style semantics, as in Definition 1, can be applied to SAFs in the same way as they are applied in AFs.

Example 4. (Example 3 continued) Consider the argumentation theory $\mathrm{AT}_2 = \langle \mathrm{AS}_2, \mathcal{K}_2 \rangle$. Figure 1 contains the graphical representation of $\mathcal{AF}(\mathrm{AT}_2) = \langle \mathsf{Args}_2, \mathsf{Att}_2 \rangle$. In this framework there are no undercuts, all the attacks from A_2 are underminers and all the other attacks are rebuts.

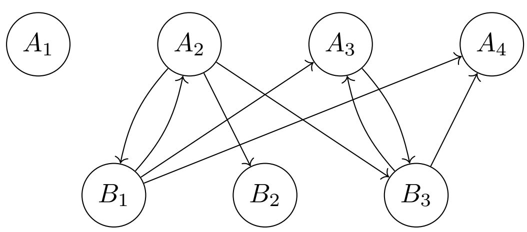


Figure 1: Graphical representation of $\mathcal{AF}(\mathrm{AT}_2)$.

Then: $\mathsf{Ext}_{\mathsf{grd}}(\mathcal{AF}(\mathrm{AT}_2)) = \{A_1\}$; and $\mathsf{Ext}_{\mathsf{sem}}(\mathcal{AF}(\mathrm{AT}_2)) = \{\{A_1, A_2, A_3, A_4\}, \{A_1, B_1, B_2, B_3\}\}$, for $\mathsf{sem} \in \{\mathsf{prf}, \mathsf{sstb}\}$.

Entailment relations, induced by the structured argumentation framework and a semantics, are defined by:

Definition 8. Let $\mathcal{AF}(\mathrm{AT}) = \langle \mathsf{Args}, \mathsf{Att} \rangle$ for a semantics sem, $\mathsf{Ext}_{\mathsf{sem}}(\mathcal{AF}) \neq \emptyset$ and let some $\phi \in \mathcal{L}$. We define:

- *Credulous entailment*: $\mathrm{AT} \mid\!\sim^{\cup}_{\mathsf{sem}} \phi$ iff for some $\mathcal{E} \in \mathsf{Ext}_{\mathsf{sem}}(\mathcal{AF})$ there is an argument $A \in \mathcal{E}$ with $\mathsf{Conc}(A) = \phi$, it is said that ϕ is *credulously accepted*;
- *Skeptical entailment*: $\mathrm{AT} \mid\!\sim^{\cap}_{\mathsf{sem}} \phi$ iff for each $\mathcal{E} \in \mathsf{Ext}_{\mathsf{sem}}(\mathcal{AF})$ there is some $A \in \mathcal{E}$ such that $\mathsf{Conc}(A) = \phi$, it is said that ϕ is *skeptically accepted*.

When arbitrary or clear from the context, the superscript will be omitted (e.g., $\mid\!\sim_{\mathsf{grd}}$ as $\mid\!\sim^{\cup}_{\mathsf{grd}}$ and $\mid\!\sim^{\cap}_{\mathsf{grd}}$ coincide).

Example 5 (Example 4 continued). For $\mathcal{AF}(\mathrm{AT}_2) = \langle \mathsf{Args}_2, \mathsf{Att}_2 \rangle$ we have that:

1. $\mathrm{AT}_2 \not\mid\!\sim_{\mathsf{grd}} \phi$ and $\mathrm{AT}_2 \not\mid\!\sim^{\cap}_{\mathsf{sem}} \phi$ for $\phi \in \{q, \neg q, r, \neg r\}$, and $\mathsf{sem} \in \{\mathsf{cmp}, \mathsf{prf}, \mathsf{sstb}\}$; while
2. $\mathrm{AT}_2 \mid\!\sim^{\cup}_{\mathsf{sem}} \phi$ for any $\phi \in \{p, q, \neg q, r, \neg r, t\}$ and $\mathsf{sem} \in \{\mathsf{cmp}, \mathsf{prf}, \mathsf{sstb}\}$;
3. $\mathrm{AT}_2 \mid\!\sim_{\mathsf{grd}} t$ and $\mathrm{AT}_2 \mid\!\sim^{\cap}_{\mathsf{sem}} t$ for $\mathsf{sem} \in \{\mathsf{cmp}, \mathsf{prf}, \mathsf{sstb}\}$; and
4. $\mathrm{AT}_2 \mid\!\sim^{\cap}_{\mathsf{sem}} p$ for $\mathsf{sem} \in \{\mathsf{prf}, \mathsf{sstb}\}$ but $\mathrm{AT}_2 \not\mid\!\sim_{\mathsf{grd}} p$.

This follows since each argument from Args_2 is part of at least one extension, but only A_1 is part of every extension. The last item follows since each sem-extension of $\mathcal{AF}(\mathrm{AT}_2)$ contains either A_4 or B_2 for $\mathsf{sem} \in \{\mathsf{prf}, \mathsf{sstb}\}$.

2.2 Necessary Notation

This notation is meant to keep the definitions of explanations in Section 3 general and short.

Notation 1. Let $\mathcal{AF} = \langle \mathsf{Args}, \mathsf{Att} \rangle$ be an AF, $A \in \mathsf{Args}$ and $\mathsf{S} \subseteq \mathsf{Args}$. Then, for some $\mathsf{sem} \in \{\mathsf{grd}, \mathsf{cmp}, \mathsf{prf}, \mathsf{sstb}\}$:

- $\mathfrak{E}^{\mathsf{sem}}_{A} = \{\mathcal{E} \in \mathsf{Ext}_{\mathsf{sem}}(\mathcal{AF}) \mid A \in \mathcal{E}\}$ denotes the set of sem-extensions of $\mathcal{AF}$ which contain A;
- $\mathfrak{E}^{\mathsf{sem}}_{\not A} = \{\mathcal{E} \in \mathsf{Ext}_{\mathsf{sem}}(\mathcal{AF}) \mid A \notin \mathcal{E}\}$ denotes the set of sem-extensions of $\mathcal{AF}$ which do not contain A.

The set of arguments that can be used to explain the acceptance of a formula differs depending on the acceptance strategy. For this the following notation will be applied.

Notation 2. Let $\mathcal{AF}(\mathrm{AT}) = \langle \mathsf{Args}, \mathsf{Att} \rangle$ be an SAF, $\phi \in \mathcal{L}$ and let $\mathsf{sem} \in \{\mathsf{grd}, \mathsf{cmp}, \mathsf{prf}, \mathsf{sstb}\}$. Then:

- $\mathsf{Args}_{\phi} = \{A \in \mathsf{Args} \mid \mathsf{Conc}(A) = \phi\}$ denotes the set of all arguments of $\mathcal{AF}(\mathrm{AT})$ with conclusion ϕ;
- $\mathsf{Args}^{\mathsf{sem},\cup}_{\phi} = \{A \in \bigcup \mathsf{Ext}_{\mathsf{sem}}(\mathcal{AF}(\mathrm{AT})) \mid \mathsf{Conc}(A) = \phi\}$ denotes the set of all arguments of $\mathcal{AF}(\mathrm{AT})$ with conclusion ϕ that are part of at least one sem-extension (i.e., that are credulously accepted);
- $\mathsf{Args}^{\mathsf{sem},\cap}_{\phi} = \begin{cases} \emptyset & \text{if } \mathrm{AT} \not\mid\!\sim^{\cap}_{\mathsf{sem}} \phi \\ \mathsf{Args}^{\mathsf{sem},\cup}_{\phi} & \text{otherwise} \end{cases}$
is the same as $\mathsf{Args}^{\mathsf{sem},\cup}_{\phi}$ if ϕ is skeptically accepted and $\emptyset$ if it is not skeptically accepted.

Example 6. (Example 4 continued) Whenever $\mathsf{Args}^{\mathsf{sem},\cap}_{p} \neq \emptyset$, there is no difference between $\cup$ and $\cap$. But $\mathsf{Args}_{q} = \mathsf{Args}^{\mathsf{sem},\cup}_{q} = \{A_3\}$ while $\mathsf{Args}^{\mathsf{sem},\cap}_{q} = \emptyset$ for $\mathsf{sem} \in \{\mathsf{cmp}, \mathsf{prf}, \mathsf{sstb}\}$.

Next it is defined what it means for two formulas to be connected in an argumentation system.

Definition 9. Let $\mathrm{AS} = \langle \mathcal{L}, \mathcal{R}, n \rangle$ be an argumentation system. Then, ϕ is *connected to* ψ if $\phi = \psi$, or:

- there is some $r \in \mathcal{R}$ with $\mathsf{Cons}(r) = \psi$ and $\phi \in \mathsf{Ant}(r)$;
- there is some $\gamma \in \mathcal{L}$ such that ϕ is connected to γ and γ is connected to ψ.

The set of all connected formulas of ψ is denoted by:

- $\mathsf{Connected}(\psi) = \{\phi \in \mathcal{L} \mid \phi \text{ is connected to } \psi\}$.

In explanations for formulas for which no argument exists the following notation will be used:

Notation 3. Let $\mathcal{AF}(\mathrm{AT}) = \langle \mathsf{Args}, \mathsf{Att} \rangle$ be an SAF and let $\phi \in \mathcal{L}$ be such that there is no argument for it in Args. Then:

- $\mathsf{NoArgAnt}(\phi) = \{\psi \mid \psi \in \bigcup\{\mathsf{Ant}(r) \mid r \in \mathsf{Rules}(\mathcal{R}, \phi)\} \text{ and } \nexists A \in \mathsf{Args} \text{ s.t. } \mathsf{Conc}(A) = \psi\}$ denotes the set of formulas in antecedents of rules for ϕ for which no argument exists.
- $\mathsf{NoArgPrem}(\phi) = \{\psi \in \mathsf{Connected}(\phi) \mid \mathsf{Rules}(\mathcal{R}, \psi) = \emptyset \text{ and } \psi \notin \mathcal{K}\}$ denotes the set of formulas that are connected to ϕ but that are not part of $\mathcal{K}$ and for which no rules exist.

