TPLP **24** (4): 921–941, 2025. © The Author(s), 2025. Published by Cambridge University Press. This is an Open Access article, distributed under the terms of the Creative Commons Attribution licence (http://creativecommons.org/licenses/by/4.0/), which permits unrestricted re-use, distribution and reproduction, provided the original article is properly cited.
doi:10.1017/S1471068424000310

Cyclic Supports in Recursive Bipolar Argumentation Frameworks: Semantics and LP Mapping∗

GIANVINCENZO ALFANO, SERGIO GRECO,
FRANCESCO PARISI and IRINA TRUBITSYNA
DIMES Department, University of Calabria, Rende, Italy,
(*e-mails:* g.alfano@dimes.unical.it, greco@dimes.unical.it,
fparisi@dimes.unical.it, i.trubitsyna@dimes.unical.it)

submitted 14 August 2024; accepted 13 September 2024

Abstract

Dung's abstract Argumentation Framework (AF) has emerged as a key formalism for argumentation in artificial intelligence. It has been extended in several directions, including the possibility to express supports, leading to the development of the Bipolar Argumentation Framework (BAF), and recursive attacks and supports, resulting in the Recursive BAF (Rec-BAF). Different interpretations of supports have been proposed, whereas for Rec-BAF (where the target of attacks and supports may also be attacks and supports) even different semantics for attacks have been defined. However, the semantics of these frameworks have either not been defined in the presence of support cycles or are often quite intricate in terms of the involved definitions. We encompass this limitation and present classical semantics for general BAF and Rec-BAF and show that the semantics for specific BAF and Rec-BAF frameworks can be defined by very simple and intuitive modifications of that defined for the case of AF. This is achieved by providing a modular definition of the sets of defeated and acceptable elements for each AF-based framework. We also characterize, in an elegant and uniform way, the semantics of general BAF and Rec-BAF in terms of logic programming and partial stable model semantics.

Keywords: abstract argumentation, argumentation semantics, partial stable models

1 Introduction

Formal argumentation has emerged as one of the important fields in Artificial Intelligence (Rahwan and Simari, 2009). In particular, Dung's abstract Argumentation Framework (AF) is a simple, yet powerful formalism for modeling disputes between two or more agents (Dung 1995). An AF consists of a set of *arguments* and a binary *attack* relation over the set of arguments that specifies the *interactions* between arguments: intuitively,

∗ We acknowledge the support from project Tech4You (ECS0000009) and PNRR MUR projects FAIR (PE0000013) and SERICS (PE00000014).

https://doi.org/10.1017/S1471068424000310 Published online by Cambridge University Press

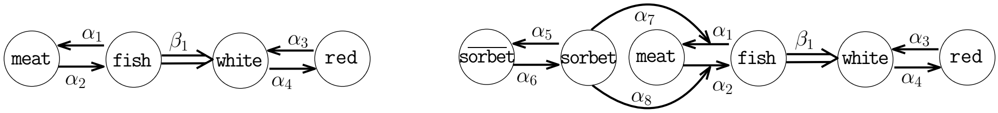


Fig 1. BAF (left) and Rec-BAF (right) of Example 1.

if argument a attacks argument b, then b is acceptable only if a is not. Hence, arguments are abstract entities whose role is entirely determined by the interactions specified by the attack relation.

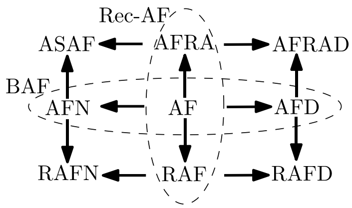
Dung's framework has been extended in many different ways, including the introduction of new kinds of interactions between arguments and/or attacks. In particular, the class of Bipolar Argumentation Frameworks (BAFs) is an interesting extension of AF that allows for also modelling *supports* between arguments (Nouioua and Risch 2011; Villata *et al.* 2012). Different interpretations of supports have been proposed in the literature. Further extensions consider second-order interactions (Villata *et al.* 2012), for example, attacks to attacks/supports, as well as more general forms of interactions such as recursive AFs where attacks can be recursively attacked (Baroni *et al.* 2011; Cayrol *et al.* 2017) and recursive BAFs, where attacks/supports can be recursively attacked/supported (Gottifredi *et al.* 2018). The next example shows two extensions of AF, whereas an overview of the extensions of AF studied in this paper is provided at the end of this section.

Example 1.
Consider a scenario regarding a restaurant and the possible menus to be suggested to customers. The arguments denote elements that are available and that can occur in a menu, whereas attacks denote incompatibility among elements. Consider the scenario shown in Figure 1 (left), where the attacks α_i $(1 \le i \le 4)$ denote incompatibilities, whereas the support β_1 from `fish` to `white` states that a menu may have white wine only if it also contains fish. Regarding the attacks, they state that the menu cannot contain both meat and fish as well as both white and red wine.

Assume now there exist two arguments `sorbet` and $\overline{\texttt{sorbet}}$ (no sorbet), attacking each other, stating that the menu may contain or not contain the sorbet. The resulting framework is shown in Figure 1 (right). The (recursive) attacks α_7 and α_8 from `sorbet` to α_1 and α_2 state that if sorbet is in the menu, then the incompatibility between fish and meat is not valid anymore.

While the semantics of argumentation is universally accepted, for the frameworks extending AF with supports and recursive attacks/supports, several interpretations of their role have been proposed, giving rise to different semantics (Rahwan and Simari 2009; Cohen *et al.* 2014; Cayrol *et al.* 2021). Following Dung's approach, the meaning of recursive AF-based frameworks is still given by relying on the concept of extension. However, the extensions of an *AF with recursive attacks and supports* also include the (names of) attacks and supports, which intuitively contribute to determine the set of accepted arguments (Cayrol *et al.* 2017, 2018).

https://doi.org/10.1017/S1471068424000310 Published online by Cambridge University Press



Legend of Acronyms:
AF: abstract Argumentation Framework [Dung,1995]
BAF: Bipolar AF
AFN: AF with Necessities [Nouia and Risch,2011]
AFD: AF with Deductive supports [Villata et al.,2012]
Rec-AF: Recursive-AF
AFRA: AF with Recursive Attacks [Baroni et al.,2011]
RAF: Recursive AF [Cayrol et al.,2017]
Rec-BAF: Recursive-BAF
ASAF: Attack-Support AF [Gottifredi et al.,2018]
RAFN: Recursive AF with Necessities [Cayrol et al.,2018]
AFRAD: AF with Rec. Att. and Ded. supports [Alfano et al.,2020]
RAFD: Recursive AF with Deductive supports [Alfano et al.,2020]

Fig 2. AF-based frameworks investigated in the paper. Rec-BAF frameworks are in the corners.

1.1 AF-based frameworks

Among the different frameworks extending AF some of them share the same structure, although they have different semantics. Thus, in the following, we distinguish between framework and *class* of frameworks. Two frameworks sharing the same syntax (i.e., the structure) belong to the same (syntactic) class. For instance, BAF is a syntactic class, whereas Argumentation Framework with Necessities (AFN) and Argumentation Framework with Deductive Supports (AFD) are two specific frameworks sharing the same BAF syntax; their semantics differ because they interpret supports in different ways.

Figure 2 overviews the frameworks extending AF studied in this paper. Horizontal arrows denote the addition of supports with two different semantics (necessary semantics in the left direction and deductive semantics in the right direction), whereas vertical arrows denote the extension with recursive interactions (i.e., attacks and supports); the two directions denote two different semantics proposed in the literature for determining the acceptance status of attacks. The frameworks Attack-Support Argumentation Framework (ASAF) and Recursive Argumentation Framework with Necessities (RAFN) (as well as Argumentation Framework with Recursive Attacks-Supports (AFRAD) and Recursive Argumentation Framework with Deductive Supports (RAFD)) belong to the same class, called *Recursive BAF* (Rec-BAF), combining AFN (resp., AFD) with AFRA and RAF, respectively. The differences between ASAF and RAFN (resp., AFRAD and RAFD) semantics are not in the way they interpret supports, both based on the necessity (resp., deductive) interpretation, but in a different determination of the status of attacks as they extend AFRA and RAF, respectively. Clearly, the frameworks in the corners are the most general ones. However, for the sake of presentation, before considering the most general frameworks, we also analyze the case of BAFs.

So far, the semantics of BAF and Rec-BAF frameworks has been defined only for restricted classes, called *acyclic*. Although recently two new semantics have been proposed for general AFN and RAFN, these semantics extending the ones defined for acyclic frameworks are quite involved. Thus, in this paper, we propose simple extensions of the semantics defined for acyclic BAF and Rec-BAF that apply to all specific frameworks in these classes (e.g., AFN, AFD, ASAF, RAFN, etc.) and show that, as for the acyclic case, the semantics of a framework can be also computed by considering an "equivalent" logic program under partial stable model semantics.

https://doi.org/10.1017/S1471068424000310 Published online by Cambridge University Press

1.2 Contributions

The main contributions are as follows:

- We propose classical semantics for general BAF and Rec-BAF and show that the semantics for specific BAF and Rec-BAF frameworks can be defined by very simple and intuitive modifications of that defined for the case of AF;
- We show that any general (Rec-)BAF Δ can be mapped into an equivalent logic program P_Δ under (partial) stable model semantics;
- We investigate the complexity of the verification and credulous and skeptical acceptance problems for (Rec-)BAF under semantics $\sigma \in \{\texttt{gr}, \texttt{co}, \texttt{st}, \texttt{pr}\}$. It turns out that BAF and Rec-BAF are as expressive as AF, though more general relations (i.e., supports and recursive relations) can be easily expressed.

2 Preliminaries

2.1 Argumentation frameworks

An abstract *Argumentation Framework* (AF) is a pair $\langle A, \Omega \rangle$, where A is a set of *arguments* and $\Omega \subseteq A \times A$ is a set of *attacks*. An AF can be seen as a directed graph, whose nodes represent arguments and edges represent attacks; an attack $(a, b) \in \Omega$ from a to b is represented by $a \rightarrow b$.

