Notes on Formal Argumentation Theory

Xin Chen (chenxin_hello@outlook.com)

Latest edit: December 29, 2024

This note gives a concise introduction to Dung's abstract argumentation theory [9] and some extensions (probabilistic extension, in particular). We mainly investigate Dung's original notions of complete, grounded, preferred, and stable semantics. However, for other semantics which are available in the literature since Ding's seminal work [9], for instance, semi-stable, ideal, eager, stage, CF2 and stage2 semantics [2, 5], will be considered in the next version of this note.

Contents

1 Abstract Argumentation

This section is devoted to the *abstract argumentation theory*（抽象论辩理论）introduced in the seminal paper [9] by Dung. This formalism is based on the idea that arguments are defeasible entities which may attack each other and whose acceptance is subject to a given reasonable criterion (called *semantics*). Formally, an argumentation framework is represented as a directed graph in which the arguments are represented as nodes and the attack relations as edges.

Given such a graph, a key question is which set(s) of arguments can be accepted. That is, once the argumentation framework has been constructed, how to determine which arguments to accept or reject? To answer this kind of questions corresponding to define an *argumentation semantics*（论辩语义）. By and large, two kinds of approaches for argumentation semantics are available in the literature:

(1) the *extension*–based approach（基于外延的方法）, proposed in Dung's original paper [9], and

(2) the *labelling*–based approach（基于标签的方法）.

In this note, however, for the sake of simplicity, we only focus on the first one.

1.1 Abstract Argumentation frameworks

An (abstract) argumentation framework is nothing, mathematically speaking, but a pair consists of a set of arguments and a binary relation representing the attack relationship between arguments, that is, a *directed graph* in which the arguments as nodes and the attack relations as arrows. An argument is an abstract entity whose role is solely determined by its relations to other arguments.

Definition 1.1 (Argumentation frameworks, AFs) An *argumentation framework* (论辩框架) is a pair

$$AF = (Arg, \rightharpoonup)$$

where Arg is a non–empty set of arguments, and $\rightharpoonup \subseteq Arg \times Arg$ is a binary relation on Ar. $\dashv$

Given an AF $AF = (Arg, \rightharpoonup)$, let $a, b \in Arg$, we say that a *attacks/defeats* b (accordingly, a is an *attacker* of b) iff $a \rightharpoonup b$ holds. For any $X \cup \{a\} \subseteq Arg$, then we say that X *attacks* a, denoted as $X \rightharpoonup a$, if there exists $b \in X$ s.t. $b \rightharpoonup a$. Likewise, we say that a *attacks* X, written as $a \rightharpoonup X$, if there is $b \in X$ s.t. $a \rightharpoonup b$. For $a \in Arg$, we define the sets a^- and a^+ as follows: (各种攻击关系)

$$\begin{aligned} a^- &:= \{b \in Arg \mid b \rightharpoonup a\}, \\ a^+ &:= \{b \in Arg \mid a \rightharpoonup b\}. \end{aligned}$$

Those two operations can be extended att any subset $X \subseteq Arg$ by

$$X^\pm = \bigcup_{a \in X} a^\pm \qquad \pm \in \{+, -\}.$$

The attack relations between argument(s) aforementioned are summarized in Table 1.

Table 1: The attack relations between argument(s)

notation	meaning
$a \rightharpoonup b$	a attacks b
$a \rightharpoonup X$	a attacks X $(\exists b \in X : a \rightharpoonup b)$
$X \rightharpoonup a$	X attacks a $(\exists b \in X : b \rightharpoonup a)$
$X \rightharpoonup Y$	X attacks Y $(\exists a \in X, \exists b \in Y : a \rightharpoonup b)$

Definition 1.2 (Defense) Let $AF = (Arg, \rightharpoonup)$ be an AF and $X \subseteq Arg$. The set X *defends* $a \in Arg$ (or, "a *acceptable* w.r.t. X"[1]) iff $\forall b \in Arg : b \rightharpoonup a \Rightarrow X \rightharpoonup b$ (or equivalently, $\forall b \in a^- : X \rightharpoonup b$, where a^- is the set of attackers of a). (辩护) $\dashv$

According to vacuous truth, if $a^- = \emptyset$ (such a are called *initial arguments*, the set of initial arguments of an AF denoted as $\mathcal{NI}(AF)$, i.e., $\mathcal{IN}(AF) = \{a \in Ar \mid a^- = \emptyset\}$) then a defended by any set of arguments, including $\emptyset$ of course.

Definition 1.3 (Characteristic function) Let $AF = (Arg, \rightharpoonup)$ be an AF. The *characteristic function* of AF is mapping $C_{AF} : \wp(Arg) \to \wp(Arg)$ such that

$$C_{AF}(X) = \{a \in Arg \mid X \text{ defends } a\}$$

for each $X \subseteq Arg$. (刻画函数) $\dashv$

[1] This is the original terminology in Dung's paper [9].

Proposition 1.4 For any AF $AF = (Arg, \rightharpoonup)$, then:

(1) $C_{AF}(\emptyset) = \mathcal{IN}(AF) = \{a \in Arg \mid a^- = \emptyset\}$.

(2) $\mathcal{IN}(AF) \subseteq C_{AF}(X)$.

(3) $C_{AF}(X) = (X^+)^+$.

where $X \subseteq Arg$. $\dashv$

Definition 1.5 (Conflict–freeness) Let $AF = (Arg, \rightharpoonup)$ be an AF and $X \subseteq Arg$. X is said to be *conflict–free* (无冲突) iff $\neg\exists a, b \in X : a \rightharpoonup b$. $\dashv$

Clearly, the emptyset $\emptyset$ is conflict–free. Furthermore, if each argument has at least one attacker, i.e. $a^- \neq \emptyset$ for every argument a, then $\emptyset$ is a (conflict–free) fixed point of the characteristic function. (什么情况下空集是不动点)

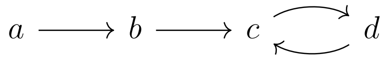
After introduced the notion of *defense*, a basic requirement for a set of arguments is the capability to defend all its elements.

Definition 1.6 (Admissibility) Let $AF = (Arg, \rightharpoonup)$ be an AF. A set $X \subseteq Arg$ is called an *admissible set* (可容许集) iff (1) X is conflict–free and (2) $X \subseteq C_{AF}(X)$. $\dashv$

Thus, an admissible set is required to be both internally coherent, that is, conflict-free and able to defend its elements. Admissible sets always exist. A very trivial case is that the empty set $\emptyset$ is admissible for any argumentation framework. (空集是可容许集)

Example 1.7

$AF :$



- non–empty conflict–free sets: $\{a\}, \{b\}, \{c\}, \{d\}, \{a, c\}, \{a, d\}, \{b, d\}$.

- non–empty admissible sets: $\{a\}, \{d\}, \{a, c\}, \{a, d\}$.