Intuitively, $\mathsf{NoArgAnt}$ determines the formulas for which arguments are missing in order for an argument for ϕ to be available, while $\mathsf{NoArgPrem}$ determines the formulas that are not derivable from $\mathcal{AF}(\mathrm{AT})$ (neither from $\mathcal{K}$ nor as a conclusion of some rule) and which could be part of the derivation of an argument for ϕ.

Example 7. Consider AS_2 from Example 3, but let $\mathcal{K}_2' = \mathcal{K}_p^2$ (i.e., $\mathcal{K}_n^2 = \emptyset$). It follows that the arguments A_1, A_2, A_3 and A_4 no longer exist. Thus there is no argument for $\neg r$ nor for q (though there is still an argument for p: B_2). We have that: $\mathsf{NoArgAnt}(q) = \{t, \neg r\}$, $\mathsf{Connected}(q) = \{t, \neg r\}$ and $\mathsf{NoArgPrem}(q) = \{t\}$.

3 Basic Explanations

We now define basic explanations in terms of two functions. $\mathbb{D}$ determines the *depth* of the explanation, how "far away" we should look when considering attacking and defending arguments as explanations. $\mathbb{F}$ determines the *form* of the explanation, whether we want, for example, an argument as an explanation or only its premises. A formal definition of these functions is not provided since domain ($\mathbb{F}$) and codomain ($\mathbb{D}$ and $\mathbb{F}$) are not fixed. We will sometimes use the superscriptes acc and na to denote the function used in the context of acceptance [resp. non-acceptance] explanations.

See Appendix A for an algorithm that computes the basic explanations.

3.1 Basic Explanations for Acceptance

We define two types of acceptance explanations, where $\cap$-explanations provide all the reasons why an argument or formula can be accepted by a skeptical reasoner, while $\cup$-explanations provide one reason why an argument or formula can be accepted by a credulous reasoner. For the purpose of this section let $\mathbb{D}^{\mathsf{acc}}(A, \mathsf{S}) = \mathsf{DefBy}(A, \mathsf{S})$ and $\mathbb{F}^{\mathsf{acc}}(\mathsf{T}) = \mathsf{id}(\mathsf{T}) = \mathsf{T}$ (i.e., $\mathsf{id}(\mathsf{S}) = \mathsf{S}$ for any set S).

3.1.1 Explanations for Accepted Arguments

An argument explanation for an accepted argument A consists of the arguments that defend it, depending on the extensions considered according to the acceptability strategy.

Definition 10 (Argument explanation)**.** Let $\mathcal{AF} = \langle \mathsf{Args}, \mathsf{Att} \rangle$ be an AF and let $A \in \mathsf{Args}$ be an accepted argument, given some $\mathsf{sem} \in \{\mathsf{cmp}, \mathsf{grd}, \mathsf{prf}, \mathsf{sstb}\}$ and an acceptance strategy ($\cap$ or $\cup$). Then:

$$\mathsf{Acc}^{\cap}_{\mathsf{sem}}(A) = \bigcup_{\mathcal{E} \in \mathsf{Ext}_{\mathsf{sem}}(\mathcal{AF})} \mathbb{D}^{\mathsf{acc}}(A, \mathcal{E});$$
$$\mathsf{Acc}^{\cup}_{\mathsf{sem}}(A) \in \{\mathbb{D}^{\mathsf{acc}}(A, \mathcal{E}) \mid \mathcal{E} \in \mathfrak{E}^{\mathsf{sem}}_{A}\}.$$

$\mathsf{Acc}^{\cap}_{\mathsf{sem}}(A)$ provides for each sem-extension $\mathcal{E}$ the arguments that defend A in $\mathcal{E}$, and $\mathsf{Acc}^{\cup}_{\mathsf{sem}}(A)$ the arguments that defend A in one of the sem-extensions.

Example 8 (Example 2 continued). Recall $\mathcal{AF}_1 = \langle \mathsf{Args}_1, \mathsf{Att}_1 \rangle$. We have that:

- $\mathsf{Acc}^{\cup}_{\mathsf{prf}}(A_2) = \{A_4\}$;
- $\mathsf{Acc}^{\cup}_{\mathsf{prf}}(A_3) = \{A_3\}$.

3.1.2 Explanations for Accepted Formulas

In structured argumentation explanations for the acceptance of a formula ϕ can be requested, in addition to argument explanations. For ϕ to be accepted, at least one argument for ϕ must exist. Therefore, the existence of such an argument is part of the explanation as well.

Definition 11 (Formula explanation)**.** Let $\mathcal{AF}(\mathrm{AT}) = \langle \mathsf{Args}, \mathsf{Att} \rangle$ be an SAF and let $\phi \in \mathcal{L}$ be such that $\mathrm{AT} \mid\!\sim^{\star}_{\mathsf{sem}} \phi$, for $\mathsf{sem} \in \{\mathsf{cmp}, \mathsf{grd}, \mathsf{prf}, \mathsf{sstb}\}$ and $\star \in \{\cap, \cup\}$. Here $\mathsf{S} = \mathsf{Args}^{\mathsf{sem},\cap}_{\phi}$, $A \in \mathsf{Args}^{\mathsf{sem},\cup}_{\phi}$ and $\mathsf{S}_A \in \{\mathbb{D}^{\mathsf{acc}}(A, \mathcal{E}) \mid \mathcal{E} \in \mathfrak{E}^{\mathsf{sem}}_{A}\}$:

$$\mathsf{Acc}^{\cap}_{\mathsf{sem}}(\phi) = \left\langle \mathbb{F}^{\mathsf{acc}}(\mathsf{S}),\ \mathbb{F}^{\mathsf{acc}}\left(\bigcup_{B \in \mathsf{S}} \bigcup_{\mathcal{E} \in \mathfrak{E}^{\mathsf{sem}}_{B}} \mathbb{D}^{\mathsf{acc}}(B, \mathcal{E}) \right) \right\rangle;$$

$$\mathsf{Acc}^{\cup}_{\mathsf{sem}}(\phi) = \left\langle \mathbb{F}^{\mathsf{acc}}(A),\ \mathbb{F}^{\mathsf{acc}}\left(\mathsf{S}_A\right) \right\rangle.$$

The first part of the explanation denotes arguments for ϕ (recall Notation 2) – all arguments in the case of $\mathsf{Acc}^{\cap}_{\mathsf{sem}}(\phi)$ and one argument in the case of $\mathsf{Acc}^{\cup}_{\mathsf{sem}}(\phi)$. The second part of the explanation is similar to the set of arguments in an argument explanation, although now the function $\mathbb{F}$ is applied to it. This makes it possible to change the form of the explanation (e.g., premises instead of arguments). The main difference with argument explanations is that more than one argument for ϕ may be considered in the $\cap$-explanation. The (skeptical) $\cap$-explanation again takes all extensions in $\mathfrak{E}^{\mathsf{sem}}_{B}$ into account to determine the arguments that defend B, while for the (credulous) $\cup$-explanation again the defending arguments for A from just one extension in $\mathfrak{E}^{\mathsf{sem}}_{A}$ are taken.

Example 9. (Example 5 continued) Consider the SAF $\mathcal{AF}(\mathrm{AT}_2)$ for $\mathrm{AT}_2 = \langle \mathrm{AS}_2, \mathcal{K}_2 \rangle$. Recall that $\mathrm{AT}_2 \mid\!\sim^{\cap}_{\mathsf{prf}} p$, hence:

- $\mathsf{Acc}^{\cap}_{\mathsf{prf}}(p) = \langle \{A_4, B_2\}, \{A_2, A_3, B_1\} \rangle$.

For other formulas the $\mathsf{Acc}^{\cap}_{\mathsf{sem}}$-explanation does not apply, since none of these are skeptically accepted. However:

- $\mathsf{Acc}^{\cup}_{\mathsf{prf}}(q) = \langle \{A_3\}, \{A_2, A_3\} \rangle$;
- $\mathsf{Acc}^{\cup}_{\mathsf{prf}}(\neg q) = \langle \{B_3\}, \{B_1, B_3\} \rangle$.

3.2 Basic Explanations for Non-Acceptance

Similar to acceptance explanations, there are two types of non-acceptance explanations: $\cap$-explanations for why an argument or formula is not accepted in some extensions (i.e., is not skeptically accepted), and $\cup$-explanations for why an argument or formula is not accepted in all extensions (i.e., is not credulously accepted). For this let $\mathbb{D}^{\mathsf{na}}(A, \mathsf{S}) = \mathsf{NotDef}(A, \mathsf{S})$ and $\mathbb{F}^{\mathsf{na}}(\mathsf{T}) = \mathsf{id}(\mathsf{T}) = \mathsf{T}$.

3.2.1 Explanations for Non-Accepted Arguments

In any Dung-style semantics based on the complete semantics, an argument is not accepted if it is attacked and it is not defended by an accepted argument. Hence, intuitively, the explanation for the non-acceptance of an argument is the set of arguments for which no defense exists.