Given an AF $\Delta = \langle A, \Omega \rangle$ and a set $\mathbf{S} \subseteq A$ of arguments, an argument $a \in A$ is said to be *i*) *defeated* w.r.t. $\mathbf{S}$ iff $\exists b \in \mathbf{S}$ such that $(b, a) \in \Omega$, and *ii*) *acceptable* w.r.t. $\mathbf{S}$ iff for every argument $b \in A$ with $(b, a) \in \Omega$, there is $c \in \mathbf{S}$ such that $(c, b) \in \Omega$. The sets of defeated and acceptable arguments w.r.t. $\mathbf{S}$ (where Δ is understood) are next defined.

Definition 1.
For any AF $\Delta = \langle A, \Omega \rangle$ and set of arguments $\mathbf{S} \subseteq A$, the set of arguments defeated by $\mathbf{S}$ and acceptable w.r.t. $\mathbf{S}$ is defined as follows:

- $\mathit{def}(\mathbf{S}) = \{a \in A \mid \exists\, b \in \mathbf{S} \,.\, b \rightarrow a\}$;
- $\mathit{acc}(\mathbf{S}) \ = \{a \in A \mid \forall\, b \in A \,.\, b \rightarrow a \text{ implies } b \in \mathit{def}(\mathbf{S})\}$.

Given an AF $\langle A, \Omega \rangle$, a set $\mathbf{S} \subseteq A$ of arguments is said to be *(i) conflict-free* iff $\mathbf{S} \cap \mathit{def}(\mathbf{S}) = \emptyset$ and *(ii) admissible* iff it is conflict-free and $\mathbf{S} \subseteq \mathit{acc}(\mathbf{S})$. Moreover, $\mathbf{S}$ is a

- *complete extension* iff it is conflict-free and $\mathbf{S} = \mathit{acc}(\mathbf{S})$;
- *preferred extension* iff it is a $\subseteq$-maximal complete extension;
- *stable extension* iff it is a total (i.e., $\mathbf{S} \cup \mathit{def}(\mathbf{S}) = A$) preferred extension;
- *grounded extension* iff it is the $\subseteq$-smallest complete extension.

The set of complete (resp., preferred, stable, grounded) extensions of a framework Δ will be denoted by $\texttt{co}(\Delta)$ (resp., $\texttt{pr}(\Delta)$, $\texttt{st}(\Delta)$, $\texttt{gr}(\Delta)$). We often denote extensions by pairs denoting both accepted and defeated elements using the notation $\widehat{\sigma}(\Delta)$ instead of $\sigma(\Delta)$ (with $\sigma \in \{\texttt{co}, \texttt{st}, \texttt{pr}, \texttt{gr}\}$), where $\widehat{\sigma}(\Delta) = \{\widehat{\mathbf{S}} = \langle \mathbf{S}, \mathit{def}(\mathbf{S}) \rangle \mid \mathbf{S} \in \sigma(\Delta)\}$.

https://doi.org/10.1017/S1471068424000310 Published online by Cambridge University Press

Example 2.
Let $\Delta = \langle A, \Omega \rangle$ be an AF, where $A = \{\mathtt{a,\ b,\ c,\ d}\}$ and $\Omega = \{\mathtt{(a,b),\ (b,a),\ (a,c),\ (b,c),\ (c,d),\ (d,c)}\}$. The set of complete extensions of Δ is $\mathtt{co}(\Delta) = \{\emptyset, \{\mathtt{d}\}, \{\mathtt{a,\ d}\}, \{\mathtt{b,\ d}\}\}$. Consequently, $\mathtt{pr}(\Delta) = \mathtt{st}(\Delta) = \{\{\mathtt{a,\ d}\}, \{\mathtt{b,\ d}\}\}$, $\mathtt{gr}(\Delta) = \{\emptyset\}$.

Two main problems in formal argumentation are *verification* and *acceptance* (Dvorák *et al.* 2023; Fazzinga *et al.* 2022; Alfano *et al.* Alfano et al., 2023, 2023a,b,c,d, 2024b,c; Dvorák *et al.* 2022,2024). Given an AF $\Delta = \langle A, \Omega \rangle$, a set $S \subseteq A$ of arguments, and a semantics $\sigma \in \{\mathtt{gr}, \mathtt{co}, \mathtt{st}, \mathtt{pr}\}$, the *verification* problem is the problem of deciding S is a σ-extension of Δ (i.e., $S \in \sigma(\Delta)$). Given an AF $\Delta = \langle A, \Omega \rangle$, an argument $a \in A$, and a semantics $\sigma \in \{\mathtt{gr}, \mathtt{co}, \mathtt{st}, \mathtt{pr}\}$, the *credulous (*resp., *skeptical) acceptance* problem is the problem of deciding whether argument a is credulously (resp., skeptically) accepted under semantics σ, that is, deciding whether a belongs to at least one (resp., all) σ-extension of the framework Δ. Clearly, for the grounded semantics, which admits exactly one extension, these problems become identical. The complexity of acceptance and verification problems for AF has been thoroughly investigated (see Dvorák and Dunne (2017) for a survey). The semantics of an AF Δ can be also computed by considering the logic program P_Δ, under partial stable model (PSM) semantics, derived from $\Delta = \langle A, \Omega \rangle$ as follows:

$$P_\Delta = \left\{ a \leftarrow \bigwedge_{(b,a) \in \Omega} \neg b \mid a \in A \right\}. \tag{1}$$

Partial stable models are 3-valued. A partial interpretation M is a partial stable model of a ground program P if it is the minimal model of the program P^M obtained from replacing negated literals in the rules' body with their truth values in M. The set of partial stable models of a program P is denoted as $\mathcal{PS}(P)$. It has been shown that for any AF Δ, $\widehat{\mathtt{co}}(\Delta) = \mathcal{PS}(P_\Delta)$. More information on partial stable model semantics can be found in Appendix A.

2.2 Bipolar argumentation frameworks

A *Bipolar Argumentation Framework* (BAF) is a triple $\langle A, \Omega, \Gamma \rangle$, where A is a set of *arguments*, $\Omega \subseteq A \times A$ is a set of *attacks*, and $\Gamma \subseteq A \times A$ is a set of *supports*. A BAF can be represented by a directed graph with two types of edges: *attacks* and *supports*, denoted by $\rightarrow$ and $\Rightarrow$, respectively. A *support path* $a_0 \overset{+}{\Rightarrow} a_n$ from argument a_0 to argument a_n is a sequence of n edges $a_{i-1} \Rightarrow a_i$ with $0 < i \leq n$. We use $\Gamma^+ = \{(a, b) \mid a, b \in A \ \wedge a \overset{+}{\Rightarrow} b\}$ to denote the set of pairs (a, b) such that there exists a support path from a to b. A BAF is said to be acyclic if it does not have support cycles; that is, there is no argument a such that $a \overset{+}{\Rightarrow} a$.

Different interpretations of the support relation have been proposed in the literature (Rahwan and Simari 2009; Cayrol *et al.* 2021). In this paper, we concentrate on the necessity and deductive interpretations (Nouioua and Risch 2011; Villata *et al.* 2012). Necessary support is intended to capture the following intuition: if argument a supports argument b, then the acceptance of b implies the acceptance of a and the non-acceptance of a implies the non-acceptance of b, whereas the deductive interpretation states that the acceptance of a implies that b is also accepted.

https://doi.org/10.1017/S1471068424000310 Published online by Cambridge University Press

The two BAF frameworks obtained by interpreting the supports as necessities or as deductive are, respectively, called Argumentation Framework with Necessities (AFN) and Argumentation Framework with Deductive Supports (AFD).

It is worth noting that for AFN, the following implication holds: *(i)* $a \rightarrow b$ and $b \overset{+}{\Rightarrow} c$ imply $a \rightarrow c$ (called *secondary or derived attack*). Similarly, for AFD, the following implication holds: *(ii)* $a \rightarrow b$ and $c \overset{+}{\Rightarrow} b$ imply $a \rightarrow c$ (called *mediated or derived attack*). Considering Example 1, under the necessary interpretation of supports, there exists a secondary attack from `meat` to `white`.

There has been a long debate on the fact that the support relation should be acyclic as it is inherently transitive. On the other side, a framework represents a situation where several agents may act, and, thus, support cycles could be present. In this subsection we assume that the support relation is acyclic, as originally defined by Nouioua and Risch (2011). A semantics for the general AFN, albeit a bit involved, has been recently proposed by Nouioua and Boutouhami (2023).

The aim of this paper, explored in Section 4, is the definition of intuitive semantics for general Recursive BAF, which extend both semantics defined for BAF and acyclic BAF. To this end, in the rest of this subsection, we discuss semantics for acyclic BAF, whereas in Section 3, we present an intuitive semantics for general BAF. For BAF with supports interpreted as necessities (AFN), the new semantics is equivalent to that proposed by Nouioua and Boutouhami (2023).

2.2.1 AF with necessary supports (AFN)

The semantics of AFN with acyclic supports can be defined by first redefining the definition of defeated and acceptable sets as follows.

Definition 2.
For any AFN $\langle A, \Omega, \Gamma \rangle$ and set of arguments $\mathbf{S} \subseteq A$:

- $def(\mathbf{S}) = \{a \in A \mid (\exists b \in \mathbf{S} \,.\, b \rightarrow a) \vee (\exists c \in def(\mathbf{S}) \,.\, c \Rightarrow a\}$;
- $acc(\mathbf{S}) = \{a \in A | (\forall b \in A . b \rightarrow a \text{ implies } b \in def(\mathbf{S})) \wedge (\forall c \in A .\, c \Rightarrow a \text{ implies } c \in acc(\mathbf{S})\}$.

It is worth noting that $def(\mathbf{S})$ and $acc(\mathbf{S})$ are defined recursively. The definitions of conflict-free and admissible sets, as well as the definitions of complete, preferred, stable, and grounded extensions are the same as those introduced for AF.

Example 3.
Let $\Delta = \langle$\{`fish, meat, white, red`\}, \{`(fish,meat), (meat,fish), (white,red), (red,white)`\}, \{`(fish,white)`\}$\rangle$ be the AFN shown on Figure 1(left). Then, def(\{`fish,white`\}) = \{`meat,red`\} and acc(\{`fish,white`\}) = \{`fish,white`\}. Δ has six complete extensions, which are $E_0 = \emptyset$, $E_1 =$ \{`red`\}, $E_2 =$ \{`fish`\}, $E_3 =$ \{`fish, red`\}, $E_4 =$ \{`fish, white`\}, and $E_5 =$ \{`meat,red`\}. Moreover, E_3, E_4, and E_5 are preferred (and also stable), while E_0 is the grounded extension.