- $C_{AF}(\{a\}) = \{a\}$, $C_{AF}(\{d\}) = \{d, a\}$, $C_{AF}(\{a, c\}) = \{a, c\}$, $C_{AF}(\{a, d\}) = \{a, c, d\}$. $\dashv$

Finally, it is worth recalling that admissibility and defense are related by a basic property. In terms of extensions, if an admissible set defends an argument, it is possible to add the argument to the set while preserving its admissibility and its capability to defend any other argument. This was proved in the so–called Dung's Fundamental Lemma.

Lemma 1.8 (Dung's Fundamental Lemma [9]) For any AF $AF = (Arg, \rightharpoonup)$, let X be an admissible set and a, b be arguments defended by X. Then

(1) $X' = X \cup \{a\}$ is an admissible set;

(2) X' defends b.

⊣

1.2 An overview of (extension-based) argumentation semantics

In this subsection we will provide an overview of some well–known argumentation semantics, starting from the very basic notion of “naïve semantics” [2, § 3.2] and then discussing Dung’s original concepts of *complete*, *stable*, *preferred* and *grounded* semantics [9].

Remark 1.9 In the literature, *admissibility* and *conflict–freeness* mentioned in § 1.1 are sometimes viewed as semantics, and sometimes as properties. We choose to treat them as properties in this note. ⊣

Definition 1.10 (Extension–based semantics) An *extension–base semantics* (*semantics*, for short) σ associates with any argumentation framework $AF = (Arg, \rightharpoonup)$ is a subset of $\wp(Arg)$, denoted as $\sigma(AF)$. The elements in $\sigma(AF)$ are called *extensions* (under semantics σ) (外延).

Let $\mathcal{D}^\sigma$ be the class of AFs where a semantic σ is defined, that is,

$$\mathcal{D}^\sigma = \{AF \mid \sigma(AF) \neq \emptyset\}.$$

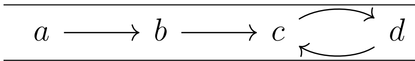
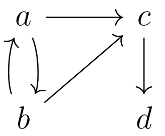
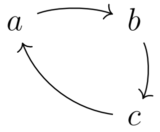
A semantics σ is called *universally defined* if $\mathcal{D}^\sigma$ includes all AFs. If for any $AF \in \mathcal{D}^\sigma$ we have that $|\sigma(AF)| = 1$, i.e. the semantics σ always prescribes exactly one extension, then σ is said to belong to the *unique–status approach*, it is said to belong to the *multiple–status approach*, otherwise. (基于外延的语义) ⊣

1.2.1 Naïve Semantics

Naïve semantics (denoted as $\mathcal{NA}$) corresponds to selecting as many arguments as possible, provided that there are no conflicts among them. It is a sort of greedy strategy, driven by the only criterion of avoiding conflicts. Formally it corresponds to requiring conflict–freeness together with a maximality property.

Definition 1.11 (Naïve extensions) Let $(Arg, \rightharpoonup)$ be an AF. A set $X \subseteq Arg$ is called a *naïve extension* iff X is a maximal conflict–free set. ⊣

Example 1.12

AF	naïve extensions $\mathcal{NA}(AF)$
	$\{a, c\}, \{a, d\}, \{b, d\}$
	$\{a, d\}, \{b, d\}, \{c\}$
	$\{a\}, \{b\}, \{c\}$

⊣

1.2.2 Complete Semantics (完全语义)

Complete semantics (denoted as $\mathcal{CO}$) can be regarded as a strengthening of the basic requirements enforced by the idea of admissibility, it lies at the heart of all Dung's semantics (see Fig. 1).

A complete extension is a conflict–free set which includes precisely those arguments it defends. That is, if an argument is defended by the set it should be in the set, and if an argument is not defended by the set, it should not be in the set.

Definition 1.13 (Complete extensions) Let $AF = (Arg, \rightharpoonup)$ be an AF. A set $X \subseteq Arg$ is called a *complete extension* iff X is conflict–free and $X = C_{AF}(X)$. ⊣

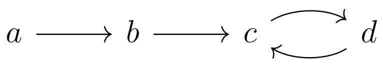
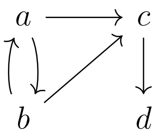
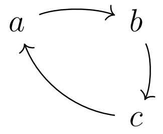
Technically this means that a complete extension is a *conflict–free fixed point* of the characteristic function. It is clear that every complete extension is an admissible set, but the reverse does not hold in general. (无冲突不动点)
"Intuitively, the notion of complete extension captures the kind of confident relational agent who believe in every thing he can defend"[9].

Proposition 1.14 Let $AF = (Arg, \rightharpoonup)$ be any AF.

- $\mathcal{CO}(AF) \neq \emptyset$, that is, complete semantics is universally defined (see Definition 1.10).
- $\emptyset \in \mathcal{CO}(AF)$ iff $\mathcal{NI}(AF) = \emptyset$, where $\mathcal{NI}(AF) \coloneqq \{a \mid a^- = \emptyset\}$.
- $\forall E \in \mathcal{CO}(AF) : \mathcal{NI}(AF) \subseteq E$.

⊣

Example 1.15

AF	complete extensions $\mathcal{E}_{\mathcal{CO}}(AF)$
	$\{a\}, \{a, d\}, \{a, c\}$
	$\emptyset, \{a, d\}, \{b, d\}$
	$\emptyset$

$\dashv$

1.2.3 Grounded Semantics （基语义）

To accept only the arguments that one cannot avoid accepting, to reject only the arguments that one cannot avoid rejecting, and abstaining as much as possible. This gives rise to the most skeptical semantics among those based on complete extensions, namely the *grounded semantics* (denoted as $\mathcal{GR}$).

Definition 1.16 (The grounded extension) Let $AF = (Arg, \rightharpoonup)$ be an AF. The *grounded extension* of AF is a minimal conflict–free fixed point of the characteristic function. $\dashv$

Notice that the uniqueness of the grounded extension. Since the characteristic function C_{AF} is monotonic, it follows from the Knaster-Tarski Theorem that C_{AF} has a unique smallest fixed point, it can then be proved that this fixed point is also conflict–free.

Proposition 1.17 For any AF $AF = (Arg, \rightharpoonup)$, the following statements are equivalent:

(1) X is a minimal conflict–free fixed point of C_{AF}.

(2) X is the smallest fixed point of C_{AF}.

$\dashv$

It follows that:

- the grounded extension is unique (i.e. grounded semantics belongs to the unique-status approach);
- the grounded extension is the least complete extension, in particular it is included in any complete extension.