Definition 12 (Non-acceptance argument explanation)**.** Let $\mathcal{AF} = \langle \mathsf{Args}, \mathsf{Att} \rangle$ be an AF and let $A \in \mathsf{Args}$ be an argument that is not accepted, given some $\mathsf{sem} \in \{\mathsf{cmp}, \mathsf{grd}, \mathsf{prf}, \mathsf{sstb}\}$ and some $\star \in \{\cap, \cup\}$. Then:

$$\mathsf{NotAcc}^{\cap}_{\mathsf{sem}}(A) = \bigcup_{\mathcal{E} \in \mathfrak{E}^{\mathsf{sem}}_{\cancel{A}}} \mathbb{D}^{\mathsf{na}}(A, \mathcal{E});$$

$$\mathsf{NotAcc}^{\cup}_{\mathsf{sem}}(A) = \bigcup_{\mathcal{E} \in \mathsf{Ext}_{\mathsf{sem}}(\mathcal{AF})} \mathbb{D}^{\mathsf{na}}(A, \mathcal{E}).$$

So the non-acceptance argument explanation contains all the arguments in Args that attack A and for which no defense exists in: some sem-extensions (for $\cap$) of which A is not a member; all sem-extensions (for $\cup$). That for $\cap$ only some extensions have to be considered follows since A is skeptically non-accepted as soon as $\mathfrak{E}^{\mathsf{sem}}_{\cancel{A}} \neq \emptyset$, while A is credulously non-accepted when $\mathfrak{E}^{\mathsf{sem}}_{\cancel{A}} = \mathsf{Ext}_{\mathsf{sem}}(\mathcal{AF})$.

Example 10. (Example 5 continued) Recall $\mathcal{AF}(\text{AT}_2)$. Then:

- $\mathsf{NotAcc}^{\cup}_{\mathsf{grd}}(A_3) = \{B_1, B_3\}$;
- $\mathsf{NotAcc}^{\cup}_{\mathsf{prf}}(B_3) = \{A_2, A_3\}$.

3.2.2 Explanations for Non-Accepted Formulas

The non-acceptance of a formula ϕ can have two causes: either there is no argument for ϕ at all (i.e., it is not derivable) or all arguments for ϕ are attacked. In the first case ϕ is not part of the knowledge base $\mathcal{K}$. Moreover, if there are rules with ϕ as consequent, for each rule there is at least one antecedent for which no argument exists.

Definition 13 (Non-derivability explanation)**.** Let $\mathcal{AF}(\text{AT})$ be an SAF and let ϕ be some non-derivable formula. Then:

$$\begin{aligned}\mathsf{NotDer}(\phi) = \ & \langle \mathsf{Rules}(\mathcal{R}, \phi), \\ & \mathsf{NoArgAnt}(\phi), \mathsf{NoArgPrem}(\phi) \rangle .\end{aligned}$$

The idea is that the explanation points out the gaps in the argumentation theory: the missing knowledge base elements and/or missing rules. If there are rules for ϕ these are collected in the first part of the explanation, the second part contains the missing antecedents of these rules (if there would be arguments for all antecedents, there would be an argument for ϕ) and the third part contains the formulas that are connected to ϕ but for which no rule exists (i.e., formulas which are neither part of the knowledge base nor the consequent of a rule).

Example 11. (Example 7 continued) Consider again AS_2 from Example 3, with the knowledge base $\mathcal{K}_2'$ from Example 7 (i.e., $\mathcal{K}_2' = \mathcal{K}_2 \setminus \{t\}$). There are no arguments for $\neg r$ and q:

- $\mathsf{NotDer}(\neg r) = \langle \{d_3\}, \{t\}, \{t\} \rangle$;
- $\mathsf{NotDer}(q) = \langle \{d_4\}, \{t, \neg r\}, \{t\} \rangle$.

This follows since, although there is a rule for q (i.e., $d_4 \in \mathcal{R}_d^2$) [resp. for $\neg r$ (i.e., $d_3 \in \mathcal{R}_d^2$)], there is some $\psi \in \mathsf{Ant}(d_4)$ [resp. $\psi \in \mathsf{Ant}(d_3)$] (i.e., $\psi = t$ [resp. $\psi = \neg r$]) such that there is no argument for t [resp. $\neg r$] in $\mathcal{AF}(\text{AT}_2)$ and when looking at the missing premises to derive q [resp. $\neg r$] the formula t, necessary for d_3 is found.

Like for non-acceptance argument explanations, if an argument for ϕ exists but it is not accepted, there has to be an attacker for which there is no defense.

Definition 14 (Non-acceptance formula explanation)**.** Let $\mathcal{AF}(\mathrm{AT}) = \langle \mathsf{Args}, \mathsf{Att} \rangle$ be an SAF and let $\phi \in \mathcal{L}$ be such that $\mathrm{AT} \not\mathrel{|\!\!\sim}^{\star}_{\mathsf{sem}} \phi$, given some $\mathsf{sem} \in \{\mathsf{cmp}, \mathsf{grd}, \mathsf{prf}, \mathsf{sstb}\}$ and $\star \in \{\cap, \cup\}$. Here, $\mathsf{S}_\phi = \mathsf{Args}_\phi$.

$$\mathsf{NotAcc}^{\cap}_{\mathsf{sem}}(\phi) = \left\langle \mathbb{F}^{\mathsf{na}}(\mathsf{S}_\phi), \mathbb{F}^{\mathsf{na}} \left(\bigcup_{A \in \mathsf{S}_\phi} \bigcup_{\mathcal{E} \in \mathfrak{E}^{\mathsf{sem}}_{\not A}} \mathbb{D}^{\mathsf{na}}(A, \mathcal{E}) \right) \right\rangle$$

$$\mathsf{NotAcc}^{\cup}_{\mathsf{sem}}(\phi) = \left\langle \mathbb{F}^{\mathsf{na}}(\mathsf{S}_\phi), \mathbb{F}^{\mathsf{na}} \left(\bigcup_{A \in \mathsf{S}_\phi} \bigcup_{\mathcal{E} \in \mathsf{Ext}_{\mathsf{sem}}(\mathcal{AF})} \mathbb{D}^{\mathsf{na}}(A, \mathcal{E}) \right\rangle \right).$$

These explanations consist of the existing arguments for ϕ and the arguments for which no defense exists from $\mathcal{E}$ under $\mathbb{D}^{\mathsf{na}}$. Similar to non-acceptance argument explanations, for $\cap$ only the extensions without any argument for ϕ have to be considered, while for $\cup$ all extensions have to be accounted for. By assumption $\mathsf{S}_\phi \neq \emptyset$, since otherwise the explanation for the non-acceptance of ϕ would be its non-derivability.

Example 12. (Example 9 continued) Consider again $\mathcal{AF}(\mathrm{AT}_2)$. Recall that all arguments are credulously accepted, we do however have:

- $\mathsf{NotAcc}^{\cup}_{\mathsf{prf}}(q) = \langle \{A_3\}, \{B_1, B_3\} \rangle$;
- $\mathsf{NotAcc}^{\cup}_{\mathsf{prf}}(\neg q) = \langle \{B_3\}, \{A_2, A_3\} \rangle$.

4 Varying $\mathbb{D}$ and $\mathbb{F}$

This section proposes several variations for $\mathbb{D}$ and $\mathbb{F}$, the main purpose of which is to show the flexibility of the basic framework. We focus on notions of defense, which are suitable for the completeness-based semantics in this paper. For, for example, naive semantics, one might want to base $\mathbb{D}$ on conflicts instead. In Section 4.4 these variations are discussed in the context of a real-life application.

4.1 Notions of Defense

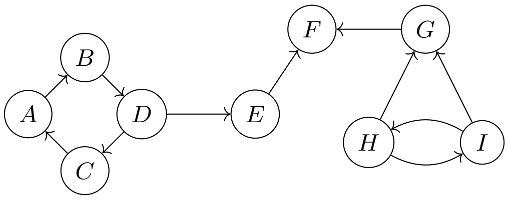
We start by only considering the arguments that defend themselves against all attacks.

Definition 15. Let $\mathcal{AF} = \langle \mathsf{Args}, \mathsf{Att} \rangle$ be an AF, $A, B \in \mathsf{Args}$ and let $\mathcal{E} \in \mathsf{Ext}_{\mathsf{sem}}(\mathcal{AF})$ for some semantics sem. Then:

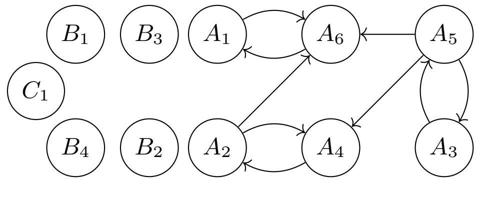
$$\begin{aligned} &\mathsf{FinalDef}(A, \mathcal{E}) = \{B \in \mathsf{DefBy}(A, \mathcal{E}) \mid \forall C \in \mathsf{Args} \text{ s.t. } (C, B) \\ &\in \mathsf{Att}, (B, C) \in \mathsf{Att}\} \cup \bigcup \{\mathsf{DefBy}(B, \mathcal{E}) \mid B \in \mathsf{DefBy}(A, \mathcal{E}), \\ &\forall C \in \mathsf{DefBy}(B, \mathcal{E}), \mathsf{DefBy}(C, \mathcal{E}) = \mathsf{DefBy}(B, \mathcal{E}) \text{ and } \nexists D \\ &\in \mathsf{DefBy}(B, \mathcal{E}) \text{ s.t. } \forall E \in \mathsf{Args} \text{ s.t. } (E, D) \in \mathsf{Att}, (D, E) \in \mathsf{Att}\} \end{aligned}$$

denotes the set of arguments that defend A in $\mathcal{E}$ and that are not attacked at all, defend themselves against any attacker or are part of an even cycle that is not attacked.

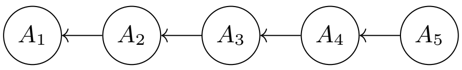
Intuitively this means that these arguments that defend A do not need other arguments to be defended and, given $\mathcal{E}$, can be considered as safe to be accepted. To see why even cycles should be regarded, take a look at the following example:



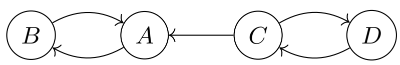
(a) $\mathcal{AF}_3$, Example 13.



(b) $\mathcal{AF}(\mathrm{AT}_6)$, Example 19.



(c) $\mathcal{AF}_4$, Example 14.