An alternative and equivalent way to define the semantics of acyclic AFN is to rewrite them in terms of "equivalent" AF, by introducing secondary attacks and deleting supports. The derived AF of the AFN $\langle A, \Omega, \Gamma \rangle$ of Example 3 is $\langle A, \Omega \cup$ \{`(meat,white)`\}$\rangle$.

https://doi.org/10.1017/S1471068424000310 Published online by Cambridge University Press

2.2.2 AF with deductive supports (AFD)

The semantics of AFD is dual w.r.t. that of AFN; that is, we can transform a BAF with deductive supports into an equivalent BAF with necessary supports by simply reversing the direction of the support arrows (Cayrol *et al.* 2021). Equivalently, the semantics of acyclic AFD can be defined in terms of AF by adding mediated attacks and removing the supports. Alternatively, it can be presented as in Definition 2 by replacing the support $c \Rightarrow a$ with the support $a \Rightarrow c$.

2.3 Recursive BAF

By combining the concepts of bipolarity and recursive interactions, more general argumentation frameworks have been defined. A *Recursive Bipolar Argumentation Framework (Rec-BAF)* is a tuple $\langle A, R, T, \mathbf{s}, \mathbf{t} \rangle$, where A is a set of arguments, R is a set of attack names, T is a set of support names, and $\mathbf{s}$ (resp., $\mathbf{t}$) is a function from $R \cup T$ to A (resp., to $A \cup R \cup T$), which is mapping each attack/support to its source (resp., target).

Considering AF with recursive attacks, that is, AF in which the target of an attack can also be an attack, two different semantics have been defined, giving rise to two specific frameworks: Recursive Abstract Argumentation Framework (RAF) (Cayrol *et al.* 2017) and Abstract Argumentation Framework with Recursive Attacks (AFRA) (Baroni *et al.* 2011). We do not further discuss these two frameworks as they are just special cases of Rec-BAF. We mention them only for the fact that they were proposed before Rec-BAF frameworks, and the differences in their semantics, combined with different interpretations of supports, give rise to different specific Rec-BAF frameworks, namely, the ones appearing at the corner of Figure 2. We first discuss the two frameworks where supports are interpreted as necessities: RAFN (extending RAF) (Cayrol *et al.* 2018) and ASAF (extending AFRA) (Gottifredi *et al.* 2018). Regarding the extensions of RAF and AFRA with deductive supports, called Recursive Argumentation Framework with Deductive Supports (RAFD), and Argumentation Framework with Recursive Attacks-Supports (AFRAD) (Alfano *et al.* 2020b), their semantics will be recalled at the end of this section.

We now discuss semantics for "acyclic" Rec-BAF, although recently there have been two contributions defining the semantics of general RAF and RAFN (Lagasquie-Schiex 2023). We do not further discuss these semantics as they are quite involved, and in the next two sections, we present our main contribution, consisting of the definition of new semantics for general BAF and Rec-BAF, extending in a natural way the semantics defined for acyclic BAF and acyclic Rec-BAF. As the underlying structure representing a Rec-BAF is not a graph, the definition of (a)cyclicity has been formulated in terms of the acyclicity of the BAF obtained by replacing every attack $a \overset{\alpha}{\rightarrow} b$ with $a \Rightarrow \alpha$ and $\alpha \rightarrow b$ and every support $a \overset{\beta}{\Rightarrow} b$ with $a \Rightarrow \beta$ and $\beta \Rightarrow b$. Note that the so-obtained auxiliary BAF is only used to formally define and check acyclicity in Rec-BAF, not to provide the semantics, which is instead recalled next.

2.3.1 Recursive AF with necessities (RAFN)

The RAFN framework has been proposed by Cayrol *et al.* (2018). The semantics combines the RAF interpretation of attacks in RAF with the necessary interpretation of supports

https://doi.org/10.1017/S1471068424000310 Published online by Cambridge University Press

of AFN. Here we consider a simplified version, where supports have a single source and the support relation is acyclic. Differences between RAFN and ASAF semantics are highlighted in blue.

Definition 3.
For any acyclic RAFN $\langle A, \Sigma, T, \mathbf{s}, \mathbf{t}\rangle$ and set $\mathbf{S} \subseteq A \cup R \cup T$, we have that:

- $def(\mathbf{S}) = \{X \in A \cup R \cup T \mid (\exists \alpha \in R \cap \mathbf{S}\,.\,\mathbf{t}(\alpha) = X \wedge \mathbf{s}(\alpha) \in \mathbf{S}) \vee (\exists \beta \in T \cap \mathbf{S}\,.\,\mathbf{t}(\beta) = X \wedge \mathbf{s}(\beta) \in def(\mathbf{S}))\}$;
- $acc(\mathbf{S}) = \{X \in A \cup R \cup T \mid (\forall \alpha \in R\,.\,\mathbf{t}(\alpha) = X \text{ implies } (\alpha \in def(\mathbf{S}) \vee \mathbf{s}(\alpha) \in def(\mathbf{S}))) \wedge (\forall \beta \in T\,.\,\mathbf{t}(\beta) = X \text{ implies } (\beta \in def(\mathbf{S}) \vee \mathbf{s}(\beta) \in acc(\mathbf{S})))\}$.

2.3.2 Attack-support AF (ASAF)

The semantics combines the AFRA interpretation of attacks with that of BAF under the necessary interpretation of supports (i.e., AFN). For the sake of presentation, we refer to the formulation presented by Alfano *et al.* (Alfano et al., 2020a, 2024a), where attack and support names are first-class citizens, giving the possibility to represent multiple attacks and supports from the same source to the same target. For any ASAF Δ and $\mathbf{S} \subseteq A \cup R \cup T$, the *defeated* and *acceptable* sets (given $\mathbf{S}$) are defined as follows.

Definition 4.
Given an acyclic ASAF $\langle A, R, T, \mathbf{s}, \mathbf{t}\rangle$ and a set $\mathbf{S} \subseteq A \cup R \cup T$, we define:

- $def(\mathbf{S}) = \{X \in A \cup R \cup T \mid (X \in R \wedge \mathbf{s}(X) \in def(\mathbf{S})) \vee (\exists \alpha \in R \cap \mathbf{S}\,.\,\mathbf{t}(\alpha) = X) \vee (\exists \beta \in T \cap \mathbf{S}\,.\,\mathbf{t}(\beta) = X \wedge \mathbf{s}(\beta) \in def(\mathbf{S}))\}$;
- $acc(\mathbf{S}) = \{X \in A \cup R \cup T \mid (X \in R \text{ implies } \mathbf{s}(X) \in acc(\mathbf{S})) \wedge (\forall \alpha \in R.\,\mathbf{t}(\alpha) = X \text{ implies } \alpha \in def(\mathbf{S})) \wedge (\forall \beta \in T\,.\,\mathbf{t}(\beta) = X \text{ implies } (\beta \in def(\mathbf{S}) \vee \mathbf{s}(\beta) \in acc(\mathbf{S})))\}$.

Again, the notions of *conflict-free*, *admissible sets*, and the different types of extensions can be defined in a standard way (see Section 2.1) by considering $\mathbf{S} \subseteq A \cup R \cup T$ and by using the definitions of defeated and acceptable sets reported above. It is worth noting that the differences between ASAF and RAFN semantics (highlighted in blue) are not in the way they interpret supports (both based on the necessity interpretation) but in a different determination of the status of attacks as they extend AFRA and RAF, respectively. Moreover, for each semantics, the RAFN extensions can be derived from the corresponding ASAF extensions and vice versa.

Example 4.
Consider the Rec-BAF Δ with the necessary supports of Figure 1(right), and assume arguments are denoted by their initials. The preferred (and also stable) extensions prescribing sorbet in the menu under RAFN (resp., ASAF) semantics are $E_0 = \{\texttt{s},\texttt{m},\texttt{f},\texttt{w}, \beta_1, \alpha_3, \ldots, \alpha_8\}$ and $E_1 = \{\texttt{s},\texttt{m},\texttt{f},\texttt{r}, \beta_1, \alpha_3, \ldots, \alpha_8\}$, (resp., $E'_0 = \{\texttt{s},\texttt{m},\texttt{f},\texttt{w}, \beta_1, \alpha_4, \alpha_6, \alpha_7, \alpha_8\}$ and $E'_1 = \{\texttt{s},\texttt{m},\texttt{f},\texttt{r}, \beta_1, \alpha_3, \alpha_6, \alpha_7, \alpha_8\}$). Note that under ASAF semantics, attacks α_3 (resp., α_4) and α_6 are not part of the extension E'_0 (resp., E'_1) as their sources (i.e., $\texttt{r}$, $\texttt{w}$, and $\bar{\texttt{s}}$, respectively) are defeated.

https://doi.org/10.1017/S1471068424000310 Published online by Cambridge University Press

2.3.3 Recursive AF with deductive supports (RAFD)

For BAFs, necessary support and deductive support are dual (i.e., it is possible to transform a BAF with necessity into an equivalent BAF with deductive supports by simply reversing the direction of the support arrows) (Cayrol *et al.* 2021). However, in the case of Rec-BAFs that are not BAFs, this duality no longer holds. This happens because the target of supports and attacks in Rec-BAF may also be other supports and attacks. For this reason, we next explicitly recall the semantics for Rec-BAF with deductive supports.

Definition 5.
For any acyclic RAFD $\langle A, R, T, \mathbf{s}, \mathbf{t}\rangle$ and set $\mathbf{S} \subseteq A \cup R \cup T$, we have that:

- $\mathit{def}(\mathbf{S}) = \{X \in A \cup R \cup T \mid (\exists \alpha \in R \cap \mathbf{S} \,.\, \mathbf{t}(\alpha) = X \wedge \mathbf{s}(\alpha) \in \mathbf{S}) \vee$
  $(\exists \beta \in T \cap \mathbf{S} \,.\, \mathbf{s}(\beta) = X \wedge \mathbf{t}(\beta) \in \mathit{def}(\mathbf{S}))\}$;
- $\mathit{acc}(\mathbf{S}) = \{X \in A \cup R \cup T \mid (\forall \alpha \in R \,.\, \mathbf{t}(\alpha) = X \text{ implies } (\alpha \in \mathit{def}(\mathbf{S}) \vee \mathbf{s}(\alpha) \in \mathit{def}(\mathbf{S}))) \wedge$
  $(\forall \beta \in T \,.\, \mathbf{s}(\beta) = X \text{ implies } (\beta \in \mathit{def}(\mathbf{S}) \vee \mathbf{t}(\beta) \in \mathit{acc}(\mathbf{S})))\}$.