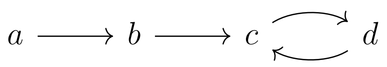
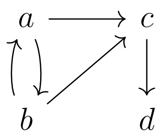
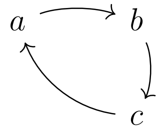
The grounded extension of an argumentation framework AF will be denoted as $GR(AF)$. By definition, the grounded extension coincides with the intersection of all complete extensions, that is

$$GR(AF) = \bigcap \mathcal{CO}(AF).$$

Hence, a unique grounded extension always exists, although it may be the empty set.

Proposition 1.18 If there are no initial arguments in an AF, then the grounded extension is the empty set. That is, $\mathcal{IN}(AF) = \emptyset \Rightarrow GR(AF) = \emptyset$. $\dashv$

Example 1.19 (Grounded extension)

AF	grounded extensions $\mathcal{GR}(AF)$
	$\{a\}$
	$\emptyset$
	$\emptyset$

$\dashv$

1.2.4 Preferred Semantics (优先语义)

The idea of maximizing accepted arguments is expressed by *preferred semantics* ($\mathcal{PR}$, in symbol).

Definition 1.20 (Preferred extensions) Let $AF = (Arg, \rightharpoonup)$ be an AF. A *preferred extension* is a maximal (w.r.t. $\subseteq$) admissible set of AF. $\dashv$

Every argumentation framework possesses at least one preferred extension. Hence, preferred extension semantics is always defined for any argumentation framework. Relationships of preferred extensions with other semantics notions have been analyzed in Dung's [9]. Preferred extensions can for instance equivalently be characterized as maximal complete extensions.

Proposition 1.21 Let $AF = (Arg, \rightharpoonup)$ be an AF and $X \subseteq Arg$. The following two statements are equivalent:

(1) X is a maximal admissible set of AF.

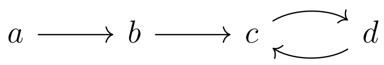
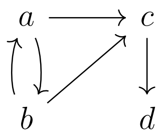
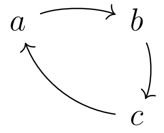
(2) X is a maximal complete extension of AF.

$\dashv$

This in particular implies that the grounded extension is included in any preferred extension, as it is in any complete extension.

Preferred semantics has often been regarded as the most satisfactory semantics in the context of Dung's framework.

Example 1.22 (Preferred extensions)

AF	preferred extensions $\mathcal{PR}(AF)$
	$\{a, d\}, \{a, c\}$
	$\{a, d\}, \{b, d\}$
	$\emptyset$

$\dashv$

1.2.5 Stable Semantics (稳定语义)

Stable semantics (denoted as $\mathcal{ST}$) relies on a very simple intuition: an extension should be able to attack all arguments not included in it.

Definition 1.23 (Stable extensions) Let $AF = (Arg, \rightharpoonup)$ be an AF. A *stable extension* (稳定外延) of AF is a conflict–free set X such that $X \cup X^+ = Arg$. $\dashv$

A stable extension is an admissible set. In the context of game theory, the notion of stable extension coincides with the notion of *stable solution* of n-person games [9]. Every stable extension is a preferred extension, but not vice versa.

Proposition 1.24 Let $AF = (Arg, \rightharpoonup)$ be an AF and $X \subseteq Arg$. The following statements are equivalent:

(1) X is a stable extension.

(2) X is an admissible set s.t. $X \cup X^+ = Arg$.

(3) X is a complete extension s.t. $X \cup X^+ = Arg$.

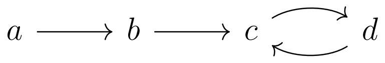
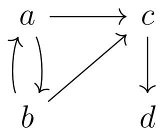
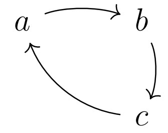
(4) X is a preferred extension s.t. $X \cup X^+ = Arg$.

(5) $X^+ = Arg \setminus X$.

⊣

No stable extension is empty, and not every argument framework has stable extensions.

Example 1.25 (Stable extensions)

AF	stable extensions $\mathcal{ST}(AF)$
	$\{a,d\},\{a,c\}$
	$\{a,d\},\{b,d\}$
	–

⊣

1.2.6 Semi-Stable Semantics (coming in soon)

1.2.7 Ideal and Eager Semantics (coming in soon)

1.2.8 Stage Semantics (coming in soon)

1.2.9 $CF2$ and stage2 Semantics (coming in soon)

cf2: $\mathcal{CF}2$

ideal: $\mathcal{ID}$

semi–stable: $\mathcal{SST}$

stage: $\mathcal{STG}$

Definition 1.26 Let $AF = (Arg, \rightharpoonup)$ be an AF. For a conflict–free set X, it holds that:

- $X \in \mathcal{ID}(AF)$ iff X is $\subseteq$–maximal among $\{X' \mid X' \in ad(AF)\ \&\ X' \subseteq E$ for each $E \in \mathcal{PR}(AF)\}$.

- $X \in \mathcal{SST}(AF)$ iff $X \in ad(AF)$ and there is no $T \in ad(AF)$ with $X^{\oplus} \subset T^{\oplus}$, where $X^{\oplus} = X^{+} \cup X^{-}$, similar for $T^{\oplus}$.

- $X \in \mathcal{STG}(AF)$ iff there is no $T \in cf(AF)$ with $X^{\oplus} \subset T^{\oplus}$.

Following, we give the recursive definition of $CF2$ semantics. $X \in \mathcal{CF}2(AF)$ if

- $X \in \mathcal{NA}(AF)$ if $|SCC_{S_F}| = 1$, and

- $\forall S \in SCC_{S_F}$, $X \cap S \in \mathcal{CF}2(AF\downharpoonleft)$; otherwise.

$\dashv$

1.3 Computational problems and Complexity

When studying computational complexity we are only interested in AFs where the arguments set Arg is finite.

Given an AF $AF = (Arg, \rightharpoonup)$ and a semantics/property σ, there are several kinds of computational problems usually be considered:

(1) the *verification problem* （验证问题）, denoted as Var_σ: deciding whether a set $X \subseteq Arg$ is a σ–extension of AF, that is, whether $X \in \sigma(AF)$?

(2) the *credulous acceptance problem* （轻信接受问题）, denoted as $Cred_\sigma$: for $a \in Arg$, deciding whether a is credulously accepted, that is deciding whether a belongs to a σ–extension of AF. I.e., whether $a \in \bigcup \sigma(AF)$?

Every argument that does not attack itself will be credulously accepted under the conflict–free property.

(3) the *skeptical acceptance problem* （谨慎接受问题）, denoted as $Skept_\sigma$: deciding whether a is skeptically accepted, that is deciding whether a belongs to every σ–extension of AF. I.e., whether $a \in \bigcap \sigma(AF)$?

Another task is deciding whether an AF provides any coherent conclusion.

(4) *Existence of an extension* （外延存在问题）, $Exists_\sigma$: Is $\sigma(AF) \neq \emptyset$?

(5) Existence of a non–empty extension （非空外延存在问题）, $Exists_\sigma^{\neg\emptyset}$: Does there exist a set $S \neq \emptyset$ such that $S \in \sigma(AF)$?