(d) $\mathcal{AF}'_5$, Example 16.

Figure 2: Graphical representations of the AFs in Section 4.

Example 13. (Figure 2(a)) Note that $\mathsf{Ext}_{\mathsf{grd}}(\mathcal{AF}_3) = \emptyset$, while $\mathsf{Ext}_{\mathsf{sem}}(\mathcal{AF}_3) = \{\{A, D, F, H\}, \{A, D, F, I\}, \{B, C, E, H\}, \{B, C, E, I\}\}$ for $\mathsf{sem} \in \{\mathsf{prf}, \mathsf{sstb}\}$. Let $\mathcal{E} = \{A, D, F, H\}$. Then $\mathsf{FinalDef}(F, \mathcal{E}) = \{A, D, H\}$. This follows since H defends itself against the attack from I and $\{A, D\}$ is part of an even cycle that is not attacked. If even cycles would not be covered by $\mathsf{FinalDef}$, the defense of the attack (E, F) would not be accounted for.

Another option is to consider only the arguments that directly defend the considered argument.

Definition 16. Let $\mathcal{AF} = \langle \mathsf{Args}, \mathsf{Att} \rangle$ be an AF, $A, B \in \mathsf{Args}$ and let $\mathcal{E} \in \mathsf{Ext}_{\mathsf{sem}}(\mathcal{AF})$ for some semantics sem. Then: $\mathsf{DirDef}(A, \mathcal{E}) = \{B \in \mathcal{E} \mid B \text{ directly defends } A\}$, denotes the set of arguments in $\mathcal{E}$ that *directly* defend A.

One reason for looking at direct conflicts might be that direct conflicts are often more clear from the context than indirect conflicts.

Example 14. (Figure 2(c)). Here $\mathsf{Ext}_{\mathsf{sem}}(\mathcal{AF}_4) = \{\{A_1, A_3, A_5\}\}$ for any $\mathsf{sem} \in \{\mathsf{grd}, \mathsf{cmp}, \mathsf{prf}, \mathsf{sstb}\}$. Moreover:

- $\mathsf{Acc}(A_1) = \{A_3, A_5\}$ for $\mathbb{D} = \mathsf{DefBy}$;
- $\mathsf{Acc}(A_1) = \{A_5\}$ for $\mathbb{D} = \mathsf{FinalDef}$; and
- $\mathsf{Acc}(A_1) = \{A_3\}$ for $\mathbb{D} = \mathsf{DirDef}$.

This minimal example can be seen as a discussion in the form of a sequence of arguments attacking and defending the topic A_1. When at the end an explanation for the acceptance of A_1 is requested: DefBy returns all arguments that defend A_1; $\mathsf{FinalDef}$ returns the last argument that was put forward, which is uncontested; and DirDef returns the argument against the direct attacker of the topic.

Example 15. (Example 9 continued) Consider $\mathcal{AF}(\mathrm{AT}_2)$. Then, for $\mathbb{F}^{\mathsf{acc}} = \mathsf{id}$:

- $\mathsf{Acc}^{\cap}_{\mathsf{prf}}(p) = \langle \{A_4, B_2\}, \{A_2, A_3, B_1\} \rangle$, for $\mathbb{D}^{\mathsf{acc}} = \mathsf{DirDef}$;
- $\mathsf{Acc}^{\cap}_{\mathsf{prf}}(p) = \langle \{A_4, B_2\}, \{A_2, B_1\} \rangle$, for $\mathbb{D}^{\mathsf{acc}} = \mathsf{FinalDef}$.

In the case of non-acceptance explanations, $\mathbb{D}$ was defined as the set of all attacking arguments against which no defense exists. The next definition considers only those attackers that A does not (in)directly attack itself.

Definition 17. Let $\mathcal{AF} = \langle \mathsf{Args}, \mathsf{Att} \rangle$ be an AF, $A, B \in \mathsf{Args}$ and let $\mathcal{E} \in \mathsf{Ext}_{\mathsf{sem}}(\mathcal{AF})$ be an extension for some semantics sem. Then: $\mathsf{NoDir}(A, \mathcal{E}) = \{B \in \mathsf{NotDef}(A, \mathcal{E}) \mid A \text{ does not (in)directly attack } B\}$ denotes the set of arguments that attack A for which no defense exists in $\mathcal{E}$ and which are not attacked by A itself.

Intuitively, the members of $\mathsf{NoDir}(A, \mathcal{E})$ attack A but in order to defend A against the attack another argument than A itself is necessary.

Example 16. Let $\mathcal{AF}_5 = \langle \{A, B\}, \{(A, B), (B, A)\} \rangle$. Here $\mathsf{Ext}_{\mathsf{prf}}(\mathcal{AF}_5) = \{\{A\}, \{B\}\}$, $\mathsf{NotAcc}^{\cap}(A) = \{B\}$ for $\mathbb{D} = \mathsf{NotDef}$ but $\mathsf{NotAcc}^{\cap}_{\mathsf{prf}}(A) = \emptyset$ for $\mathbb{D} = \mathsf{NoDir}$ since by accepting A, A can indeed be concluded. Now let $\mathcal{AF}'_5$ as in Figure 2(d). Then $\mathsf{Ext}_{\mathsf{prf}}(\mathcal{AF}'_5) = \{\{A, D\}, \{B, C\}, \{B, D\}\}$, $\mathsf{NotAcc}^{\cap}_{\mathsf{prf}}(A) = \{B, C\}$ for $\mathbb{D} = \mathsf{NotDef}$ and $\mathsf{NotAcc}^{\cap}_{\mathsf{prf}}(A) = \{C\}$ for $\mathbb{D} = \mathsf{NoDir}$, since in order to defend A, just accepting A is not enough, D is needed to defend against the attack from C.

Example 17. (Example 12 continued) Consider $\mathcal{AF}(\mathrm{AT}_2)$ from Example 3. Then, for $\mathbb{F}^{\mathsf{acc}} = \mathsf{id}$ and $\mathbb{D}^{\mathsf{na}} = \mathsf{NoDir}$:

- $\mathsf{NotAcc}^{\cap}_{\mathsf{prf}}(q) = \langle \{A_3\}, \{B_1\} \rangle$;
- $\mathsf{NotAcc}^{\cap}_{\mathsf{prf}}(\neg q) = \langle \{B_3\}, \{A_2\} \rangle$.

4.2 Element Explanations

In structured argumentation, one can provide full arguments as the explanation (e.g., $\mathbb{F} = \mathsf{id}$), but the structure of the arguments provides other possibilities as well.

Definition 18. Let $\mathcal{AF}(\mathrm{AT}) = \langle \mathsf{Args}, \mathsf{Att} \rangle$ be an SAF and $\mathsf{S} \subseteq \mathsf{Args}$ a set of formulas. Then $\mathsf{AntTop}(\mathsf{S}) = \{\mathsf{Ant}(\mathsf{TopRule}(A)) \mid A \in \mathsf{S}\}$ denotes the set of antecedents of the top rule of all arguments in S.

The above definition, combined with the introduced notation in Definition 5, provides some ideas of how $\mathbb{F}$ can be defined. For example, explanations in terms of premises explain the conclusion in terms of knowledge base items. The notion AntTop provides explanations in terms of closely related information and the rule with which the conclusion is derived from that information.

Example 18. (Examples 9 and 12 continued) Consider $\mathcal{AF}(\mathrm{AT}_2)$ from Example 3. Then, for $\mathbb{D}^{\mathsf{acc}} = \mathsf{DefBy}$ and $\mathbb{D}^{\mathsf{na}} = \mathsf{NotDef}$:

- $\mathsf{Acc}^{\cap}_{\mathsf{prf}}(p) = \langle \{t, r\}, \{t, r\} \rangle$ for $\mathbb{F}^{\mathsf{acc}} = \mathsf{Prem}$;
- $\mathsf{Acc}^{\cap}_{\mathsf{prf}}(p) = \langle \{q, r\}, \{t, \neg r\} \rangle$ for $\mathbb{F}^{\mathsf{acc}} = \mathsf{AntTop}$;
- $\mathsf{NotAcc}^{\cup}_{\mathsf{prf}}(q) = \langle \{t\}, \{r\} \rangle$ for $\mathbb{F}^{\mathsf{na}} = \mathsf{Prem}$;
- $\mathsf{NotAcc}^{\cup}_{\mathsf{prf}}(q) = \langle \{\neg r, t\}, \{r\} \rangle$ for $\mathbb{F}^{\mathsf{na}} = \mathsf{AntTop}$.

4.3 Comparing the Size of Explanations

When choosing a definition for $\mathbb{D}$ and $\mathbb{F}$ the size of the resulting explanation might be one of the considerations. While for $\mathbb{F}$ this depends on the AF (e.g., an argument might have many premises or the top rule might have only one antecedent), for $\mathbb{D}$ the size of the different definitions can be compared. We will apply $\leq$ to the size of the sets, i.e., $\mathsf{S}_1 \leq \mathsf{S}_2$ denotes $|\mathsf{S}_1| \leq |\mathsf{S}_2|$.

Proposition 1. *Let $\mathcal{AF} = \langle \mathsf{Args}, \mathsf{Att} \rangle$ be an AF, let $A \in \mathsf{Args}$ and let $\mathcal{E} \in \mathsf{Ext}_{\mathsf{sem}}(\mathcal{AF})$ be an extension for it. Where $\preceq \in \{\leq, \subseteq\}$:*

1. $\mathsf{DirDef}(A, \mathcal{E}) \preceq \mathsf{DefBy}(A, \mathcal{E})$;
2. $\mathsf{FinalDef}(A, \mathcal{E}) \preceq \mathsf{DefBy}(A, \mathcal{E})$;
3. $\mathsf{NoDir}(A, \mathcal{E}) \preceq \mathsf{NotDef}(A, \mathcal{E})$.