We have highlighted in blue the differences between the RAFN and the RAFD definitions of defeated and acceptable arguments.

2.3.4 AF with recursive attacks and deductive supports (AFRAD)

Definition 6.
Given an acyclic AFRAD $\langle A, R, T, \mathbf{s}, \mathbf{t}\rangle$ and a set $\mathbf{S} \subseteq A \cup R \cup T$, we define:

- $\mathit{def}(\mathbf{S}) = \{X \in A \cup R \cup T \mid (X \in R \wedge \mathbf{s}(X) \in \mathit{def}(\mathbf{S})) \vee$
  $(\exists \alpha \in R \cap \mathbf{S} \,.\, \mathbf{t}(\alpha) = X) \vee$
  $(\exists \beta \in T \cap \mathbf{S} \,.\, \mathbf{s}(\beta) = X \wedge \mathbf{t}(\beta) \in \mathit{def}(\mathbf{S}))\}$;
- $\mathit{acc}(\mathbf{S}) = \{X \in A \cup R \cup T \mid (X \in R \text{ implies } \mathbf{s}(X) \in \mathit{acc}(\mathbf{S})) \wedge$
  $(\forall \alpha \in R \,.\, \mathbf{t}(\alpha) = X \text{ implies } \alpha \in \mathit{def}(\mathbf{S})) \wedge$
  $(\forall \beta \in T \,.\, \mathbf{s}(\beta) = X \text{ implies } (\beta \in \mathit{def}(\mathbf{S}) \vee \mathbf{t}(\beta) \in \mathit{acc}(\mathbf{S})))\}$.

Again, we have highlighted in blue the differences between the ASAF and the AFRAD definitions of defeated and acceptable arguments. Analogously to the case of RAFN versus ASAF, RAFD and AFRAD semantics may differ only in the status of attacks.

2.3.5 Mappings to other formalisms

It has been shown that any acyclic Rec-BAF Δ can be mapped to (i) an "equivalent" AF Λ so that $\mathsf{co}(\Delta) \equiv \mathsf{co}(\Lambda)$ (i.e., $\mathsf{co}(\Delta) = \mathsf{co}(\Lambda)$ modulo meta-arguments introduced in the rewriting), and (ii) a logic program P_Δ so that $\widehat{\mathsf{co}}(\Delta) = \mathcal{PS}(P_\Delta)$ (Alfano *et al.* 2020b). We next extend these results to general Rec-BAF.

3 A new semantics for BAF

In this section, we present a new and intuitive semantics for general BAF. In the rest of this section, whenever we refer to a BAF, we intend either an AFN (i.e., a BAF where supports are intended as necessities) or an AFD (where supports are intended as deductive).

https://doi.org/10.1017/S1471068424000310 Published online by Cambridge University Press

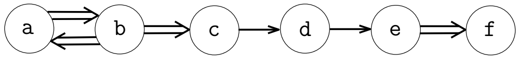


Fig 3. AFN Δ of Example 5.

We start by introducing new definitions for defeated and acceptable sets that extend the ones recalled in the previous section. To distinguish the defeated and acceptable sets defined for general BAF from those defined for acyclic BAF, we introduce new functions *DEF* and *ACC*.

Definition 7.
For any general AFN $\langle A, \Omega, \Gamma \rangle$ and set of arguments $\mathbf{S} \subseteq A$:

- $DEF(\mathbf{S}) = \{a \in A \mid (\exists b \in \mathbf{S} \,.\, b \to a) \vee (\exists c \in DEF(\mathbf{S}) \,.\, c \Rightarrow a) \vee a \overset{+}{\Rightarrow} a\}$;
- $ACC(\mathbf{S}) = \{a \in A \mid (\forall b \in A . b \to a \text{ implies } b \in DEF(\mathbf{S})) \wedge (\forall c \in A . c \Rightarrow a \text{ implies } c \in ACC(\mathbf{S})\}$.

As for the acyclic case, the semantics of general AFD is dual w.r.t. that of general AFN. Thus, we can transform any AFD into an equivalent AFN by reversing supports.

The main difference between the definitions of defeated and acceptable sets for acyclic and general BAF (highlighted in blue) consists in the fact that the new definition of acceptable set *ACC* explicitly excludes self-supported arguments (i.e., arguments in a support cycle or supported by an argument in a support cycle), which have been assumed to be defeated. This means that self-supported arguments are always defeated, independently from the specific set $\mathbf{S}$. Thus, to compute the BAF semantics, it is possible to first state that all arguments belonging to some support cycle are defeated (i.e., they are false arguments), and then, after removing them from the framework, we can compute the semantics of a support-acyclic BAF. Intuitively, this is in line with the self-supportless principle in logic programs that ensures no literal in an answer set is supported exclusively by itself or through a cycle of dependencies that lead back to itself. As an example, consider the BAF $\langle \{a\}, \emptyset, \{(a,a)\} \rangle$. According to our semantics, argument a is always defeated. Any alternative semantics prescribing a as true would be in contrast with the logic formulation of the AF, that according to our semantics, is rewritten as a rule $a \leftarrow a$ (literally, a if a), whose unique minimal model is $\emptyset$ (where a is false). Notably, as discussed in more detail in Section 6, this intuition is also reflected in the stable semantics of Abstract Dialectical Framework (ADF) (Brewka *et al*. 2013). We now give another example to illustrate our semantics.

Example 5.
Let $\Delta = \langle \{\mathtt{a,b,c,d,e,f}\}, \{\mathtt{(c,d)}, \mathtt{(d,e)}\}, \{\mathtt{(a,b)}, \mathtt{(b,a)}, \mathtt{(b,c)}, \mathtt{(e,f)}\} \rangle$ be the AFN shown on Figure 3. Then, $DEF(\{\mathtt{d}\}) = \{\mathtt{a,b,c,e,f}\}$ and $ACC(\{\mathtt{d}\}) = \{\mathtt{d}\}$. The AFN Δ has a unique complete extension $E = \{\mathtt{d}\}$ that is grounded, preferred, and stable.

As stated next, the semantics introduced extend those defined for acyclic BAF.

Proposition 1.
For any acyclic AFN $\Delta = \langle A, \Omega, \Gamma \rangle$ and semantics $\sigma \in \{\mathtt{co}, \mathtt{gr}, \mathtt{pr}, \mathtt{st}\}$, $\sigma(\langle A, \Omega, \Gamma \rangle)$ computed by means of Definition 7 coincides with $\sigma(\Delta)$ computed by means of Definition 2.

https://doi.org/10.1017/S1471068424000310 Published online by Cambridge University Press

Proof.
The result follows by observing that, for any set $\mathbf{S} \subseteq A$, we have that $DEF(\mathbf{S}) = def(\mathbf{S})$ and $ACC(\mathbf{S}) = acc(\mathbf{S})$ as there is no argument $a \in A$ s.t. $a \overset{\pm}{\Rightarrow} a$. $\square$

As AFD semantics is dual to the one of AFN, a similar result holds for acyclic AFD.

3.1 LP mapping

As done by Alfano *et al.* (2020b) for acyclic BAF, the semantics here presented can be defined even in terms of logic programs under partial stable model semantics.

To this end, we now provide the mappings from general BAF to propositional programs, so that the set of σ-extensions of any general BAF Δ is equivalent to that of partial stable models of the corresponding logic program P_Δ. The logic rules of P_Δ are derived from the topology of the AF. Basically, the rules in P_Δ extend the ones defined for AF (Equation 1) as the body of a rule defining an argument a also contains the (positive) conjunction of arguments supporting it.

Definition 8.
Given an AFN (resp., AFD) $\Delta = \langle A, \Omega, \Gamma \rangle$, then P_Δ (the propositional program derived from Δ) contains, for each argument $a \in A$, a rule

$$a \leftarrow \bigwedge_{(b,a)\in\Omega} \neg b \ \wedge \bigwedge_{(c,a)\in\Gamma} c \quad \left(\text{resp. } a \leftarrow \bigwedge_{(b,a)\in\Omega} \neg b \ \wedge \bigwedge_{(a,c)\in\Gamma} c \right). \tag{2}$$

Again, we have highlighted in blue the difference between the two mappings. The next theorem states the equivalence between (general) BAF Δ and logic program P_Δ under partial stable model semantics. As already stated, whenever we use the term BAF, we intend either AFN or AFD.

Theorem 1.
For any BAF Δ, $\widehat{\mathsf{co}}(\Delta) = \mathcal{PS}(P_\Delta)$.

Proof.
We denote with $sup(\Delta) = \{a | a \overset{\pm}{\Rightarrow} a\} \cup \{b | \exists a \in sup(\Delta).\ a \Rightarrow b\}$ when Δ is an AFN; $sup(\Delta) = \{a | a \overset{\pm}{\Rightarrow} a\} \cup \{b | \exists a \in sup(\Delta).\ b \Rightarrow a\}$ when Δ is an AFD.