Finally, we will also consider the problem of deciding whether a semantics yields a unique extension for a given an AF:

(6) *Uniqueness of the solution*, $Unique_\sigma$, Is there a unique set $X \in \sigma(AF)$, i.e., is $\sigma(AF) = \{X\}$?

Clearly, CA_τ and SA_τ are identical problems with $\tau \in \{\mathcal{GR}, \mathcal{ID}\}$. For $\sigma \in \{\mathcal{CO}, \mathcal{GR}, \mathcal{PR}, \mathcal{ST}\}$, the computational complexity of those problems are summarized in table 2 (for more detail, see [10, 11]).

................................some symbols ....

$AF \mid\!\sim^c_\sigma a$ credulously accepted

$AF \mid\!\sim^s_\sigma a$ skeptically accepted

$AF \mid\!\sim^\circ_\sigma a$ where $\circ \in \{s, c\}$

Table 2: Complexity of some problems

semantics or properties	Ver_σ	$Cred_\sigma$	$Skept_\sigma$
cf	in L	in L	trivial
ad	in L	NP–c	trivial
$\mathcal{CO}$	in P	NP–c	P–c
$\mathcal{GR}$	in P	in P	in P
$\mathcal{PR}$	coNP–c	NP–c	$\prod_2^p$–c
$\mathcal{ST}$	in P	NP–c	coNP–c
$\mathcal{SST}$ (semi–stable)	coNP–c	$\sum_2^p$–c	$\prod_2^p$–c

Cf [11, Table 1]

1.4 A summary

The grounded extension is uniquely determined and always exists [9], and not every framework is guaranteed to produce a stable extension.

It is well–known that the set of **complete extensions** forms a *complete semilattice* w.r.t. $\subseteq$, where $GR(AF)$ is the meet element, whereas the greatest elements are the preferred extensions.

- for any of the considered semantics σ except stable semantics we have that $\sigma(AF) \neq \emptyset$ holds, i.e., these semantics always propose at least one extension.

- Grounded and ideal semantics always yield exactly one extension.

- stable, semi-stable, and stage semantics coincide for AFs with at least one stable extension.

$\mathcal{ST} \subseteq \mathcal{STG} \subseteq \mathcal{NA} \subseteq cf$

$\mathcal{ST} \subseteq \mathcal{SST} \subseteq \mathcal{PR} \subseteq \mathcal{CO} \subseteq ad \subseteq cf$

$\mathcal{ST} \subseteq \mathcal{CF}2 \subseteq \mathcal{NA}$

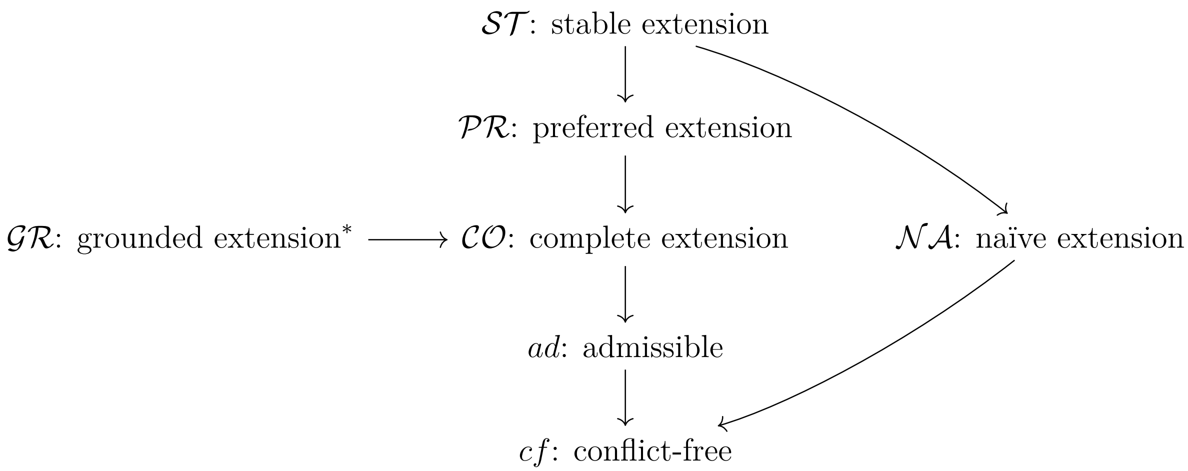

$\mathcal{ST}$: stable extension
$\mathcal{PR}$: preferred extension
$\mathcal{GR}$: grounded extension*
$\mathcal{CO}$: complete extension
$\mathcal{NA}$: naïve extension
ad: admissible
cf: conflict-free

Figure 1: Relations among extension–based semantics

Note: The intuitive meaning of arrow “$x \longrightarrow y$” is that “every x is a y”. For example, every stable extension is a preferred extension. * denotes this semantics is unique status semantics.

Table 3: A summary of the examples mentioned

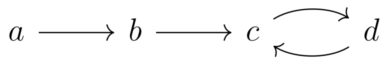
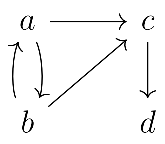
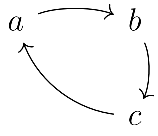
AF	naïve	complete	grounded	preferred	stable
	$\{a, c\}$ $\{a, d\}$ $\{b, d\}$	$\{a\}$ $\{a, c\}$ $\{a, d\}$	$\{a\}$	$\{a, c\}$ $\{a, d\}$	$\{a, c\}$ $\{a, d\}$
	$\{a, d\}$ $\{b, d\}$ $\{c\}$	$\emptyset$ $\{a, d\}$ $\{b, d\}$	$\emptyset$	$\{a, d\}$ $\{b, d\}$	$\{a, d\}$ $\{b, d\}$
	$\{a\}$ $\{b\}$ $\{c\}$	$\emptyset$	$\emptyset$	$\emptyset$	–

Table 4: Some basic concepts

$AF = (Arg, \rightharpoonup)$ is an AF and $X \cup \{a, b\} \subseteq Arg$.

