

Defining the Semantics of Abstract Argumentation Frameworks through Logic Programs and Partial Stable Models (Extended Abstract)*

Gianvincenzo Alfano, Sergio Greco, Francesco Parisi and Irina Trubitsyna

DIMES Department, University of Calabria, Rende, Italy

{g.alfano, greco, fparisi, i.trubitsyna}@dimes.unical.it

Abstract

Extensions of Dung’s Argumentation Framework (AF) include the class of Recursive Bipolar AFs (Rec-BAFs), i.e. AFs with recursive attacks and supports. We show that a Rec-BAF Δ can be translated into a logic program P_Δ so that the extensions of Δ under different semantics coincide with subsets of the partial stable models of P_Δ .

1 Introduction

Dung’s abstract Argumentation Framework (AF) [Dung, 1995] is a simple yet powerful formalism for modelling disputes between two or more agents which 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 which allows for modelling the *support* between arguments [Nouioua and Risch, 2011; Villata *et al.*, 2012]. Different interpretations of the notion of support have been proposed [Cayrol and Lagasque-Schiex, 2013; Cohen *et al.*, 2014]. *Deductive support* [Villata *et al.*, 2012] is intended to capture the following intuition: if argument a supports argument b , then the acceptance of a implies the acceptance of b ; thus, the non-acceptance of b implies the non-acceptance of a . On the other hand, *necessary support* [Nouioua and Risch, 2011; Baroni *et al.*, 2011] is interpreted in a dual way [Cayrol and Lagasque-Schiex, 2013]: if a supports b , then the acceptance of a is necessary to get the acceptance of b ; equivalently, accepting b implies accepting a . An *AFN (AF with Necessities)* is a BAF where supports are interpreted as necessities [Nouioua and Risch, 2011]; an *AFD (AF with Deductive supports)* is a BAF where supports are interpreted as deductions [Villata *et al.*, 2012]. Clearly, the way the support is interpreted changes the set of *extensions*, i.e. the set of acceptable elements of an argumentation framework.

Further extensions of AF consider recursive AFs and recursive BAFs where attacks/supports can be recursively at-

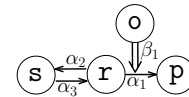


Figure 1: Rec-BAF of Example 1.

tacked/supported [Baroni *et al.*, 2011; Cayrol *et al.*, 2017; Gottifredi *et al.*, 2018; Cayrol *et al.*, 2018].

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 (AFRA)* [Baroni *et al.*, 2011] and of an *Attack-Support Argumentation Framework (ASAF)* [Cohen *et al.*, 2015; Gottifredi *et al.*, 2018] also include the (names of) attacks and supports that intuitively contribute to determine the set of accepted arguments. Particularly, the acceptability of an attack is related to the acceptability of its source argument: an attack in AFRA is defeated even when its source argument is defeated. This is not the case for *Recursive AF (RAF)* [Cayrol *et al.*, 2017] and *Recursive AF with Necessities (RAFN)* frameworks [Cayrol *et al.*, 2018], which offer a different semantics for recursive AFs and recursive BAFs with necessary supports, respectively. From a syntax standpoint all the argumentation frameworks mentioned above can be viewed as (possibly restricted forms of) *Recursive Bipolar Argumentation Frameworks (Rec-BAFs)*, though semantically different because having different interpretations of the supports and different ways of determining the status of attacks.

Example 1 Consider a situation where we want to decide whether to play tennis on the basis of some information. Assume we have the following arguments: r (it is raining), p (play tennis), o (the tennis court is outside), s (it is sunny), and the implications: (α_1) if it is raining, then we do not play tennis, and (β_1) if the tennis court is outside, then α_1 should hold. Notice that arguments r and s are mutually conflicting. The scenario can be modelled using the Rec-BAF Δ of Figure 1 where α_1 , as well as α_2 and α_3 modelling the mutual conflicts between r and s , are *attacks* (denoted by \rightarrow) and β_1 is a *support* (denoted by \Rightarrow).