- $\widehat{\mathsf{co}}(\Delta) \subseteq \mathcal{PS}(P_\Delta)$. We prove that for any $\widehat{\mathbf{S}} \in \widehat{\mathsf{co}}(\Delta)$, $\widehat{\mathbf{S}} \in \mathcal{PS}(P_\Delta)$. Indeed, P_Δ contains, for each atom $a \in A$, a rule $a \leftarrow \bigwedge_{(b,a)\in\Omega} \neg b \wedge \bigwedge_{(c,a)\in\Gamma} c$ (or $a \leftarrow \bigwedge_{(b,a)\in\Omega} \neg b \wedge \bigwedge_{(a,c)\in\Gamma} c$ if Δ is an AFD). Moreover, $P_\Delta^{\widehat{\mathbf{S}}}$ (the positive instantiation of P_Δ w.r.t. $\widehat{\mathbf{S}}$) contains positive rules defining exactly the arguments in $\mathbf{S}$, whose bodies contains only (positive) arguments in $\mathbf{S}$. Since the arguments $a \in A \cap sup(\Delta)$ do not appear in $\mathbf{S}$ by construction (they are false in the rules' body as appearing in $DEF(\mathbf{S})$), we have that $P_\Delta^{\widehat{\mathbf{S}}}$ does not contain cycles of positive literals, and thus $T^\omega_{P_\Delta^{\widehat{\mathbf{S}}}}(\emptyset) = \mathbf{S}$ and $\mathbf{S}$ is a PSM for $P_\Delta^{\widehat{\mathbf{S}}}$.
- $\mathcal{PS}(P_\Delta) \subseteq \widehat{\mathsf{co}}(\Delta)$. Consider a PSM $M \in \mathcal{PS}(P_\Delta)$, $pos(M) = T^\omega_{P_\Delta^M}(\emptyset)$. $pos(M) \subseteq A$ is conflict-free w.r.t. Δ. Indeed, assuming that there are two arguments $a, b \in pos(M)$ such that $(a, b) \in \Omega$, this means that the rule defining b in P_Δ contains in the body

https://doi.org/10.1017/S1471068424000310 Published online by Cambridge University Press

a literal $\neg a$. This is not possible as in such a case $b \notin T^{\omega}_{P^M_\Delta}(\emptyset)$. Assuming that Δ is an AFN, and that a attacks b indirectly through a supported attack $a \rightarrow a_1 \Rightarrow \cdots \Rightarrow a_n \Rightarrow b$. In such a case $a_1, \ldots, a_n, b \notin T^{\omega}_{P^M_\Delta}(\emptyset)$. Assuming that Δ is an AFD, and that a attacks b indirectly through a mediated attack $a \rightarrow a_1 \Leftarrow \cdots \Leftarrow a_n \Leftarrow b$. In such a case $a_1, \ldots, a_n, b \notin T^{\omega}_{P^M_\Delta}(\emptyset)$. Thus, $pos(M)$ is conflict-free. $pos(M) \subseteq A$ does not contain any argument $a \in sup(\Delta)$. Indeed, assuming for contradiction that such an argument $a \in pos(M)$ exists, then there must exist at least one other argument literal b in the body of the rule defining a s.t. (i) $b \in pos(M)$ and $b \in sup(\Delta)$. The same holds for any argument of the form of b. Thus, M is not a minimal model, contradicting the assumption. Moreover, from Definition 7, considering that $pos(M) = T^{\omega}_{P^M_\Delta}(\emptyset)$, we derive that $pos(M) = ACC(pos(M))$. □

The previous theorem states that the set of complete extensions of any BAF Δ coincides with the set of PSMs of the derived logic program P_Δ. Consequently, the set of stable and preferred extensions (resp., the grounded extension) coincide with the set of *total stable* and *maximal-stable* models (resp., the *well-founded* model) of P_Δ (Van Gelder *et al*. 1991; Saccà 1997).

Example 6.
The propositional program P_Δ derived from the AFN Δ of Example 5 is $\{\mathtt{a} \leftarrow \mathtt{b};\ \mathtt{b} \leftarrow \mathtt{a};\ \mathtt{c} \leftarrow \mathtt{b};\ \mathtt{e} \leftarrow \neg\mathtt{d};\ \mathtt{d} \leftarrow \neg\mathtt{c};\ \mathtt{f} \leftarrow \mathtt{e};\}$, where $\mathcal{PS}(P_\Delta) = \widehat{\mathtt{co}}(\Delta) = \{\{\neg\mathtt{a}, \neg\mathtt{b}, \neg\mathtt{c},\ \mathtt{d}, \neg\mathtt{e}, \neg\mathtt{f}\}\}$.

It is worth noting that, from the above result and the fact that any LP can be converted into an "equivalent" AF (Caminada *et al*. 2015), it is also possible to convert any BAF into an "equivalent" AF in polynomial time.

3.2 Computational complexity

The complexity of verification and acceptance problems for acyclic BAF are well-known and coincide with the corresponding ones on AF, as any acyclic BAF can be rewritten into an AF. In this section, we show that the same holds also for general BAF; that is, the complexities of the verification and acceptance problems for general BAF are the same as those known for AF.

Proposition 2.
For any BAF $\Delta = \langle A, \Omega, \Gamma\rangle$, *and semantics* $\sigma \in \{\mathtt{co}, \mathtt{gr}, \mathtt{pr}, \mathtt{st}\}$, *checking whether a set of arguments* $\mathbf{S} \subseteq A$ *is a* σ*-extension for* Δ *is (i) in PTIME for* $\sigma \in \{\mathtt{gr}, \mathtt{co}, \mathtt{st}\}$ *and (ii) coNP-complete for* $\sigma = \mathtt{pr}$.

Proof.
Lower bounds derive from the complexity results of the same problems for AF (Dvorák and Dunne, 2017), as BAF is a generalization of AF. As for upper bounds, the proof can be carried out by writing P_Δ (in PTIME) and check that $(P_\Delta, \widehat{\mathbf{S}} = \mathbf{S} \cup \{\neg x \mid x \in DEF(\mathbf{S})\}, \sigma^*)$ is a true instance of the verification problem in LP (Saccà 1997), where $\sigma^* = \mathtt{WF}$ (resp., $\mathcal{PS}, \mathcal{TS}, \mathcal{MS}$) iff $\sigma = \mathtt{gr}$ (resp., $\mathtt{co}$, $\mathtt{st}$, $\mathtt{pr}$). As for Theorem 1 we have that $\widehat{\mathtt{co}}(\Delta) = \mathcal{PS}(P_\Delta)$, the result follows. □

https://doi.org/10.1017/S1471068424000310 Published online by Cambridge University Press

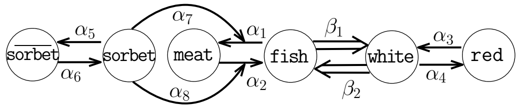


Fig 4. Rec-BAF of Example 7.

Proposition 3.
For any BAF $\Delta = \langle A, \Omega, \Gamma \rangle$ *and semantics* $\sigma \in \{\texttt{co}, \texttt{gr}, \texttt{pr}, \texttt{st}\}$, *checking whether an argument* $g \in A$ *is*

- *credulously accepted under* σ *is:* (i) *in PTIME for* $\sigma = \texttt{gr}$; *and* (ii) *NP-complete for* $\sigma \in \{\texttt{co}, \texttt{st}, \texttt{pr}\}$;
- *skeptically accepted under* σ *is:* (i) *in PTIME for* $\sigma = \texttt{gr}$; (ii) *coNP-complete for* $\sigma \in \{co, st\}$; *and* (iii) Π_2^P*-complete for* $\sigma = \texttt{pr}$.

Proof.
Same strategy used in the proof of Proposition 2 can be used, where we check that (P_Δ, g, σ^*) is a true instance of the credulous/skeptical acceptance problem in LP (Saccà 1997). □

4 A new semantics for recursive BAF

In this section, we present new semantics for Recursive BAF (Rec-BAF) frameworks. Analogously to the case of BAF, we assume that self-supported arguments (*w.r.t.* a set $\mathbf{S}$), that is arguments a s.t. there exists a cycle $a \overset{\beta_1}{\Rightarrow} a_1 \overset{\beta_2}{\Rightarrow} \cdots \overset{\beta_n}{\Rightarrow} a$ $.\{\beta_1, \ldots, \beta_n\} \subseteq \mathbf{S}$, are always defeated. Thus, the definition of defeated elements can be accomplished by adding such a condition.

Definition 9.
For any general RAFN $\langle A, R, T, \mathbf{s}, \mathbf{t} \rangle$ and set $\mathbf{S} \subseteq A \cup R \cup T$, we have that:

- $DEF(\mathbf{S}) = \{X \in A \cup R \cup T \mid (\exists \alpha \in R \cap \mathbf{S}\, .\, \mathbf{t}(\alpha) = X \wedge \mathbf{s}(\alpha) \in \mathbf{S}) \vee$
  $(\exists \beta \in T \cap \mathbf{S}\, .\, \mathbf{t}(\beta) = X \wedge \mathbf{s}(\beta) \in DEF(\mathbf{S})) \vee$
  $(\exists \text{ cycle } X \overset{\beta_1}{\Rightarrow} a_1 \overset{\beta_2}{\Rightarrow} \cdots \overset{\beta_n}{\Rightarrow} X\ .\{\beta_1, \ldots, \beta_n\} \subseteq \mathbf{S})\}$;
- $ACC(\mathbf{S}) = \{X \in A \cup R \cup T | (\forall \alpha \in R\, .\, \mathbf{t}(\alpha) = X$ implies
  $(\alpha \in DEF(\mathbf{S}) \vee \mathbf{s}(\alpha) \in DEF(\mathbf{S}))) \wedge$
  $(\forall \beta \in T\, .\, \mathbf{t}(\beta) = X$ implies $(\beta \in DEF(\mathbf{S}) \vee \mathbf{s}(\beta) \in ACC(\mathbf{S})))\}$.

Again, we have highlighted in blue the differences between the definition of defeated and accepted arguments for general and acyclic RAFN. As for the case of BAF we assume as defeated those arguments that "depends" on a cycle of support $\beta_1, \ldots, \beta_n$ and all supports $\beta_1, \ldots, \beta_n$ are part of the candidate extension $\mathbf{S}$.

Example 7.
Consider the RAFN Δ (shown in Figure 4) obtained from that of Figure 1 by adding the support β_2 with $\mathbf{s}(\beta_2) = \texttt{white}$ and $\mathbf{t}(\beta_2) = \texttt{fish}$, and assume arguments are denoted by their initials. The preferred (and also stable) extensions under RAFN semantics are $E_1 = \{\texttt{m}, \texttt{r}, \bar{\texttt{s}}, \alpha_1, \ldots, \alpha_8,\ \beta_1,\ \beta_2\}$ and $E_2 = \{\texttt{m}, \texttt{r}, \texttt{s}, \alpha_3, \ldots, \alpha_8,\ \beta_1,\ \beta_2\}$. Observe that, for $E_3 = \{\texttt{f}, \texttt{w}, \bar{\texttt{s}}, \alpha_1, \ldots, \alpha_8,\ \beta_1,\ \beta_2\}$ (resp., $E_4 = \{\texttt{f}, \texttt{m}, \texttt{w}, \texttt{s}, \alpha_3, \ldots, \alpha_8, \beta_1, \beta_2\}$), it holds

https://doi.org/10.1017/S1471068424000310 Published online by Cambridge University Press

that $\mathtt{f,w} \in DEF(E_3)$ (resp., $\mathtt{f,w} \in DEF(E_4)$), and thus, E_3 and E_4 are not stable (and preferred) extensions.