Notions	Definition or equivalent descriptions
$a \rightharpoonup b$	a attacks/defeats b a is an attacker/counterargument of b
$X \rightharpoonup a$	X attacks a: $\exists b \in X : b \rightharpoonup a$
a^-	$a^- = \{b \in Arg \mid b \rightharpoonup a\}$
a^+	$a^+ = \{b \in Arg \mid a \rightharpoonup b\}$
X^-	$X^- = \{b \in Arg \mid \exists a \in X : b \rightharpoonup a\} = \bigcup_{a \in X} a^-$
X^+	$X^+ = \{b \in Arg \mid \exists a \in X : a \rightharpoonup b\} = \bigcup_{a \in X} a^+$
$\mathcal{IN}(AF)$: the set of initial arguments	$\mathcal{IN}(AF) = \{a \in Arg \mid a^- = \emptyset\}$
X defends a / a is admissible w.r.t. X	$\forall b : b \rightharpoonup a \Rightarrow X \rightharpoonup b$ [X defends all initial arguments]
C_{AF}: the characteristic function	$C_{AF}(X) = \{a \mid X \text{ defends } a\}$ $C_{AF}X = (X^+)^+$ [$\emptyset$ is a fixed point if $\mathcal{IN}(AF) = \emptyset$]
$cf(AF) \ni X$ is conflict–free	$\nexists a, b \in X : a \rightharpoonup b$ [$\emptyset$ is conflict–free] $\nexists a, b \in X : a \in b^-$ $X \cap X^+ = \emptyset$
$ad(AF) \ni X$ is admissible	X is conflict–free and $X \subseteq C_{\mathcal{AF}}(X)$ [$\emptyset$ is admissible]
Semantics	
$\mathcal{NA}(AF) \ni X$ is a *naïve extension*	X is a maximal conflict–free set
$\mathcal{CO}(AF) \ni X$ is a *complete extension*	X is a conflict–free and $C_{AF}(X) = X$ conflict–free fixed point
$\mathcal{GR}(AF) \ni X$ is the *grounded extension*	X is the least fixed point of C_{AF} X is the minimal complete extension $X = GR(AF) = \bigcap \mathcal{CO}(AF)$
$\mathcal{PR}(AF) \ni X$ is a *preferred extension*	X is a maximal admissible set X is a maximal complete extension
$\mathcal{ST}(AF) \ni X$ is a *stable extension*	X is conflict–free and $X \cup X^+ = Arg$ $X^+ = Arg \setminus X$ X is a preferred extension and $X \cup X^+ = Ar$ X is a complete extension and $X \cup X^+ = Ar$ X is admissible and $X \cup X^+ = Ar$ $X = \{a \in Ar \mid X \not\rightharpoonup a\}$ X is conflict–free and attacks each argument that is not in X

2 Some extensions

[18]

2.1 Preference–based AF

Several works generalizing Dung's framework to handle preferences over arguments have been proposed.

Definition 2.1 (Preference–based Argumentation Frameworks, PAFs) A *preference–based argumentation framework* (PAF) is a triple $(Arg, \rightharpoonup, >)$ such that $(Arg, \rightharpoonup)$ is an AF and $>$ is a strict partial order (i.e. an irreflexive, asymmetric and transitive relation) on Arg, called *preference relation.* $\dashv$

For any $a, b \in Arg$, $a > b$ means that "a is better than b" for the agent.

The preference in a PAF $(Arg, \rightharpoonup, >)$ working as follows: classical argumentation semantics are used to obtain the extensions of the underlying $(Arg, \rightharpoonup)$, and then the preference relation $>$ is used to obtain a preference relation $\sqsupseteq$ over such extensions, so that the *best extensions* w.r.t. $\sqsupseteq$ are eventually selected.

2.2 Value–based AF

The motivation behind VAFs (value–based argumentation frameworks) is to offer an explanatory mechanism accounting for choices between distinct justifiable collections S and T, which are not collectively acceptable, i.e., S and T may be admissible under Dung's semantics, however, $S \cup T$ fails to be.

2.3 Probabilistic AF

Recently, there are has been an increasing interest in modeling uncertainty in argumentation. This has been carried out by combining probability theory with formal argumentation. The extension of Dung's abstract argumentation framework with probability theory is called *probabilistic argumentation framework.*

In general a probabilistic argumentation framework consists of probabilistic arguments and probabilistic attacks [12, 15]. As shown in [16], however, an argumentation framework with probabilities on both arguments and attacks can be transformed into an equivalent one with certain attack relations (i.e. the probability of each attack is 1).[2] Hence, for the sake of brevity, in the section we only focus on probabilistic argumentation frameworks where only arguments are uncertain.

[2] The equivalent transformation between those two kinds of probabilistic AFs has linear complexity [16].

Definition 2.2 (Probabilistic Argumentation Frameworks, PrAFs) A *probabilistic argumentation framework* (PrAF) is a tuple

$$PAF = (Arg, \rightharpoonup, P)$$

where $(Arg, \rightharpoonup)$ is an argumentation framework and P is a function assigning a non-zero probability value to each argument in Arg, that is, $P\colon Arg \to (0,1]$ (or $P\colon Arg \to (0,1] \cap \mathbb{Q}$ to facilitate the calculation). $\dashv$

Intuitively, the value assigned by P to an argument a represents the probability that a actually occurs, moreover, every attack $a \rightharpoonup b$ occurs with conditional probability 1, that is, a attacks b whenever both a and b occur. Or in a *justification perspective*: $P(a)$ is treated as the probability that a is a justified point and therefore should appear in the graph, and $1 - P(a)$ is the probability that a is not a justified point and so should not appear in the graph.

Thus, an argument $a \in Arg$ is viewed as a probabilistic event which is independent from the other events associated with other arguments $a \neq b \in Arg$.

The meaning of a PrAF is given in terms of *possible worlds*. Given a PrAF $PAF = (Arg, \rightharpoonup, P)$, a *possible world* of PAF is an AF $(Arg', \rightharpoonup')$ which is the restriction of $(Arg, \rightharpoonup)$ to Arg', that is, $Arg' \subseteq Arg$ and $\rightharpoonup' = \rightharpoonup \cap (Arg' \times Arg')$. The set of all possible worlds of PAF is denoted as $pw(PAF)$.

The probability distribution over arguments can generate a probability distribution over possible worlds (subgraphs) of the original argument framework. Using the subgraphs, we can then explore the notions of probability distributions over admissible sets, extensions, and inferences.

The intuition hereby is as follows: for $a \in Arg$, $P(a)$ is the probability that a belongs to an arbitrary possible world (full subgraph) of PAF; while $1 - P(a)$ is the probability that a does not exist in an arbitrary possible world (full subgraph).

(using the justification perspective [13], where $P(a)$ means the probability that a is justified in appearing in the graph)

We can use the probability assigned to each argument to generate a probability distribution over the subgraphs. So each subgraph can be viewed as an "interpretation" of what the argument graph should be. If all the arguments have probability 1, then the argument graph itself has probability 1, and that is the only interpretation we should consider (using the constellations approach). But, if one or more arguments has a probability less than 1, then there will be multiple "interpretations". So for instance, if there is exactly one argument a in the framework G with probability less than 1, then there are two "interpretations" , the first with a, and the second without a. So using the justification perspective, with the constellations approach, we can treat the set of subgraphs of a G as a *sample space*, where one of the subgraphs is the "true" argument graph.