Recently there has been an increasing interest in studying the relationships between argumentation frameworks and logic programming (LP). In particular, the semantic equivalence between (complete) extensions in AF and 3-valued stable models in LP was first established in [Wu *et al.*, 2009]

*This paper is based on the work “On the semantics of abstract argumentation frameworks: A logic programming approach”, that was presented at the 36th International Conference on Logic Programming [Alfano *et al.*, 2020c], awarded as best technical paper.

and further studied in [Caminada *et al.*, 2015]. Efficient mappings from AF to *Answer Set Programming* (i.e. LP with *Stable Model* semantics [Gelfond and Lifschitz, 1988]) have been investigated as well [Sakama and Rienstra, 2017; Gaggl *et al.*, 2015]. However, the relationships between LP and more general frameworks extending AF such as Rec-BAFs has not been adequately studied. Thus, in this paper, by generalizing the work in [Caminada *et al.*, 2015], we introduce a general approach for characterizing the extensions of different AF-based frameworks (e.g. AFRA, RAF, ASAF, RAFN) under several well-known argumentation semantics in terms of Partial Stable Models (PSMs) of logic programs. This is achieved by providing a modular definition of the sets of *defeated* and *acceptable* elements (i.e. arguments, attacks and supports) for each AF-based framework, and by leveraging on the connection between argumentation semantics and subsets of PSMs. In particular, for any framework belonging to the class of Rec-BAFs, our formulation of acceptable elements allows us to easily and systematically derive a propositional program whose PSMs corresponds to the extensions (under a given semantics) of the given framework.

2 Preliminaries

We first recall Dung’s framework and then introduce the general class of recursive bipolar argumentation frameworks. We also briefly recall logic programs and partial stable semantics.

2.1 Argumentation Framework

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.

Different semantics notions have been defined leading to the characterization of collectively acceptable sets of arguments, called *extensions* [Dung, 1995]. 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} are defined as follows (where Δ is understood):

- $Def(\mathbf{S}) = \{a \in A \mid \exists b \in \mathbf{S}. (b, a) \in \Omega\}$;
- $Acc(\mathbf{S}) = \{a \in A \mid \forall b \in A. (b, a) \in \Omega \Rightarrow b \in Def(\mathbf{S})\}$.

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

Given an AF $\langle A, \Omega \rangle$, a set $\mathbf{S} \subseteq A$ is an *extension* called: *complete* iff it is conflict-free and $\mathbf{S} = Acc(\mathbf{S})$; *preferred* iff it is a \subseteq -maximal complete extension; *stable* iff it is a total preferred extension, i.e. a preferred extension such that $\mathbf{S} \cup Def(\mathbf{S}) = A$; *semi-stable* iff it is a preferred extension such that $\mathbf{S} \cup Def(\mathbf{S})$ is \subseteq -maximal; *grounded* iff it is the \subseteq -smallest complete extension; *ideal* iff it is the \subseteq -biggest complete extension contained in every preferred extension.

We use $\mathcal{CO}(\Delta)$ (resp., $\mathcal{PR}(\Delta)$, $\mathcal{ST}(\Delta)$, $\mathcal{SST}(\Delta)$, $\mathcal{GR}(\Delta)$, $\mathcal{ID}(\Delta)$) to denote the set of complete (resp., preferred, stable, semi-stable, grounded, ideal) extensions of a framework Δ .

2.2 Recursive Bipolar AFs

A *Recursive Bipolar Argumentation Framework* (Rec-BAF) is a tuple $\langle A, \Sigma, \Pi, s, t \rangle$, where A is a set of arguments, Σ is a set of attack names, Π is a set of support names, s (resp., t) is a function from $\Sigma \cup \Pi$ to A (resp., to $A \cup \Sigma \cup \Pi$) mapping each attack/support to its source (resp., target). In the following, given a set Φ such that either $\Phi \subseteq \Sigma$ or $\Phi \subseteq \Pi$, we denote by *i)* $\Phi^* = \{(s(\gamma), t(\gamma)) \mid \gamma \in \Phi\}$ the set of pairs connected by an attack/support edge, and *ii)* Φ^+ the transitive closure of Φ . It is assumed that Π^* is acyclic.