Definition 10.
For any general ASAF $\langle A, R, T, \mathbf{s}, \mathbf{t}\rangle$ and set $\mathbf{S} \subseteq A \cup R \cup T$, we have that:

- $DEF(\mathbf{S}) = \{X \in A \cup R \cup T \mid (X \in R \wedge \mathbf{s}(X) \in DEF(\mathbf{S})) \vee$
  $(\exists \alpha \in R \cap \mathbf{S} \,.\, \mathbf{t}(\alpha) = X) \vee$
  $(\exists \beta \in T \cap \mathbf{S} \,.\, \mathbf{t}(\beta) = X \ \wedge \ \mathbf{s}(\beta) \in DEF(\mathbf{S})) \vee$
  $(\exists \text{ cycle } X \overset{\beta_1}{\Rightarrow} a_1 \overset{\beta_2}{\Rightarrow} \cdots \overset{\beta_n}{\Rightarrow} X \ .\{\beta_1, \ldots, \beta_n\} \subseteq \mathbf{S})\}$;
- $ACC(\mathbf{S}) = \{X \in A \cup R \cup T \mid (X \in R \text{ implies } \mathbf{s}(X) \in ACC(\mathbf{S})) \wedge$
  $(\forall \alpha \in R.\, \mathbf{t}(\alpha) = X \text{ implies } \alpha \in DEF(\mathbf{S})) \wedge$
  $(\forall \beta \in T \,.\, \mathbf{t}(\beta) = X \text{ implies } (\beta \in DEF(\mathbf{S}) \vee \mathbf{s}(\beta) \in ACC(\mathbf{S})))\}$.

Again, we have highlighted in blue the differences between the definition of defeated and accepted arguments for general and acyclic ASAF. As for the case of BAF, we assume as defeated those arguments that "depends" on a cycle of support $\beta_1, \ldots, \beta_n$ and all supports $\beta_1, \ldots, \beta_n$ are part of the candidate extension $\mathbf{S}$.

Example 8.
Consider the ASAF Δ shown in Figure 4, and assume arguments are denoted by their initials. The preferred (and also stable) extensions under ASAF semantics are: $E_1 = \{\mathtt{m,r,} \ \bar{\mathtt{s}}, \alpha_2, \alpha_3, \alpha_6, \beta_1, \beta_2\}$ and $E_2 = \{\mathtt{m,r,s}, \alpha_3, \alpha_5, \alpha_7, \alpha_8, \beta_1, \beta_2\}$. Observe that, for $E_3 = \{\mathtt{f,w}, \bar{\mathtt{s}}, \alpha_1, \alpha_4, \alpha_6, \beta_1, \beta_2\}$ (resp., $E_4 = \{\mathtt{f,m,w,s}, \alpha_4, \alpha_5, \alpha_7, \alpha_8, \beta_1, \beta_2\}$) it holds that $\mathtt{f,w} \in DEF(E_3)$ (resp., $\mathtt{f,w} \in DEF(E_4)$) and thus E_3 and E_4 are not stable (and preferred) extensions.

As for BAF our Rec-BAF semantics coincide with that of acyclic Rec-BAF defined in Section 2 whenever the Rec-BAF is support-acyclic, as stated in the following proposition.

Proposition 4.
Let Δ be a RAFN (resp., ASAF) and $\sigma \in \{\mathtt{co}, \mathtt{gr}, \mathtt{pr}, \mathtt{st}\}$ a semantics. If Δ is acyclic, then $\sigma(\Delta)$ computed by means of Definition 3 (resp., Definition 4) coincides with $\sigma(\Delta)$ computed by means of Definition 9 (resp., Definition 10).

Proof.
The result follows by observing that, for any acyclic Rec-BAF $\Delta = \langle A, R, T, \mathbf{s}, \mathbf{t}\rangle$ and set $\mathbf{S} \subseteq A \cup R \cup T$, we have that $DEF(\mathbf{S}) = def(\mathbf{S})$ and $ACC(\mathbf{S}) = acc(\mathbf{S})$. □

The semantics of general Rec-BAFs under deductive support (i.e., RAFD, and AFRAD) are obtained from the definition of defeated and acceptable elements in acyclic Rec-BAF (i.e., *def*, *acc*) by (i) replacing *def* and *acc* with DEF and ACC and (ii) adding the condition $\exists$ cycle $X \overset{\beta_1}{\Rightarrow} a_1 \overset{\beta_2}{\Rightarrow} \cdots \overset{\beta_n}{\Rightarrow} X \ .\{\beta_1, \ldots, \beta_n\} \subseteq \mathbf{S}$ in the disjunct condition of DEF, as done in Definition 9 for RAFN and Definition 10 for ASAF.

4.1 LP mapping

We now provide the mappings from general Rec-BAF to ground logic programs, so that the set of σ-extensions of any general Rec-BAF Δ is equivalent to that of partial stable models of the corresponding logic program P_Δ. Basically, the logic rules of the program

https://doi.org/10.1017/S1471068424000310 Published online by Cambridge University Press

P_Δ are derived from the topology of Δ. The main difference w.r.t. the rules generated for BAF is that we now have rules defining all elements (arguments, attacks, and supports) denoted by X. Thus, for instance, regarding RAFN, a target element X is true if (i) for every attack α, either α or the source of α is false, and (ii) for every support β, either β is false, or the source of β is true.

Definition 11.
Given an RAFN (resp., RAFD) $\Delta = \langle A, R, T, \mathbf{s}, \mathbf{t}\rangle$, then P_Δ (the propositional program derived from Δ) contains, for each element $X \in A \cup R \cup T$, a rule

$$X \leftarrow \bigwedge_{\alpha \in R \wedge \mathbf{t}(\alpha)=X} (\neg\alpha \vee \neg\mathbf{s}(\alpha)) \wedge \bigwedge_{\beta \in T \wedge \mathbf{t}(\beta)=X} (\neg\beta \vee \mathbf{s}(\beta)).$$

$$\left(\text{resp. } X \leftarrow \bigwedge_{\alpha \in R \wedge \mathbf{t}(\alpha)=X} (\neg\alpha \vee \neg\mathbf{s}(\alpha)) \wedge \bigwedge_{\beta \in T \wedge \mathbf{s}(\beta)=X} (\neg\beta \vee \mathbf{t}(\beta)).\right)$$

Definition 12.
For any ASAF (resp., AFRAD) $\Delta = \langle A, R, T, \mathbf{s}, \mathbf{t}\rangle$, P_Δ (the propositional program derived from Δ) contains, for each $X \in A \cup R \cup T$, a rule of the form

$$X \leftarrow \varphi(X) \wedge \bigwedge_{\alpha \in R \wedge \mathbf{t}(\alpha)=X} \neg\alpha \wedge \bigwedge_{\beta \in T \wedge \mathbf{t}(\beta)=X} (\neg\beta \vee \mathbf{s}(\beta))$$

$$\left(\text{resp. } X \leftarrow \varphi(X) \wedge \bigwedge_{\alpha \in R \wedge \mathbf{t}(\alpha)=X} \neg\alpha \wedge \bigwedge_{\beta \in T \wedge \mathbf{s}(\beta)=X} (\neg\beta \vee \mathbf{t}(\beta))\right)$$

where $\varphi(X) = \mathbf{s}(X)$ if $X \in R$; otherwise, $\varphi(X) = \mathtt{true}$.

The set of complete extensions of any Rec-BAF Δ coincides with the set of PSMs of the derived propositional program P_Δ.

Theorem 2.
For any Rec-BAF Δ, $\widehat{\mathtt{co}}(\Delta) = \mathcal{PS}(P_\Delta)$.

Proof.
The proof is similar to that of Theorem 1. The only difference is that complete extensions also contain attacks and supports, whereas the logic program also contains rules defining attacks and supports and, consequently, the partial stable models contain arguments, attacks, and supports. □

Example 9.
Consider the RAFN Δ shown in Figure 4, and assume arguments are denoted by their initials. The propositional program P_Δ derived from Δ is as follows:

$\mathtt{m} \leftarrow \neg\alpha_1 \vee \neg\mathtt{f}$;	$\mathtt{f} \leftarrow (\neg\alpha_2 \vee \neg\mathtt{m}) \wedge (\neg\beta_2 \vee \mathtt{w})$;	$\mathtt{s} \leftarrow \neg\alpha_6 \vee \neg\bar{\mathtt{s}}$;	$\bar{\mathtt{s}} \leftarrow \neg\alpha_5 \vee \neg\mathtt{s}$;
$\mathtt{r} \leftarrow \neg\alpha_4 \vee \neg\mathtt{w}$;	$\mathtt{w} \leftarrow (\neg\alpha_3 \vee \neg\mathtt{r}) \wedge (\neg\beta_1 \vee \mathtt{f})$;	$\alpha_1 \leftarrow \neg\alpha_7 \vee \neg\mathtt{s}$;	$\alpha_2 \leftarrow \neg\alpha_8 \vee \neg\mathtt{s}$;
$\alpha_3 \leftarrow$; $\alpha_4 \leftarrow$;	$\alpha_5 \leftarrow$; $\alpha_6 \leftarrow$;	$\alpha_7 \leftarrow$; $\alpha_8 \leftarrow$;	$\beta_1 \leftarrow$; $\beta_2 \leftarrow$;

where $\mathcal{PS}(P_\Delta) = \widehat{\mathtt{co}}(\Delta) = \{E_0, E_1, E_2, E_3, E_4, E_5\}$, with: $E_0 = \{\alpha_3, \ldots, \alpha_8, \beta_1, \beta_2\}$, $E_1 = E_0 \cup \{\mathtt{m},\mathtt{r}\}$, $E_2 = E_0 \cup \{\mathtt{m},\mathtt{s}\}$, $E_3 = E_0 \cup \{\mathtt{m},\mathtt{r},\mathtt{s}\}$, $E_4 = E_0 \cup \{\bar{\mathtt{s}}, \alpha_1, \alpha_2\}$, and $E_5 = E_0 \cup \{\mathtt{m},\mathtt{r}, \bar{\mathtt{s}}, \alpha_1, \alpha_2\}$.

https://doi.org/10.1017/S1471068424000310 Published online by Cambridge University Press

Observe that, from the above result and the fact that any LP can be converted into an "equivalent" AF (Caminada *et al.* 2015), it is also possible to convert any Rec-BAF into an "equivalent" AF in polynomial time.