Definition 2.3 The *interpretation* for a PrAF $PAF = (Arg, \rightharpoonup, P)$ is function I over the set $pw(PAF)$ such that each $w \in pw(PAF)$ is assigned by I the probability

$$I(w) = \left(\prod_{a \in Arg(w)} P(a) \right) \cdot \left(\prod_{a \in Arg \setminus Arg(w)} (1 - P(a)) \right)$$

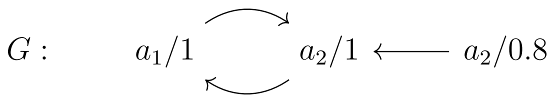
where $Arg(w)$ is the set of arguments in w. $\dashv$

Thus, the probability of a subgraph captures the degree of certainty that the subgraph contains exactly the arguments that are regarded as holding.

Theorem 2.4 ([15], revised) Let $PAF = (Arg, \rightharpoonup, P)$ be a PrAF, the function I is a *probability distribution* on the set $pw(PAF)$, i.e., I is a function such that $\sum_{w \in pw(PAF)} I(w) = 1$. $\dashv$

Proof. By induction on the size of Arg. See the proof of [14, Prop. 3 in p. 60] for details. ■

Example 2.5 Considering following PrAF



Clearly G has eight subgraphs, then we get the following probability distribution over the subgraphs:

	$G_1 = G$	G_2	G_3	G_4	G_5	G_6	G_7	G_8
I	0.8	0.2	0					

$\dashv$

If all the arguments in a PrAF G have probability of 1, then the only subgraph of G to have non-zero probability is itself, and so it has probability 1. At the other extreme, if all the arguments in a probabilistic argument graph G have probability of 0, then the empty graph has probability 1.

Given a PrAF $G = (Arg, \rightharpoonup, P)$, and a set of arguments $\Gamma \subseteq Arg$, we want to calculate the probability that Γ is a σ–extension, which we denote by $P_G^\sigma(\Gamma)$, where $\sigma \in \{ad, co, pr, gr, st\}$. For this, we take the sum of the probability of the full subgraphs for which Γ is a σ–extension.

The analogous *credulous acceptance problem* in the context of a probabilistic argumentation framework, i.e. the probability that a given set of arguments is an extension under a given semantics, is the following.

Definition 2.6 (Probabilistic credulous acceptance) Given a PrAF $PAF = (Arg, \rightharpoonup, P)$, an argument $a \in Arg$, the probability $PrCA^{\sigma}_{PAF}(a)$ that a is credulously acceptable w.r.t. semantics σ is

$$PrCA^{\sigma}_{PAF}(g) = \sum_{w \,\in\, pw(PAF) \;\wedge\; a \,\in\, \bigcup \sigma(w)} I(w).$$

⊣

Thus, the probability that an argument a is credulously accepted according to a semantics σ is defined as the sum of the probabilities of the possible worlds of a PrAF for which argument a is credulously accepted.

Computing $PrCA^{\sigma}_{PAF}(a)$ is $FP^{\#P}$–hard for $\sigma \in \{\mathcal{GR}, \mathcal{PR}, \mathcal{ST}, \mathcal{STT}\}$ [Fazzinga, Flesca, and Furfaro 2018], where $FP^{\#P}$ is the class of functions computable by a polynomial–time Turing machine with a $\#P$ oracle.

Probabilistic credulous acceptance does not express the probability that a given argument is accepted. However, a new problem, called *Probabilistic Acceptance*, was given in [1] can be intuitively stated as follows. Given a probabilistic AF, a semantics σ and a goal argument a, computing the probability that a is accepted.

Definition 2.7 (Probabilistic Acceptance) Given a PrAF $PAF = (Arg, \rightharpoonup, P)$ and an argument $a \in Arg$, the probability $PrA^{\sigma}_{PAF}(a)$ that a is acceptable w.r.t. semantics σ is

$$PrA^{\sigma}_{PAF}(a) = \sum_{w \in pw(PAF) \wedge E \in \sigma(w) \wedge g \in E} I(w) \cdot Pr(E, w, \sigma)$$

where $Pr(\cdot, w, \sigma)$ is a PDF over the set $\sigma(w)$. ⊣

3 Dynamic

An important feature of the argumentation forms is that, in practice, these are not *static* systems: a particular AF $AF = (Arg, \rightharpoonup)$ represents only a "snapshot" of the environment, and, as further facts, information and opinions emerge the form of the initial view may change significantly in order to accommodate these. For example, additional arguments may have to be considered so changing AF; existing attacks may cease to apply and new attacks (arising from changes to AF) come into force.

It is clear that accounting for such dynamic aspects raises a number of issues in terms of assessing the acceptability status of individual arguments.

4 Overview

4.1 Belief Revision and Argumentation

There are several works that combine belief revision and argumentation. Coste-Marquis et al. [7] introduce novel enforcement strategies for modifying argumentation frameworks to make sure that a specific set of arguments is part of its extension, which are close to the process of belief revision. Baroni et al. compare and relate belief revision and argumentation as approaches to model reasoning processes [3]. Coste-Marquis et al. [On the revision of argumentation systems: minimal change of arguments statuses, in: Proceedings of the Fourteenth International Conference on Principles of Knowledge Representation and Reasoning, 2014, pp. 52–61.] derive argumentation systems that satisfy given revision formulas, i.e., given an argumentation system and a revision formula that expresses how the statuses of some arguments have to be changed under a chosen semantics, the derived argumentation systems are such that the corresponding extensions are as close as possible to the extensions of the input system. Diller et al. [8] follow this work and study how to update an argumentation framework with new information, either a formula or another argumentation framework, and defines rationality criteria for such updates. Paglieri and Castelfranchi propose a data-oriented belief revision that enables incorporating computational argumentation [17]. Booth et al. [6] propose two methods to restore consistency when an agent's belief state, composed of a Dung's argumentation framework and a propositional constraint, are inconsistent: firstly, a normal expansion that may alter the set of complete labellings; and secondly, belief revision techniques that aim to retain similarity to the original labellings.

5 Interlude: Structured Argumentation

When we want a more detailed formalisation of arguments we can turn to structured argumentation, in which we assume a *formal language* for representing **knowledge**, and specifying how *arguments* and *counterarguments* can be constructed from that knowledge.

An argument is then said to be *structured* in the sense that normally the *premises* and *claim* of the argument are made explicit, and the relationship between the premises and claim is formally defined. This means we can describe arguments and attacks is structured argumentation as follows:

- *Argument*

  is a tuple containing *support* (premises) for the argument and the *claim* (conclusion) of the argument.

  other information: how the claim is obtained from the support

- *Attack*

  a binary relation over arguments that denotes when one argument is in conflict with another argument.

  In structured argumentation, there is normally a formal definition for determining when this relation holds.

attack vs. defeat

defeat relation which is used as the binary relation in the Dung framework.