This follows since $\mathsf{DirDef}(A, \mathcal{E})$ and $\mathsf{FinalDef}(A, \mathcal{E})$ are always subsets of $\mathsf{DefBy}(A, \mathcal{E})$ and $\mathsf{NoDir}(A, \mathcal{E})$ is always a subset of $\mathsf{NotDef}(A, \mathcal{E})$. Indeed, $\mathsf{Acc}^{\cap}_{\mathsf{prf}}(p)$ is both $\leq$- and $\subseteq$-smaller for $\mathbb{D}^{\mathsf{acc}} = \mathsf{DirDef}$ than for $\mathbb{D}^{\mathsf{acc}} = \mathsf{DefBy}$ (see Example 15). Similarly, $\mathsf{NotAcc}^{\cap}_{\mathsf{prf}}(q)$, is $\leq$- and $\subseteq$-smaller for $\mathbb{D}^{\mathsf{na}} = \mathsf{NoDir}$ than for $\mathbb{D}^{\mathsf{na}} = \mathsf{NotDef}$ (see Example 17).

4.4 Applying the Basic Framework

One of the inspirations for this paper is an argumentation-based system in use by the Dutch National Police, which assists citizens who might have been the victim of internet trade fraud (e.g., malicious web shops or traders) in filing a criminal report [1, 10]. From this report basic observations such as 'money was paid by the complainant to the counterparty' or 'no package was delivered to the complainant' are collected, and these observations are used as premises in legal arguments to infer whether or not the report concerns a possible case of fraud. This conclusion is then provided to the complainant who filed the report. The system is based on ASPIC$^+$ [11], with axioms (the observations) and defeasible rules (based on Dutch law concerning fraud), and all attacks are rebuts. The next example illustrates such an argumentation framework.

Example 19. Let $\mathrm{AS}_6 = \langle \mathcal{L}_6, \mathcal{R}_6, n \rangle$ be an argumentation system, where $\mathcal{L}_6$ contains the propositions p (the complainant *paid*), w (the *wrong* package arrived), fk (the product is *fake*), su (the product looks *suspicious*), re (counterparty states that the product is *real*), cd (the *complainant delivered*), cpd (the *counterparty delivered*) and f (it is *fraud*) and their negations and where $\mathcal{R}_6$ is such that the following arguments can be derived from $\mathcal{K}_6 = \mathcal{K}_n^6 = \{p, w, su, re\}$:

$$
\begin{array}{lll}
B_1 : p & C_1 : B_1 \Rightarrow cd & \\
B_2 : w & A_1 : B_2 \Rightarrow \neg f & A_4 : A_3 \Rightarrow \neg cpd \\
B_3 : su & A_2 : B_2 \Rightarrow cpd & A_5 : B_4 \Rightarrow \neg fk \\
B_4 : re & A_3 : B_3 \Rightarrow fk & A_6 : C_1, A_4 \Rightarrow f
\end{array}
$$

Figure 2(b) shows the corresponding SAF $\mathcal{AF}(\mathrm{AT}_6)$. The preferred extensions of $\mathcal{AF}(\mathrm{AT}_6)$, only mentioning the A arguments, are $\{A_1, A_2, A_3\}$, $\{A_1, A_2, A_5\}$, $\{A_1, A_3, A_4\}$ and $\{A_3, A_4, A_6\}$. None of $A_1, \ldots, A_6$ is skeptically accepted and all are credulously accepted. Take conclusion f, where $\mathcal{E} = \{A_3, A_4, A_6, B_1, B_2, B_3, B_4, C_1\}$. Then:

- $\mathsf{Acc}^{\cup}_{\mathsf{prf}}(f) = \langle \{A_6\}, \{A_3, A_4, A_6\} \rangle$ for $\mathbb{F}^{\mathsf{acc}} = \mathsf{id}$ and $\mathbb{D}^{\mathsf{acc}} \in \{\mathsf{DefBy}, \mathsf{DirDef}\}$;
- $\mathsf{Acc}^{\cup}_{\mathsf{prf}}(f) = \langle \{p, su\}, \{p, su\} \rangle$ for $\mathbb{F}^{\mathsf{acc}} = \mathsf{Prem}$ and $\mathbb{D}^{\mathsf{acc}} \in \{\mathsf{DefBy}, \mathsf{DirDef}\}$;
- $\mathsf{Acc}^{\cup}_{\mathsf{prf}}(f) = \langle \{cd, \neg cpd\}, \{su\} \rangle$ for $\mathbb{F}^{\mathsf{acc}} = \mathsf{AntTop}$ and $\mathbb{D}^{\mathsf{acc}} = \mathsf{FinalDef}$;
- $\mathsf{NotAcc}^{\cup}_{\mathsf{prf}}(\neg f) = \langle \{A_1\}, \{A_3, A_4, A_6\} \rangle$ for $\mathbb{F}^{\mathsf{na}} = \mathsf{id}$ and $\mathbb{D}^{\mathsf{acc}} = \mathsf{NotDef}$;
- $\mathsf{NotAcc}^{\cup}_{\mathsf{prf}}(\neg f) = \langle \{A_1\}, \{A_3, A_4\} \rangle$ for $\mathbb{F}^{\mathsf{na}} = \mathsf{id}$ and $\mathbb{D}^{\mathsf{acc}} = \mathsf{NoDir}$.

Looking at the different possibilities for $\mathbb{F}$, we see that instead of the full arguments we can also return just the premises (observations) of the supporting arguments, so 'f because p and su'. This is what the police system currently does. The reasoning behind this is that citizens understand these more factual observations better than more legal concepts such as delivering under a contract. On the other hand, for the public prosecutor involved in the processing of complaints, an explanation in legal terms – 'f because cd and $\neg cpd$' (based on AntTop) – might make more sense.

For $\mathbb{D}$ there are also different options. For example, $\mathsf{FinalDef}$ returns arguments that do not need other arguments to defend them. That A_3 is such an argument w.r.t. A_6 means that this argument A_3 for fk is the 'main reason' we accept f, that is, without A_3 the conclusion f will never be accepted. With NoDir, no directly conflicting arguments are given (e.g., A_6 which directly conflicts with A_1). This avoids explanations such as '(the argument for) $\neg f$ is not accepted because (there is an argument for) f'.

4.5 Overview

In this section we have considered variations for the functions $\mathbb{D}$ and $\mathbb{F}$. Acceptance explanations can be given in terms of all the defending arguments ($\mathbb{D} = \mathsf{DefBy}$), the arguments that need no further defense ($\mathbb{D} = \mathsf{FinalDef}$), and arguments that defend against direct conflicts ($\mathbb{D} = \mathsf{DirDef}$). Non-acceptance explanations can be given in terms of all the attackers for which no defense exists ($\mathbb{D} = \mathsf{NotDef}$) and those arguments that need to be defended by another argument ($\mathbb{D} = \mathsf{NoDir}$). In a structured setting (e.g., in ASPIC$^+$), the form of these explanations can be varied. We discussed sets of arguments ($\mathbb{F} = \mathsf{id}$), sets of premises/observations ($\mathbb{F} = \mathsf{Prem}$) and sets of antecedents of the last applied rule ($\mathbb{F} = \mathsf{AntTop}$).

5 Related Work

Fan and Toni [5] define relevant explanations for a single topic argument in the form of a new *related admissibility* semantics, and show how explanations can be derived from related admissible sets for abstract argumentation and ABA. A set of arguments is called related admissible if it is admissible and each argument in it defends the topic. An explanation for an argument A (called here *RA-explanation* to avoid confusion) is then defined as a related admissible set of arguments with topic A. In the next proposition we show how RA-explanations can be expressed in our framework.

Proposition 2. *Let* $\mathcal{AF} = \langle \mathsf{Args}, \mathsf{Att} \rangle$ *be an AF and let* $A \in \mathsf{Args}$*. Then* $\{\mathsf{DefBy}(A, \mathcal{E}) \mid \mathcal{E} \in \mathfrak{E}_A^{\mathsf{adm}}\}$ *is the set of all RA-explanations for A.*

Proof. Let $\mathcal{AF} = \langle \mathsf{Args}, \mathsf{Att} \rangle$ be an AF and let $A \in \mathsf{Args}$. Suppose that $\mathfrak{E}_A^{\mathsf{adm}} \neq \emptyset$. Let $\mathsf{S} \in \{\mathsf{DefBy}(A, \mathcal{E}) \mid \mathcal{E} \in \mathfrak{E}_A^{\mathsf{adm}}\}$, we first show that S is related admissible:

S **defends** A**.** This follows by the definition of $\mathsf{S} = \mathsf{DefBy}(A, \mathcal{E})$.

S **is admissible.** Note that $\mathsf{S} \subseteq \mathcal{E}$ for some $\mathcal{E} \in \mathfrak{E}_A^{\mathsf{adm}}$, therefore S is conflict-free. Suppose that there is some $B \in \mathsf{S}$ such that B is not defended against an attack from $C \in \mathsf{Args}$. By definition of DefBy, C (in)directly attacks A. Since $A, B \in \mathcal{E}$, there is some $D \in \mathcal{E}$ such that D defends A and B against C. By assumption, $D \notin \mathsf{S}$. A contradiction with the definition of DefBy. Therefore S defends all of its arguments and is thus admissible.

Now suppose that there is some S' which is an RA-explanation for A but $\mathsf{S}' \notin \{\mathsf{DefBy}(A, \mathcal{E}) \mid \mathcal{E} \in \mathfrak{E}_A^{\mathsf{adm}}\}$. By definition of related admissible sets $A \in \mathsf{S}'$, $\mathsf{S}' \in \mathfrak{E}_A^{\mathsf{adm}}$ and for each $B \in \mathsf{S}'$, $B = A$ or B defends A in S', thus $B \in \mathsf{DefBy}(A, \mathcal{E})$, a contradiction. Hence any RA-explanation for A is in $\{\mathsf{DefBy}(A, \mathcal{E}) \mid \mathcal{E} \in \mathfrak{E}_A^{\mathsf{adm}}\}$. □

This shows that any $\mathsf{Acc}^{\cup}_{\mathsf{adm}}$-explanation is an RA-explanation and that therefore our framework is a more general version of [5].