4.2 Computational complexity

The complexity of verification and acceptance problems for acyclic Rec-BAF are well-known and coincide with the corresponding ones on AF, as any acyclic Rec-BAF can be rewritten into an AF. In this section we show that the same holds also for general Rec-BAF, that is the complexities of the verification and acceptance problems for general BAF are the same of those known for AF, as stated in the following two propositions.

Proposition 5.
For any Rec-BAF $\langle A, R, T, \mathtt{s}, \mathtt{t}\rangle$ *and semantics* $\sigma \in \{\mathtt{co}, \mathtt{gr}, \mathtt{pr}, \mathtt{st}\}$, *checking whether a set of arguments* $\mathbf{S} \subseteq A$ *is a* σ*-extension for* Δ *is* (i) *in PTIME for* $\sigma \in \{\mathtt{gr}, \mathtt{co}, \mathtt{st}\}$; *and* (ii) *coNP-complete for* $\sigma = \mathtt{pr}$.

Proof.
Lower bounds derive from the complexity results of the same problems for AF (Dvorák and Dunne 2017), as Rec-BAF is a generalization of AF. Regarding upper bounds, the proof can carried out by writing P_Δ (in PTIME) and check that $(P_\Delta, \widehat{\mathbf{S}} = \mathbf{S} \cup \{\neg x \mid x \in DEF(\mathbf{S})\}, \sigma^*)$ is a true instance of the verification problem in LP (Saccà 1997), where $\sigma^* = \mathtt{WF}$ (resp., $\mathcal{PS}, \mathcal{TS}, \mathcal{MS}$) iff $\sigma = \mathtt{gr}$ (resp., $\mathtt{co}$, $\mathtt{st}$, $\mathtt{pr}$). As for Theorem 2 we have that $\widehat{\mathtt{co}}(\Delta) = \mathcal{PS}(P_\Delta)$, the result follows. □

Proposition 6.
For any Rec-BAF $\langle A, R, T, \mathtt{s}, \mathtt{t}\rangle$ *and semantics* $\sigma \in \{\mathtt{co}, \mathtt{gr}, \mathtt{pr}, \mathtt{st}\}$, *checking whether an argument* $g \in A$ *is*

- *credulously accepted under* σ *is:* (i) *in PTIME for* $\sigma = \mathtt{gr}$; *and (ii) NP-complete for* $\sigma \in \{\mathtt{co}, \mathtt{st}, \mathtt{pr}\}$;
- *skeptically accepted under* σ *is:* (i) *in PTIME for* $\sigma = \mathtt{gr}$; (ii) *coNP-complete for* $\sigma \in \{\mathtt{co}, \mathtt{st}\}$; *and* (iii) Π_2^P*-complete for* $\sigma = \mathtt{pr}$.

Proof.
Same strategy used in the proof of Proposition 5 can be used, where we check that (P_Δ, g, σ^*) is a true instance of the credulous/skeptical acceptance problem in LP (Saccà 1997). □

It turns out that general Rec-BAF are as expressive as BAF and AF, though more general relations (i.e., cyclic supports and recursive relations) can be easily expressed.

5 Related work

Bipolarity in argumentation is discussed by Amgoud *et al.* (2004), where a formal definition of bipolar argumentation framework (BAF) extending Dung's AF by including supports is provided. A survey on different approaches to support in argumentation, as well as on different recursive AF-based frameworks is provided by Cohen *et al.* (2014) and Cayrol *et al.* (2021), respectively. However, a semantics for AFNs with cyclic

https://doi.org/10.1017/S1471068424000310 Published online by Cambridge University Press

supports have been recently defined in the literature by Nouioua and Boutouhami (2023). Essentially, it is based on avoiding considering the contribution of *incoherent* (set of) arguments, that are those occurring in support -cycles or (transitively) supported by arguments in support cycles. In the same spirit, our approach ensures that incoherent arguments are always defeated (i.e., appearing in DEF). A semantics for general RAFNs has been recently defined by Lagasquie-Schiex (2023). We believe their proposal is quite intricate as it relies on numerous definitions, whereas ours seamlessly extends the definitions of defeated and acceptable elements, previously defined for AF(N) in an elegant and uniform way.

There has been an increasing interest in studying the relationships between argumentation frameworks and logic programming (LP) (Caminada *et al*. 2015; Alfano *et al*. 2020b). In particular, the semantic equivalence between complete extensions in AF and 3-valued stable models in LP was first established by Wu *et al*. (2009). Then, the relationships of LP with AF have been further studied by Caminada *et al*. (2015). A one-to-one correspondence between extensions of general AFN and corresponding normal logic program (LP) has been proposed by Nouioua and Boutouhami (2023), under complete-based semantics. A logical encoding able to characterize the semantics of general RAFN has been proposed by Lagasquie-Schiex (2023). However, the rules defining the acceptability of arguments are a bit involved, as they require several predicate symbols. Nevertheless, Nouioua and Boutouhami (2023); Lagasquie-Schiex (2023) devoted no attention to deductive supports.

Efficient mappings from AF to *Answer Set Programming (ASP)* (i.e., LP with *Stable Model* semantics (Gelfond and Lifschitz, 1988)) have been investigated as well by Sakama and Rienstra (2017); Gaggl *et al*. (2015). The well-known AF system ASPARTIX (Dvorák *et al*. 2020) is implemented by rewriting the input AF into an ASP program and using an ASP solver to compute extensions.

Our work is complementary to approaches providing the semantics for an AF-based framework by using meta-argumentation, that is, by relying on a translation from a given AF-based framework to an AF (Cohen *et al*. 2015; Alfano *et al*. 2018a,b). In this regard, we observe that meta-argumentation approaches have the drawback of making it a bit difficult to understand the original meaning of arguments and interactions, once translated into the resulting meta-AF. In fact, those approaches rely on translations that generally require adding several meta-arguments and meta-attacks to the resulting meta-AF in order to model the original interactions. Concerning approaches that provide the semantics of argumentation frameworks by LPs (Caminada *et al*. 2015), we observe that a logic program for an AF-based framework can be obtained by first flattening the given framework into a meta-AF and then converting it into a logic program. The so-obtained program contains the translation of meta-arguments and meta-attacks that make the program much more verbose and difficult to understand (because not straightly derived from the given extended AF framework) in our opinion, compared with the direct translation we proposed. Moreover, the proposed approach uniformly deals with several AF-based frameworks. Other extensions of the Dung's framework not explicitly discussed in this paper are also captured by our technique as they are special cases of some of those studied in this paper. This is the case of *Extended AF (EAF)* and *hierarchical EAF*, which extend AF by allowing second order and stratified attacks, respectively (Modgil 2009), that are special cases of recursive attacks.

https://doi.org/10.1017/S1471068424000310 Published online by Cambridge University Press

Three-valued semantics have been also explored in the *Abstract Dialectical Framework* (ADF (Brewka *et al*. 2013; Strass and Wallner 2015; Brewka *et al*. 2020; Baumann and Heinrich 2023), that allows to explicit acceptance conditions over arguments in the form of propositional logic formulae. Recently, it has been shown that its semantics can be captured by the (monotonic three-valued) possibilistic logic (Heyninck *et al*. 2022). In particular, the semantics of an ADF relies on a characteristic operator which takes as an input a three-valued interpretation ν and returns an interpretation by considering all possible two-valued completions of ν. We point out that (general) Rec-BAF can be modeled by ADF, and in particular that verification and acceptance reasoning in Rec-BAF can be reduced to ADF. This is backed by the computational complexity of the two frameworks (Strass and Wallner 2015) as ADF is strictly more expressive than Rec-BAF under complete, preferred and stable semantics (one level higher in the polynomial hierarchy). Moreover, less expressive subclasses of ADF have been also explored. For instance, the subclass called bipolar ADFs has been shown to exhibit complexity comparable to that of AF (and to that of Rec-BAF), as is it possible to avoid considering all the possible two-valued completions through the application of Kleene logic (Baumann and Heinrich 2023). Exploring the connection between subclasses of ADF and Rec-BAF is a possible direction for future work.

6 Conclusions

In this paper, we jointly tackled two relevant aspects that were so far considered separately by the community of argumentation: extending Dung's abstract argumentation framework with recursive attacks and (general) supports, and show that the semantics for specific BAF and Rec-BAF frameworks can be defined by very simple and intuitive modifications of that defined for the case of AF. We presented in an elegant and uniform way the semantics of several (possibly cyclic) AF-based frameworks, including those for which a semantics has never been proposed.

Our semantics is inspired by the self-supportless principle in logic programs, which ensures no literal in an answer set is supported exclusively by itself or through a cycle of dependencies that lead back to itself. This principle is essential for maintaining the integrity of the answer set, ensuring that it is grounded in the logic rules provided and does not rely on circular reasoning. It is worth noting that this intuition is also reflected in the stable semantics of ADF (Brewka *et al*. 2013). Indeed, the basic intuition is that all elements of a stable model should have a non-cyclic justification. For instance, considering the (bipolar) ADF having two statements a and b where the acceptance condition of a is b and vice versa, the set $\{a, b\}$ is not a stable model. Thus, similarly to answer-set and ADF stable semantics, a principle of our semantics is that arguments cannot be circularly justified.

Finally, we believe that our approach is also complementary to that using intermediate translations to AF to define the Rec-BAF semantics. Indeed, our approach can also be used to provide additional tools for computing complete extensions using answer set solvers (Gebser *et al*. 2018) and program rewriting techniques (Janhunen *et al*. 2006; Sakama and Rienstra 2017).

https://doi.org/10.1017/S1471068424000310 Published online by Cambridge University Press

Supplementary material

To view supplementary material for this article, please visit http://doi.org/10.1017/S1471068424000310.