Four approaches to structured argumentation:

(1) **ABA**

ABA is a general framework for logic-based instantiation of abstract argumentation. A specific system in ABA is based on a **deductive system**, including a set of inference **rules**. A subset of the language in the deductive system is specified as **assumptions**. Each argument corresponds to a set of assumptions that, with the inference rules, proves a claim.

Attacks between arguments are obtained in ABA via a notion of *'contrary' assumptions*, specifying how to contradict them. Then, an argument *attacks* another if its claim contradicts (is the contrary of) some assumption in the other.

ABA is equipped with a family of computational mechanisms for capturing specific argumentation semantics, corresponding to semantics for abstract argumentation.

(2) **ASPIC+**

This framework is based on two ideas:

(i) conflicts between arguments are often resolved with explicit preferences; and

(ii) arguments are built with two kinds of inference rules: (a) *strict (deductive) rules*, whose premises guarantee their conclusion, and (b) *defeasible rules*, whose premises only create a presumption in favour of their conclusion.

The second idea implies that ASPIC+ does not primarily see argumentation as inconsistency handling in a given 'base' logic: conflicts between arguments may not only arise from the inconsistency of a knowledge base but also from the defeasibility of the reasoning steps in an argument.

Accordingly, in ASPIC+ an argument can be attacked in three ways: on its uncertain premises or on its defeasible inferences, and the latter by either attacking its conclusion or the inference itself.

ASPIC+ is not a system but a framework for specifying systems. A main objective is to identify conditions under which instantiations of ASPIC+ satisfy logical consistency and closure properties.

(3) **Defeasible Logic Programming (DeLP)**

The DeLP system is based on the language of logic programming. It includes *strong negation* and *default negation*. The language was extended with the addition of defeasible rules and presumptions (that represent defeasible assumptions). Its inference engine relies on the construction of arguments to support literals, and the dialectical analysis of the reasons for and against accepting a particular argument.

A DeLP argument, supporting a claim in the form of a literal, is considered a *warrant* for that claim if all the arguments that can be posed against it are *defeated*.

Defeat comes in the form of an undefeated argument that attacks the original argument and it is also considered *better* (using some particular comparison criterion) than the one supporting the original claim.

Attacks can be directed to the claim, on internal points of an argument, or its premises if they are presumptions of any form.

The full analysis leads to the construction of a *dialectical tree* with the original argument in the root and whose nodes are arguments that can be marked defeated or undefeated, thus determining if the root argument is a warrant or not.

Four possible answers are possible for a query: Yes, if the posed query is warranted; No, if the complement of the query is warranted; Undecided, if neither the query nor its complement is warranted; and Unknown, when the query is not in the signature of the program.

(4) **Deductive argumentation**

> This is a logic-based approach to structured argumentation. Each argument is a pair (X, p) where X is a set of logical formulas, called the *premises*, that entails p, the *claim*, where entailment is specified by the choice of base logic (e.g. classical logic, modal logic, description logic, temporal logic, conditional logic, etc).
>
> Various options are available for specifying when one argument attacks another (e.g. argument a *rebuts* argument b when the claim of a is the negation of the claim of b).
>
> If we are then to construct an *argument graph* based on the arguments that can be constructed from a knowledge base, there are various options available for selecting which arguments and attacks to include in the argument graph (e.g. being exhaustive in including all arguments and counterarguments that can be constructed from a knowledge base).

The above four approaches have many things in common but also differ at several points.

6 The ASPIC+ framework

The ASPIC+ framework is based on two ideas:

the first is that **conflicts** between arguments are often resolved with *explicit preferences*,

and the second is that arguments are built with two kinds of inference rules: *strict*, or *deductive rules*, whose premises guarantee their conclusion, and *defeasible rules*, whose premises only create a presumption in favour of their conclusion.

Accordingly, arguments can in ASPIC+ be attacked in three ways: on their uncertain premises, or on their defeasible inferences, or on the conclusions of their defeasible inferences.

ASPIC+ is not a system but a framework for specifying systems. A main objective of the study of the ASPIC+ framework is to identify conditions under which instantiations of the framework satisfy logical consistency and closure properties.

. . . .

ASPIC+ is meant to generate abstract argumentation frameworks (AFs) in the sense of Dung (1995).

Historically, the ASPIC+ framework originates from the European ASPIC project that ran from 2004 to 2007,

how can arguments be built, i.e. how can claims be supported with grounds,

how can arguments be attacked?

ASPIC+ arguments can be constructed from *fallible* and *infallible* premises (respectively called *ordinary* and *axiom* premises), and strict and defeasible inference rules, and that arguments can be attacked on their ordinary premises, the conclusions of defeasible inference rules, and the defeasible inference steps themselves.

a key feature of the ASPIC+ framework is that it accommodates the use of *preferences* over arguments: an attack from one argument to another only succeeds (as a defeat) if the attacked argument is not stronger than (strictly preferred to) the attacking argument, according to some given preference relation.

6.1 Formal definition

An AF is a pair $(Arg, \rightharpoonup)$ where Arg is a set of arguments and $\rightharpoonup$ is a binary relation of *defeat*. We say that a *strictly defeats* b if $a \rightharpoonup b$ while b does not defeat a.

Definition 6.1 (Argumentation system) An *argumentation system* is a triple

$$AS = (\mathcal{L}, \mathcal{R}, n)$$

where

(1) $\mathcal{L}$ is a logical language closed under neg

$\dashv$

7 Assumption–based Argumentation, ABA

Assumption–Based Argumentation (ABA) is a form of **structured argumentation**, the notions of argument and attack are not primitive but are instead defined in terms of other notions, these notions are *rules* (in an underlying deductive system), *assignments* and their *contraries*.

ABA is an instance of AA.

deductive system: $(\mathcal{L}, \mathcal{R})$ where $\mathcal{L}$ is a language (a set of sentences) and $\mathcal{R}$ a set of *inference rules* that induces a derivability relation $\vdash$ on $\mathcal{L}$.

$Th(T) = \{\phi \in \mathcal{L} \mid T \vdash \phi\}$

An *assumption–based argumentation framework* (ABA framework) is a tuple

$$\langle \mathcal{L}, \mathcal{R}, A, ^- \rangle$$

where

- $\langle \mathcal{L}, \mathcal{R} \rangle$ is a deductive system;
- $A \subseteq \mathcal{L}$ is a non–empty finite set of *assumptions*;
- $\bar{(\cdot)}$ is a total mapping from A to $\mathcal{L}$, called the *contrary function*, $\bar{a}$ is referred to as the *contrary* of a for $a \in A$.