García et al. [7] study explanations for abstract argumentation and `DELP`. Explanations for a claim are defined as triples of dialectical trees that provide a warrant for the claim, dialectical trees that provide a warrant for the contrary of the claim, and dialectical trees for the claim and its contrary that provide no warrant. This means, on the one hand, that explanations might contain many arguments and, on the other hand, that the receiver of the explanation is expected to understand argumentation and dialectical trees. With real-life applications in mind, we believe that explanations that rely less on the underlying AF and that can be adjusted to the application are more useful. Therefore, in our framework an explanation consists of a set of (parts of) arguments, that could be embedded in a natural language sentence to be presented to a user, as suggested in Section 4.4.

Explanations for non-accepted arguments in abstract argumentation are studied in [6, 13], both of which focus on the structure of the AF and credulous non-acceptance under admissible semantics. Note that we consider skeptical and credulous non-acceptance for several Dung-style semantics. In [6] an explanation consists of either a set of arguments or a set of attacks, the removal of which would make the argument admissible. In structured argumentation it is not always possible to remove exactly one argument (or attack). In the AF of Figure 1, A_3 would become skeptically acceptable for any semantics, if B_1 would be removed. However, when looking at the underlying argumentation theory (recall Example 3), when B_1 is removed, the arguments B_2 and B_3 do no longer exist and thus $\neg q$ is no longer a credulous conclusion. Therefore, in this paper the basic definition for non-accepted arguments is defined in terms of the arguments for which no defense exist and no suggestion is made how to change the AF in order to get the considered argument accepted. In [13], explanations are sub-frameworks, such that the considered argument is credulously non-accepted in that sub-framework and any of its super-frameworks. Though a note was added on the applicability of such explanations in a structured setting, this is not formally investigated in that paper.

Summarizing, our basic framework is (formally) shown to be more general, more flexible and specifically adjustable to the receiver of the explanation. Furthermore, none of the above-mentioned works consider the structure of the arguments when providing explanations.

6 Conclusions and Future Work

We have introduced a generic, flexible basic framework for explanations in structured and abstract argumentation. With this framework, specialized local explanations for the (non-)acceptance of arguments can be given, taking into account credulous and skeptical reasoners.

In future work, we plan to extend our framework with preferences – although showing preferences is sometimes considered less effective when providing explanations [9], the (non-)acceptance of arguments very often depends directly on them, making a preference the direct reason for (not) accepting an argument.

Given our basic framework, we will further study how our explanations formally relate to acceptance strategies and different semantics, and investigate the necessity and sufficiency of arguments and how to implement this in explanations.

Aside from formal investigations, we also want to look at how findings from the social sciences on what good explanations are (see e.g., [12, 9]) can be integrated, and how different types of explanations are evaluated by human users. Important in this respect is that explanations are *contrastive*: while people may ask *why* A*?*, they often mean *why* A *rather than* B*?*, where A is called the *fact* and B is called the *foil*. The goal is then to explain as much of the differences between fact and foil as possible. One of the challenges for an AI system is that the foil is not always explicit. We plan to study contrastive explanations within our framework by combining acceptance and non-acceptance and the knowledge of conflicting arguments and contraries in the case of an implicit foil.

Acknowledgment. This research has been partly funded by the Dutch Ministry of Justice and the Dutch National Police.

References

[1] Floris Bex, Bas Testerink, and Joeri Peters. AI for online criminal complaints: From natural dialogues to structured scenarios. In *Workshop proceedings of Artificial Intelligence for Justice at ECAI 2016*, pages 22–29, 2016.

[2] Phan Minh Dung. On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games. *Artificial Intelligence*, 77(2):321–357, 1995.

[3] Wolfgang Dvořák and Paul E. Dunne. Computational problems in formal argumentation and their complexity. In Pietro Baroni, Dov Gabay, Massimiliano Giacomin, and Leon van der Torre, editors, *Handbook of Formal Argumentation*, pages 631–688. College Publications, 2018.

[4] Lilian Edwards and Michael Veale. Slave to the algorithm: Why a 'right to an explanation' is probably not the remedy you are looking for. *Duke Law & Technology Review*, 16(1):18–84, 2017.

[5] Xiuyi Fan and Francesca Toni. On computing explanations in argumentation. In Blai Bonet and Sven Koenig, editors, *Proceedings of the 29th AAAI Conference on Artificial Intelligence (AAAI'15)*, pages 1496–1502. AAAI Press, 2015.

[6] Xiuyi Fan and Francesca Toni. On explanations for non-acceptable arguments. In Elizabeth Black, Sanjay Modgil, and Nir Oren, editors, *Proceedings of the 3rd International Workshop on Theory and Applications of Formal Argumentation, (TAFA'15)*, LNCS 9524, pages 112–127. Springer, 2015.

[7] Alejandro García, Carlos Chesñevar, Nicolás Rotstein, and Guillermo Simari. Formalizing dialectical explanation support for argument-based reasoning in knowledge-based systems. *Expert Systems with Applications*, 40(8):3233 – 3247, 2013.

[8] Carmen Lacave and Francisco J Diez. A review of explanation methods for heuristic expert systems. *The Knowledge Engineering Review*, 19(2):133–146, 2004.

[9] Tim Miller. Explanation in artificial intelligence: Insights from the social sciences. *Artificial Intelligence*, 267:1 – 38, 2019.

[10] Daphne Odekerken, AnneMarie Borg, and Floris Bex. Estimating stability for efficient argument-based inquiry. In *Proceedings of the 8th International Conference on Computational Models of Argument (COMMA'20)*, volume 326, pages 307–318. IOS Press, 2020.

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

[12] Wojciech Samek, Thomas Wiegand, and Klaus-Robert Müller. Explainable artificial intelligence: Understanding, visualizing and interpreting deep learning models. *arXiv preprint arXiv:1708.08296*, 2017.

[13] Zeynep Saribatur, Johannes Wallner, and Stefan Woltran. Explaining non-acceptability in abstract argumentation. In *Proceedings of the 24th European Conference on Artificial Intelligence (ECAI'20)*, volume 325 of *Frontiers in Artificial Intelligence and Applications*, pages 881–888. IOS Press, 2020.

A Algorithm for the Computation of the Explanations

This appendix contains a discussion on the computation and computational complexity of the basic explanations. In particular, we will provide a polynomial time algorithm that computes the explanations, once the extensions of the argumentation framework are known.

A.1 Some preliminaries

Since an (abstract) argumentation framework (AF) [2] can be seen as a directed graph, one can determine whether argument B is reachable from argument A and if so, what the distance between the two arguments is. If B is reachable from A, it can be determined, based on the distance, whether A (in)directly attacks or (in)directly defends B. These notions will be useful in calculating explanations.

Definition 19. Let $\mathcal{AF} = \langle \mathsf{Args}, \mathsf{Att} \rangle$ be an AF and let $A, B \in \mathsf{Args}$. There is an *attack-path* from A to B if $(A, B) \in \mathsf{Att}$ or there are $C_1, \ldots, C_{n-1} \in \mathsf{Args}$, such that $(A, C_1), (C_1, C_2), \ldots, (C_{n-2}, C_{n-1}), (C_{n-1}, B) \in \mathsf{Att}$ and no attack appears twice in this sequence. It is said that this attack-path has *length* n and is along the attacks $(A, C_1), (C_1, C_2), \ldots, (C_{n-1}, B)$, if $(A, B) \in \mathsf{Att}$, the path has length 1 and if $A = B$ the attack-path has length 0.[1]

Intuitively, if the length of an attack-path between arguments A and B is odd [resp. even], A (in)directly attacks [resp. defends] B.

[1] Note that an attack-path is known as a *trail* in graph theory.

A.2 The algorithm

The main idea of the basic framework for explanations is that the explanations can be calculated from any argumentation framework, when it is know if an argument is accepted and how (i.e., whether an argument is skeptically or credulously (non-)accepted and under what semantics). In order to compute these explanations we introduce an algorithm that can be applied after the acceptance of the arguments in the framework is determined.

Algorithm 1 presents a method to calculate the length of all existing attack-paths in an argumentation framework (Dist) and for each argument A the set of arguments from which a path to A exists ($\mathsf{Reach}(A)$). The algorithm is based on Procedure ReReach (*Recursive Reach*), a depth-first search algorithm.

Note that the run time of the algorithm is finite when Args and Att are finite. This is the case since for each $A \in \mathsf{Args}$ at most all attacks in Att are considered $|\mathsf{Att}|$ times.

Algorithm 1: Computing Reach and Dist.

Result: Reach and Dist are computed.
Given an AF $\mathcal{AF} = \langle \mathsf{Args}, \mathsf{Att} \rangle$;
for $A \in \mathit{Args}$ **do**
    $\mathsf{Reach}(A) = \{A\}$ and $\mathsf{Dist}(A, A) = \{0\}$;
    for $B \in \mathit{Args} \setminus \{A\}$ **do**
        $\mathsf{Dist}(A, B) = \emptyset$;
for $A \in \mathit{Args}$ **do**
    $\mathrm{ReReach}(A, A, 0, \emptyset)$;

Procedure ReReach(A, A', n, S)

$\mathsf{Visited}_0 = \mathsf{S}$;
for $A^\star \in \mathit{Args}$ *s.t.* $(A^\star, A') \in \mathit{Att}$ *and* $(A^\star, A') \notin \mathit{Visited}$ **do**
    $\mathsf{Reach}(A) = \mathsf{Reach}(A) \cup \mathsf{Reach}(A^\star)$;
    $\mathsf{Dist}(A^\star, A) = \mathsf{Dist}(A^\star, A) \cup \{n + 1\}$;
    $\mathsf{Visited} = \mathsf{Visited} \cup \{(A^\star, A')\}$;
    $\mathrm{ReReach}(A,\ A^\star,\ n + 1,\ \mathsf{Visited})$;
    $\mathsf{Visited} = \mathsf{Visited}_0$;

A.3 Complexity results

Algorithm 1 has some desirable properties. In particular, it is sound and complete (i.e., for the requested argument A it finds all the arguments from which A can be reached and the length of the attack-paths between those arguments) and it runs in polynomial time. The latter is useful since it shows that the computational complexity of computing the explanations for a certain semantics is not more complex than computing the acceptance of an argument and/or the extensions under that semantics [3].