References

Alfano, G., Calautti, M., Greco, S., Parisi, F. and Trubitsyna, I. 2023a. Explainable acceptance in probabilistic and incomplete abstract argumentation frameworks. *Artificial Intelligence* 323a, 103967.

Alfano, G., Cohen, A., Gottifredi, S., Greco, S., Parisi, F. and Simari, G. R. 2020a. Dynamics in abstract argumentation frameworks with recursive attack and support relations. In Proc. of the 24th European Conference on Artificial Intelligence (ECAI), IOS Press, 577–584.

Alfano, G., Cohen, A., Gottifredi, S., Greco, S., Parisi, F. and Simari, G. R. 2024a. Credulous acceptance in high-order argumentation frameworks with necessities: An incremental approach. *Artificial Intelligence* 333a, 104159.

Alfano, G., Greco, S., Mandaglio, D., Parisi, F. and Trubitsyna, I. 2024b. Abstract argumentation frameworks with strong and weak constraints. *Artificial Intelligence* 336b, 104205.

Alfano, G., Greco, S. and Parisi, F. 2018a. Computing extensions of dynamic abstract argumentation frameworks with second-order attacks. In *Proc. of the 22nd International Database Engineering and Application Symposium (IDEAS)*, ACM, 183–192.

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

Alfano, G., Greco, S., Parisi, F. and Trubitsyna, I. 2024c.Complexity of credulous and skeptical acceptance in epistemic argumentation framework. In *Proc. of the 38th AAAI Conference on Artificial Intelligence*, AAAI Press, 10423–10432.

Alfano, G., Greco, S., Parisi, F. and Trubitsyna, I. 2020b. On the semantics of abstract argumentation frameworks: A logic programming approach. *Theory and Practice of Logic Programming* 5, 703–718.

Alfano, G., Greco, S., Parisi, F. and Trubitsyna, I. 2022. On preferences and priority rules in abstract argumentation. In *Proc. of the 31st International Joint Conference on Artificial Intelligence*, ijcai.org pub., 2517–2524.

Alfano, G., Greco, S., Parisi, F. and Trubitsyna, I. 2023c. On acceptance conditions in abstract argumentation frameworks. *Information Sciences* 625c, 757–779.

Alfano, G., Greco, S., Parisi, F.AND Trubitsyna, I. 2023d. Preferences and constraints in abstract argumentation. In *Proc. of the 32nd International Joint Conference on Artificial Intelligence*, ijcai.org, 3095–3103.

Alfano, G., Greco, S., Parisi, F. and Trubitsyna, I. 20203b. Epistemic abstract argumentation framework: Formal foundations, computation and complexity. In *Proc. of the 2023 International Conference on Autonomous Agents and Multiagent Systems (AAMAS), ACM,* 409–417.

Amgoud, L., Cayrol, C. and Lagasquie-Schiex, M.-C. 2004. On the bipolarity in argumentation frameworks. In *Proc. of 10th International Workshop on Non-Monotonic Reasoning (NMR)*, 1–9.

Baroni, P., Cerutti, F., Giacomin, M. and Guida, G. 2011. AFRA: Argumentation framework with recursive attacks. *International Journal of Approximate Reasoning* 52, 1, 19–37.

https://doi.org/10.1017/S1471068424000310 Published online by Cambridge University Press

Baumann, R. and Heinrich, M. 2023. Bipolar abstract dialectical frameworks are covered by Kleene's three-valued logic. In *Proc. of the 32nd International Joint Conference on Artificial Intelligence* (IJCAI), ijcai.org, 3123–3131.

Brewka, G., Diller, M., Heissenberger, G., Linsbichler, T. and Woltran, S. 2020. Solving advanced argumentation problems with answer set programming. *Theory and Practice of Logic Programming* 20, 3, 391–431.

Brewka, G., Strass, H., Ellmauthaler, S., Wallner, J. P. and Woltran, S. 2013. Abstract dialectical frameworks revisited. In *Proc. of the 23rd International Joint Conference on Artificial Intelligence (IJCAI)*, AAAI Press, 803–809.

Caminada, M., Sa, S., Alcantara, J. F. L. and Dvorak, W. 2015. On the equivalence between logic programming semantics and argumentation semantics. *International Journal of Approximate Reasoning* 58, 87–111.

Cayrol, C., Cohen, A. and Lagasquie-Schiex, M.-C. 2021. Higher-order interactions (bipolar or not) in abstract argumentation: A state of the art. *FLAP* 8, 6, 1339–1436.

Cayrol, C., Fandinno, J., Del Cerro, L. F. and Lagasquie-Schiex, M. 2017. Valid attacks in argumentation frameworks with recursive attacks. In *Proc. of COMMONSENSE conference*, CEUR-WS.org.

Cayrol, C., Fandinno, J., Del Cerro, L. F. and Lagasquie-Schiex, M. 2018. Structure-based semantics of argumentation frameworks with higher-order attacks and supports. In *Proc. of COMMA*, 29–36.

Cohen, A., Gottifredi, S., Garcia, A. J. and Simari, G. R. 2014. A survey of different approaches to support in argumentation systems. *The Knowledge Engineering Review* 29, 5, 513–550.

Cohen, A., Gottifredi, S., Garcia, A. J. and Simari, G. R. 2015. An approach to abstract argumentation with recursive attack and support. *Journal of Applied Logic* 13, 4, 509–533.

Dung, P. M. 1995. On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games. *Artificial Intelligence* 77, 2, 321–358.

Dvorak, W. and Dunne, P. E. 2017. Computational problems in formal argumentation and their complexity. *FLAP* 4, 8.

Dvorak, W., Gaggl, S. A., Rapberger, A., Wallner, J. P. and Woltran, S. 2020. The ASPARTIX system suite. In *Proc. of COMMA conference*, IOS Press, 461–462.

Dvorak, W., GressLer, A., Rapberger, A. and Woltran, S. 2023. The complexity landscape of claim-augmented argumentation frameworks. *Artificial Intelligence* 317, 103873.

Dvorak, W., Hecher, M., Konig, M., Schidler, A., Szeider, S. and Woltran, S. 2022. Tractable abstract argumentation via backdoor-treewidth. In *Proc. of the 36th AAAI Conference on Artificial Intelligence, AAAI Press*, 5608–5615.

Dvorak, W., Konig, M., Ulbricht, M. and Woltran, S. 2024. Principles and their computational consequences for argumentation frameworks with collective attacks. *Journal of Artificial Intelligence Research* 79, 69–136.

Fazzinga, B., Flesca, S. and Furfaro, F. 2022. Abstract argumentation frameworks with marginal probabilities. In *Proc. of the 31st International Joint Conference on Artificial Intelligence*, ijcai.org, 2613–2619.

Gaggl, S. A., Manthey, N., Ronca, A., Wallner, J. P. and Woltran, S. 2015. Improved answer-set programming encodings for abstract argumentation. *Theory and Practice of Logic Programming* 15, 4-5, 434–448.

Gebser, M., Leone, N., Maratea, M., Perri, S., Ricca, F. and Schaub, T. 2018. Evaluation techniques and systems for answer set programming: A survey. In *Proc. of the 27th International Joint Conference on Artificial Intelligence*, ijcai.org, 5450–5456.

https://doi.org/10.1017/S1471068424000310 Published online by Cambridge University Press

GELFOND, M. and LIFSCHITZ, V. 1988. The stable model semantics for logic programming. In *Proc. of the 1st International Logic Programming Conference and Symposium*, MIT Press, 1070–1080.

GOTTIFREDI, S., COHEN, A., GARCIA, A. J. and SIMARI, G. R. 2018. Characterizing acceptability semantics of argumentation frameworks with recursive attack and support relations. *Artificial Intelligence* 262, 336–368.

HEYNICK, J., KERN-ISBERNER, G., RIENSTRA, T., SKIBA, K. and THIMM, M. 2022. Possibilistic logic underlies abstract dialectical frameworks. In *Proc. of the 31st International Joint Conference on Artificial Intelligence*, ijcai.org, 2655–2661.

JANHUNEN, T., NIEMELA, I., SEIPEL, D., SIMONS, P. and YOU, J.-H. 2006. Unfolding partiality and disjunctions in stable model semantics. *ACM Transactions on Computational Logic* 7, 1, 1–37.

LAGASQUIE-SCHIEX, M. 2023. Handling support cycles and collective interactions in the logical encoding of higher-order bipolar argumentation frameworks. *Journal of Logic and Computation* 33, 2, 289–318.

MODGIL, S. 2009. Reasoning about preferences in argumentation frameworks. *Artificial Intelligence* 173, 901–934.

NOUIOUA, F. and BOUTOUHAMI, S. 2023. Argumentation frameworks with necessities and their relationship with logic programs. *Argument & Computation* 14, 1, 17–58.

NOUIOUA, F. and RISCH, V. 2011. Argumentation frameworks with necessities. In SUM, 163–176.

RAHWAN, I. and SIMARI, G. R. 2009. *Argumentation in Artificial Intelligence*. Springer.

SACCA, D. 1997. The expressive powers of stable models for bound and unbound DATALOG queries. *Journal of Computer and System Sciences* 54, 3, 441–464.

SAKAMA, C. and RIENSTRA, T. 2017. Representing argumentation frameworks in answer set programming. *Fundamenta Informaticae* 155, 3, 261–292.

STRASS, H. and WALLNER, J. P. 2015. Analyzing the computational complexity of abstract dialectical frameworks via approximation fixpoint theory. *Artificial Intelligence* 226, 34–74.

VAN GELDER, A., ROSS, K. A. and SCHLIPF, J. S. 1991. The well-founded semantics for general logic programs. *Journal of the ACM* 38, 3, 620–650.

VILLATA, S., BOELLA, G., GABBAY, D. M. and VAN DER TORRE, L. W. N. 2012. Modelling defeasible and prioritized support in bipolar argumentation. *Annals of Mathematics and Artificial Intelligence* 66, 1-4, 163–197.

WU, Y., CAMINADA, M. and GABBAY, D. M. 2009. Complete extensions in argumentation coincide with 3-valued stable models in logic programming. *Studia Logica* 93, 2-3, 383–403.

https://doi.org/10.1017/S1471068424000310 Published online by Cambridge University Press