Each rule in $\mathcal{R}$ with a *head* and a *body*, where the head is a sentence in $\mathcal{L}$ and the body consists of $m \geq 0$ sentences in $\mathcal{L}$. Rules can be written in different formats, e.g., a rule with head σ_0 and body $\sigma_1, \ldots, \sigma_m$ can be written as

$$\sigma_0 \leftarrow \sigma_1, \ldots, \sigma_m \qquad \text{or} \qquad \frac{\sigma_1, \ldots, \sigma_m}{\sigma_0}.$$

Appendix

A Probability Measures and Distributions

Let S be a nonempty set and $\mathcal{A}$ an algebra of subsets of S, i.e., a set of subseteq of S s.t. (i) $S \in \mathcal{A}$, and (ii) if $H_1, H_2 \in \mathcal{A}$ then $\overline{H}_1, H_1 \cup H_2 \in \mathcal{A}$, where $\overline{H}_1 = S \setminus H_1$.

A *(finitely additive) probability measure* is a function $p\colon \mathcal{A} \to [0,1]$ such that:

(1) $p(S) = 1$,

(2) $p(H_1 \cup H_2) = p(H_1) + p(H_2)$, whenever $H_1 \cap H_2 = \emptyset$.

The triple $(S, \mathcal{A}, p)$ is called a *probability space*, and elements of $\mathcal{A}$ are called *measurable sets.*

For any probability measure p and $H, A \in \mathcal{A}$ such that $p(H) > 0$, the *conditional probability* $p(A|H)$ is defined in the usual way, that is,

$$p(A|H) = \frac{p(A \cap H)}{p(H)}.$$

It is known that the function $p(\cdot|H)$ is also a probability measure.

$$\boxed{\text{Bayes' Theorem: } p(A|B) = \frac{p(B|A) \cdot p(A)}{p(B)}}$$

For a finite set S, a *probability distribution* is any function $d\colon S \to [0,1]$ such that

$$\sum_{s \in S} d(s) = 1.$$

Every probability distribution d on a finite set S induces a function p on the $\wp(S)$:

$$p(H) = \sum_{s \in H} d(s),$$

which is a (finitely additive) probability measure. On the other hand, any probability measure p on the power set of S induces a distribution $d(s) = p(\{s\})$, thus, on finite sets “measures = distributions”.

References

[1] Gianvincenzo Alfano, Marco Calautti, Sergio Greco, Francesco Parisi, and Irina Trubitsyna. Explainable acceptance in probabilistic abstract argumentation: Complexity and approximation. In *Proceedings of the 17th International Conference on Principles of Knowledge Representation and Reasoning (KR2020)*, volume 17, pages 33–43, 2020. `doi:10.24963/kr.2020/4`.

[2] Pietro Baroni, Martin Caminada, and Massimiliano Giacomin. Abstract argumentation frameworks and their semantics. In Baroni et al. [4]. URL: `https://www.collegepublications.co.uk/handbooks/?00003`.

[3] Pietro Baroni, Eduardo Fermé, Massimiliano Giacomin, and Guillermo Ricardo Simari. Belief revision and computational argumentation: A critical comparison. *Journal of Logic, Language and Information*, 31(4):555–589, December 2022. `doi:10.1007/s10849-022-09369-8`.

[4] Pietro Baroni, Dov M. Gabbay, Massimiliano Giacomin, and Leendert van der Torre, editors. *Handbook of Formal Argumentation*, volume 1. College Publications, 2018. URL: `https://www.collegepublications.co.uk/handbooks/?00003`.

[5] Pietro Baroni and Massimiliano Giacomin. Semantics of abstract argument systems. In Rahwan and Simari [18], pages 25–44. `doi:10.1007/978-0-387-98197-0_2`.

[6] Richard Booth, Souhila Kaci, Tjitze Rienstra, and Leendert van der Torre. A logical theory about dynamics in abstract argumentation. In Weiru Liu, V. S. Subrahmanian, and Jef Wijsen, editors, *Scalable Uncertainty Management*, pages 148–161, Berlin, Heidelberg, 2013. Springer. `doi:10.1007/978-3-642-40381-1_12`.

[7] Sylvie Coste-Marquis, Sébastien Konieczny, Jean-Guy Mailly, and Pierre Marquis. Extension enforcement in abstract argumentation as an optimization problem. In *Proceedings of the 24th International Conference on Artificial Intelligence (IJCAI'15)*, pages 2876–2882. AAAI Press, July 2015.

[8] Martin Diller, Adrian Haret, Thomas Linsbichler, Stefan Rümmele, and Stefan Woltran. An extension-based approach to belief revision in abstract argumentation. *International Journal of Approximate Reasoning*, 93:395–423, February 2018. `doi:10.1016/j.ijar.2017.11.013`.

[9] 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. `doi:10.1016/0004-3702(94)00041-X`.

[10] Paul E. Dunne and Michael Wooldridge. Complexity of abstract argumentation. In Rahwan and Simari [18], pages 85–104. `doi:10.1007/978-0-387-98197-0_5`.

[11] W. Dvorak and Paul E. Dunne. Computational problems in formal argumentation and their complexity. In Baroni et al. [4]. URL: http://hdl.handle.net/20.500.12708/29850.

[12] Bettina Fazzinga, Sergio Flesca, and Francesco Parisi. On the complexity of probabilistic abstract argumentation frameworks. In *ACM Transactions on Computational Logic*, volume 16, pages 22:1–22:39, 2015. doi:10.1145/2749463.

[13] Anthony Hunter. Some foundations for probabilistic abstract argumentation. In *Proceedings of COMMA 2012*, Frontiers in Artificial Intelligence and Applications Vol. 245, pages 117–128. IOS Press, 2012. doi:10.3233/978-1-61499-111-3-117.

[14] Anthony Hunter. A probabilistic approach to modelling uncertain logical arguments. *International Journal of Approximate Reasoning*, 54(1):47–81, 2013. doi:10.1016/j.ijar.2012.08.003.

[15] Hengfei Li, Nir Oren, and Timothy J. Norman. Probabilistic argumentation frameworks. In Sanjay Modgil, Nir Oren, and Francesca Toni, editors, *Proc. of the 1st International Workshop on the Theorie and Applications of Formal Argumentation (TAFA 2011)*, LNAI 7132, pages 1–16, Berlin, Heidelberg, 2011. Springer. doi:10.1007/978-3-642-29184-5_1.

[16] Theofrastos Mantadelis and Stefano Bistarelli. Probabilistic abstract argumentation frameworks, a possible world view. *International Journal of Approximate Reasoning*, 119:204–219, April 2020. doi:10.1016/j.ijar.2019.12.006.

[17] Fabio Paglieri and Cristiano Castelfranchi. Revising beliefs through arguments: Bridging the gap between argumentation and belief revision in mas. In Iyad Rahwan, Pavlos Moraïtis, and Chris Reed, editors, *Argumentation in Multi-Agent Systems*, pages 78–94, Berlin, Heidelberg, 2005. Springer. doi:10.1007/978-3-540-32261-0_6.

[18] Iyad Rahwan and Guillermo R. Simari, editors. *Argumentation in Artificial Intelligence*. Springer, New York, 2009. doi:10.1007/978-0-387-98197-0.