Theorem 1 (Soundness and completeness)**.** *Let $\mathcal{AF} = \langle \mathit{Args}, \mathit{Att} \rangle$ be an AF. Then:*

1. *there is an attack-path from A to B of length n iff $n \in \mathsf{Dist}(A, B)$;*
2. *$A \in \mathsf{Reach}(B)$ iff there is an attack-path from A to B;*

In order to show the above theorem, we need some lemmas and propositions. These lemmas and propositions will partially be shown by induction proofs, for which the following remark will be useful.

Remark 1. Let $\mathcal{AF} = \langle \mathsf{Args}, \mathsf{Att} \rangle$ be an AF and let $A, B \in \mathsf{Args}$. It holds that $A = B$ iff there is an *attack-path* from A to B that has length 0. Similarly, $(A, B) \in \mathsf{Att}$ iff there is an attack-path from A to B of length 1.

Lemma 1. *If ReReach*(A, A', n, S) *is called, there is an attack-path from* A' *to* A*, of length* n*, along the attacks in* S*.*

Proof. Suppose that ReReach(A, A', n, S) is called, either at Line 7 of Algorithm 1 or at Line 6 of Procedure ReReach. We proceed by induction on n.

- **If** $n = 0$**:** then ReReach$(A, A, 0, \emptyset)$ is called at Line 7 of Algorithm 1. Since $A = A'$, by Remark 1, there is an attack-path of length 0 from A' to A, without any attacks.
- **If** $n = 1$**:** then ReReach(A, A', n, S) was called at the first iteration of Procedure ReReach. Hence $(A', A) \in \mathsf{Att}$ and $(A', A) \notin \emptyset$. By Remark 1, there is an attack-path of length 1 from A' to A.

Suppose now that the claim holds for n up to $k \geq 1$.

- **If** $n = k + 1$**:** then there is some $B \in \mathsf{Args}$ such that ReReach(A, A', n, S) is called at Line 6 of the call ReReach(A, B, k, S') where $\mathsf{S}' = \mathsf{S} \setminus \{(A', B)\}$). To see that $\mathsf{S}' = \mathsf{S} \setminus \{(A', B)\}$, note that $\mathsf{Visited}$ is updated at Line 5 with (A', B) before ReReach(A, A', n, S) is called and if $(A', B) \in \mathsf{S}'$, the call to ReReach(A, A', n, S) would not be reached.

  By induction hypothesis, there is an attack-path from B to A of length k along the attacks in S'. Since $(A', B) \notin \mathsf{S}'$, it follows that the attack (A', B) was not used in the attack-path from B to A. Therefore, the path from A' to A along (A', B) and the attack-path B to A is an attack-path from A' to A of length $k + 1$ along the attacks in S. □

The next proposition shows that Algorithm 1 is sound.

Proposition 3. *If* $n \in \mathsf{Dist}(A, B)$ *then there is an attack-path from* A *to* B *of length* n*.*

Proof. Suppose that $n \in \mathsf{Dist}(A, B)$, that there is an attack-path from A to B of length n is shown by induction on n:

- **If** $n = 0$**:** then $\mathsf{Dist}(A, B)$ was updated at Line 3 of the algorithm (since at any other place that $\mathsf{Dist}(A, B)$ might be updated, the addition is always more than 0). It follows immediately that $A = B$. Hence there is an attack path from A to B of length 0.
- **If** $n = 1$**:** then $\mathsf{Dist}(A, B)$ was updated at Line 4 of the procedure in the first iteration of the **for**-loop (since in any other iteration $n \neq 0$). It follows that $(A, B) \in \mathsf{Att}$. Hence the attack-path consists of one attack: (A, B). Thus there is an attack-path from A to B of length 1.

Suppose that the proposition holds for values of n up to k, where $k \geq 1$. Then:

- **If** $n = k + 1$**:** then $\mathsf{Dist}(A, B)$ was updated at Line 4 of Procedure ReReach. This is only the case if there is some $C \in \mathsf{Args}$ such that ReReach(B, C, k, S) was called and $(A, C) \in \mathsf{Att}$ such that $(A, C) \notin \mathsf{S}$. By induction hypothesis, there is an attack-path from C to B of length k and by Lemma 1, this attack-path is along the attacks in S. Since $(A, C) \notin \mathsf{S}$ by assumption, the path from A to B via the attack (A, C) and the attack-path from C to B is an attack-path, of length $k + 1$.

This shows that for any n, if $n \in \mathsf{Dist}(A, B)$, there is an attack-path of length n from A to B. □

In the next lemma the relation between the attacks in an attack-path and the attacks in $\mathsf{Visited}$ in the procedure is shown.

Lemma 2. *If there is an attack-path from A to B, along the attacks $(A, C_1), (C_1, C_2), \ldots, (C_{n-1}, B) \in \mathsf{Att}$, for $C_1, \ldots, C_{n-1} \in \mathsf{Args}$, then during the run of the algorithm ReReach$(B, A, n, \{(A, C_1), (C_1, C_2), \ldots, (C_{n-1}, B)\}$ will be called.*

Proof. Suppose that there is an attack-path from A to B, along the attacks $(A, C_1), (C_1, C_2), \ldots, (C_{n-1}, B) \in \mathsf{Att}$, for $C_1, \ldots, C_{n-1} \in \mathsf{Args}$. We proceed by induction on $n \geq 1$. Since $B \in \mathsf{Args}$, ReReach$(B, B, 0, \emptyset)$ will be called at Line 7 of the algorithm.

- **If** $n = 1$**:** then $(A, B) \in \mathsf{Att}$. At this point $\mathsf{Visited}$ is still empty, hence the **for**-loop at Line 2 of Procedure ReReach will be run for A. At Line 5, $\mathsf{Visited}$ becomes $\{(A, B)\}$ and at Line 6, ReReach$(B, A, 1, \{(A, B)\})$ is called.

Suppose the statement holds for values of n up to $k \geq 1$. Then:

- **If** $n = k+1$**:** then there are $C_1, \ldots, C_k$ such that there is an attack-path from A to B along the attacks $(A, C_1), \ldots, (C_k, B)$. And hence there is an attack-path from C_1 to B along the attacks $(C_1, C_2), \ldots, (C_k, B)$ of length k. By induction hypothesis, ReReach$(B, C_1, k, \{(C_1, C_2), \ldots, (C_k, B)\}$ is called during the run of the algorithm.

  Since $(A, C_1) \in \mathsf{Att}$, A is one of the arguments considered during this call. By assumption $(A, C_1), \ldots, (C_k, B)$ is an attack-path from A to B, therefore no attack appears twice. Hence $(A, C_1) \notin \{(C_1, C_2), \ldots, (C_k, B)\}$. Then $\mathsf{Visited}$ will be updated with (A, C_1) at Line 5 and at Line 6 ReReach$(B, A, k+1, \{(A, C_1), (C_1, C_2), \ldots, (C_k, B)\})$ is called.

This shows that for any n, if there is an attack-path from A to B along the attacks in S, ReReach(B, A, n, S) will be called. □

The next proposition shows that Algorithm 1 is complete.

Proposition 4. *If there is an attack-path from A to B of length n then $n \in \mathsf{Dist}(A, B)$.*

Proof. Suppose that there is an attack-path from A to B of length n. We proceed again by induction on n.

- **If** $n = 0$**:** then by Remark 1 $A = B$. By Line 3 of Algorithm 1, $0 \in \mathsf{Dist}(A, A)$ and $0 \in \mathsf{Dist}(B, B)$.

- **If** $n = 1$**:** then by Remark 1 $(A, B) \in \mathsf{Att}$. By Line 4 of Procedure ReReach, $1 \in \mathsf{Dist}(A, B)$.

Suppose that the statement holds for n up to $k \geq 1$.

- **If** $n = k+1$**:** then there are $C_1, \ldots, C_{k+1} \in \mathsf{Args}$, such that $A = C_1$, $B = C_{k+1}$, $(C_1, C_2), \ldots, (C_k, C_{k+1}) \in \mathsf{Att}$ and there are no $1 \leq i, j \leq k$ such that $i \neq j$ and $(C_i, C_{i+1}) = (C_j, C_{j+1})$ (i.e., the attack-path does not follow an attack twice). Note that for any $2 \leq i, j \leq k+1$ such that $i \leq j$, the corresponding subset of attacks is an attack-path from C_i to C_j. In particular, $(C_2, C_3), \ldots, (C_k, B)$ is an attack-path from C_2 to B of length k.

  By induction hypothesis, $k \in \mathsf{Dist}(C_2, B)$. Since $k \geq 1$, $\mathsf{Dist}(C_2, B)$ was updated at Line 4 of Procedure ReReach during the ReReach$(B, C_3, k-1, \mathsf{S}')$ call of the procedure. By Lemma 2 it follows that $\mathsf{S}' = \{(C_3, C_4), \ldots, (C_k, B)\}$. Then, at Line 5 $\mathsf{Visited}$ is updated with (C_2, C_3) and, at Line 6, ReReach$(B, C_2, k, \mathsf{S}' \cup \{(C_2, C_3)\})$ is called. Since $(C_1, C_2) \in \mathsf{Att}$ and since there is an attack-path from C_1 to C_{k+1} along the attacks of $\mathsf{S}' \cup \{(C_1, C_2), (C_2, C_3)\}$, $\mathsf{Dist}(A, B)$ will be updated at Line 4 with $k+1$. □

In our paper we are interested in the distance between two arguments (since this determines whether the relation is an attack (the distance is odd) or a defense (the distance is even)), but also in the arguments from which an argument is reachable. Both are computed by Algorithm 1 and the next lemma shows the relation between the two.

Lemma 3. $\mathsf{Dist}(A, B) \neq \emptyset$ *iff* $A \in \mathsf{Reach}(B)$.

Proof. Let $\mathcal{AF} = \langle \mathsf{Args}, \mathsf{Att} \rangle$ be an AF and let $A, B \in \mathsf{Args}$. Assume that Algorithm 1 was run on $\mathcal{AF}$. Consider both directions separately.

$\Rightarrow$ Suppose that $\mathsf{Dist}(A, B) \neq \emptyset$. This direction is shown by induction on the minimal value n in $\mathsf{Dist}(A, B)$.

- **If** $n = 0$**:** then $\mathsf{Dist}(A, B)$ was updated at Line 3 of the algorithm (since at Line 4 of the procedure the addition is always more than 0) and thus $A = B$. By Line 3 again it follows that $A \in \mathsf{Reach}(A)$.
- **If** $n = 1$**:** then $\mathsf{Dist}(A, B)$ was updated at Line 4 of the procedure during the first iteration of the **for**-loop, in which case $(A, B) \in \mathsf{Att}$. By Line 3 it follows that $A \in \mathsf{Reach}(B)$.

Now suppose that the statement holds for n up to a value of $k \geq 1$.

- **If** $n = k + 1$**:** then $\mathsf{Dist}(A, B)$ was updated at Line 4 of Procedure ReReach. Therefore, during this run of ReReach, at Line 3, $\mathsf{Reach}(B)$ is updated with $\mathsf{Reach}(A)$. Note that by Line 3 of Algorithm 1 $A \in \mathsf{Reach}(A)$ and hence $A \in \mathsf{Reach}(B)$.

$\Leftarrow$ Now assume that $A \in \mathsf{Reach}(B)$. We consider three cases:

- $A = B$, then $\mathsf{Reach}(B)$ was updated at Line 3 of Algorithm 1, such that $B \in \mathsf{Reach}(B)$ and $0 \in \mathsf{Dist}(A, B)$.
- $\mathsf{Reach}(B)$ was updated at Line 3 of Procedure ReReach, with $\mathsf{Reach}(A)$. Then at Line 4, $\mathsf{Dist}(A, B)$ is updated with $n + 1$.
- $\mathsf{Reach}(B)$ was updated at Line 3 of Procedure ReReach, with $\mathsf{Reach}(C)$ and $A \in \mathsf{Reach}(C)$. Hence, by Proposition 3, there is an attack-path from A to C and there is an attack-path from C to B. If no attack in the path from C to B is used in the attack-path from A to C, the procedure will call all arguments in the attack-path from A to C until it reaches A. At which point $\mathsf{Dist}(A, B)$ will be updated.

  Suppose now that there is some $(D_1, D_2) \in \mathsf{Att}$, such that (D_1, D_2) appears in both paths. Then there is an attack-path from A to D_1 (along the attacks $(A, E_k), \ldots, (E_1, D_1)$, where $E_k, \ldots, E_1 \in \mathsf{Args}$) and there is an attack-path from D_1 to B. Without loss of generality, suppose that (D_1, D_2) is such that there is no attack (D_1', D_2') in the attack-path from A to D_1 that also appears in the attack-path from C to B (otherwise the described procedure has to be repeated). Since there is an attack-path from D_1 to B (say of length l_d), by Lemma 2, ReReach$(B, D_1, l_d, \mathsf{S})$ is called. By assumption $\{(A, E_k), \ldots, (E_1, D_1)\} \cap \mathsf{S} = \emptyset$. Hence, for each $i \in \{1, \ldots, k\}$, during the call for ReReach$(B, D_1, l_d, \mathsf{S})$, ReReach$(B, E_i, l_d + i, \mathsf{S} \cup \{(E_i, E_{i-1}), \ldots, (E_1, D_1)\})$ is called. At ReReach$(B, E_k, l_d + k, \mathsf{S} \cup \{(E_k, E_{k-1}), \ldots, (E_1, D_1)\})$, note that $(A, E_k) \notin \mathsf{S} \cup \{(E_k, E_{k-1}), \ldots, (E_1, D_1)\}$. Hence $\mathsf{Reach}(B)$ is updated with $\mathsf{Reach}(A)$ and $\mathsf{Dist}(A, B)$ is updated with $l_d + k + 1$. Therefore $\mathsf{Dist}(A, B) \neq \emptyset$.

This shows that, in any situation, if $A \in \mathsf{Reach}(B)$ then $\mathsf{Dist}(A, B) \neq \emptyset$. $\square$

With the above results we have the proof of Theorem 1:

Proof. Let $\mathcal{AF} = \langle \mathsf{Args}, \mathsf{Att} \rangle$ be an argumentation framework, $A, B \in \mathsf{Args}$ and suppose that Algorithm 1 was run on $\mathcal{AF}$. Then:

1. Soundness and completeness of the algorithm follows immediately by Propositions 3 and 4.
2. By Lemma 3 we know that $A \in \mathsf{Reach}(B)$ iff $\mathsf{Dist}(A, B) \neq \emptyset$ and by the soundness and completeness of the algorithm (i.e., the first item) it is known that $n \in \mathsf{Dist}(A, B)$ iff there is an atack-path from A to B. $\square$

We now turn to the computational complexity of the algorithm. Note that the algorithm does not determine whether an argument is accepted or not. It is therefore important that the extensions have been determined before running the algorithm.

Theorem 2 (Computational complexity)**.** *Algorithm 1 runs in polynomial time. In particular the time complexity is* $\mathcal{O}(|\mathit{Args}| \cdot |\mathit{Att}|^2)$.

Proof. Let $\mathcal{AF} = \langle \mathsf{Args}, \mathsf{Att} \rangle$ be an argumentation framework and suppose that Algorithm 1 was run on this framework. Then:

- The first **for**-call of the algorithm takes $|\mathsf{Args}|$ time.
- Procedure ReReach runs in $|\mathsf{Att}|^2$: from each attack at most all other attacks are visited exactly once ($|\mathsf{Att}|$) and at most $|\mathsf{Att}|$ attacks end in a single argument ($|\mathsf{Att}|$).
- The procedure is called $|\mathsf{Args}|$ times from Algorithm 1.

This gives a total of $|\mathsf{Args}| + |\mathsf{Args}| \cdot |\mathsf{Att}|^2$, assuming that $\mathsf{Att} \neq \emptyset$ (this is safe to assume since argumentation could be considered interesting only when there are attacks), $\mathcal{O}(|\mathsf{Args}| \cdot |\mathsf{Att}|^2)$. □

A.4 From algorithm to explanation

Algorithm 1 determines for each argument the set of arguments from which it is reachable, as well as the distance between the arguments. From this we can define several notions that will be used in the explanations. We will denote by $\mathsf{Reach}_{\mathrm{odd}}$ [resp. $\mathsf{Reach}_{\mathrm{even}}$] the arguments with odd [resp. even] distance to the considered argument (i.e., $\mathsf{Reach}_{\mathrm{odd}}(A) = \{B \in \mathsf{Reach}(A) \mid \exists n \in \mathsf{Dist}(B, A) \text{ s.t. } n \text{ is odd}\}$ [resp. $\mathsf{Reach}_{\mathrm{even}}(A) = \{B \in \mathsf{Reach}(A) \mid \exists n \in \mathsf{Dist}(B, A) \text{ s.t. } n \text{ is even}\}$].

The next definition shows how DefBy and NotDef, used as a first suggestion for $\mathbb{D}$ in the definition of the explanations, can be defined in terms of the notions calculated by the algorithm.

Definition 20. Let $\mathcal{AF} = \langle \mathsf{Args}, \mathsf{Att} \rangle$ be an AF, $A \in \mathsf{Args}$ and $\mathcal{E} \in \mathsf{Ext}_{\mathsf{sem}}(\mathcal{AF})$ an extension for some semantics sem. Suppose that Algorithm 1 was run on $\mathcal{AF}$. Then:

- $\mathsf{DefBy}(A) = \{B \in \mathsf{Reach}_{\mathrm{even}}(A)\}$ denotes the set of arguments in Args that (in)directly defend A;
- $\mathsf{DefBy}(A, \mathcal{E}) = \mathsf{DefBy}(A) \cap \mathcal{E}$ denotes the set of arguments that (in)directly defend A in $\mathcal{E}$;
- $\mathsf{NotDef}(A, \mathcal{E}) = \{B \in \mathsf{Reach}_{\mathrm{odd}}(A) \mid \mathcal{E} \cap \mathsf{Reach}_{\mathrm{odd}}(B) = \emptyset\}$, denotes the set of all (in)direct attackers of A for which no defense exists from $\mathcal{E}$.

Example 20. In the running example from the paper, for the argumentation framework $\mathcal{AF}(\mathrm{AT}_2)$ we have that $\mathsf{Reach}(B_3) = \{A_2, A_3, B_1, B_3\}$ and $\mathsf{Dist}(A_2, B_3) = \{1, 3, 5\}$; and, where $\mathcal{E} = \{A_1, B_1, B_2, B_3\}$, we still have that $\mathsf{DefBy}(B_3, \mathcal{E}) = \{B_1, B_3\}$, while $\mathsf{NotDef}(A_2, \mathcal{E}) = \{B_1\}$.