Two different semantics have been defined under the necessary interpretation of supports, as recalled in what follows.

Recursive AF with Necessities. The *Recursive Argumentation Framework with Necessities* (RAF_N) has been proposed in [Cayrol *et al.*, 2018]. The semantics combines the RAF interpretation of attacks [Cayrol *et al.*, 2017] with that of BAF under the necessity interpretation of supports (i.e. AF_N) [Nouioua and Risch, 2011]. Here we consider a simplified version where supports have a single source and the support relation is acyclic. Formally, given an RAF_N $\langle A, \Sigma, \Pi, s, t \rangle$, $X \in (A \cup \Sigma \cup \Pi)$, $a \in A$, and $\mathbf{S} \subseteq A \cup \Sigma \cup \Pi$, we say that argument a *recursively attacks* X given \mathbf{S} (denoted as $a \text{ att}_{\mathbf{S}} X$) if either $(a, X) \in (\Sigma \cap \mathbf{S})^*$ or there exists $b \in A$ such that $(a, b) \in (\Sigma \cap \mathbf{S})^*$ and $(b, X) \in (\Pi \cap \mathbf{S})^+$.

For any RAF_N Δ and $\mathbf{S} \subseteq A \cup \Sigma \cup \Pi$, the *defeated* and *acceptable* sets (given \mathbf{S}) are defined as follows:

- $Def(\mathbf{S}) = \{X \in A \cup \Sigma \cup \Pi \mid \exists b \in A \cap \mathbf{S}. b \text{ att}_{\mathbf{S}} X\}$;
- $Acc(\mathbf{S}) = \{X \in A \cup \Sigma \cup \Pi \mid \forall b \in A. b \text{ att}_{\mathbf{S}} X \Rightarrow b \in Def(\mathbf{S})\}$.

Attack-Support AF. The *Attack-Support Argumentation Framework* (ASAF) has been proposed in [Cohen *et al.*, 2015; Gottifredi *et al.*, 2018]. The semantics combines the AFRA interpretation of attacks [Baroni *et al.*, 2011] with that of BAF under the necessary interpretation of supports (i.e. AF_N). For the sake of presentation, we consider a slight generalization of ASAF, 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.

Formally, given an ASAF $\langle A, \Sigma, \Pi, s, t \rangle$, $X \in (A \cup \Sigma \cup \Pi)$, $\alpha \in \Sigma$, and $\mathbf{S} \subseteq A \cup \Sigma \cup \Pi$, we say that *i)* α *directly or indirectly attacks* X (denoted by $\alpha \text{ def } X$) if either $t(\alpha) = X$ or $t(\alpha) = s(X)$, and *ii)* α *extendedly defeats* X given \mathbf{S} (denoted as $\alpha \text{ def}_{\mathbf{S}} X$) if either $\alpha \text{ def } X$ or there exists $b \in A$ such that $t(\alpha) = b$ and either $(b, X) \in (\Pi \cap \mathbf{S})^+$ or $(b, s(X)) \in (\Pi \cap \mathbf{S})^+$. For any ASAF Δ and $\mathbf{S} \subseteq A \cup \Sigma \cup \Pi$, the *defeated* and *acceptable* sets (given \mathbf{S}) are as follows:

- $Def(\mathbf{S}) = \{X \in A \cup \Sigma \cup \Pi \mid \exists \alpha \in \Sigma \cap \mathbf{S}. \alpha \text{ def}_{\mathbf{S}} X\}$;
- $Acc(\mathbf{S}) = \{X \in A \cup \Sigma \cup \Pi \mid \forall \alpha \in \Sigma. \alpha \text{ def}_{\mathbf{S}} X \Rightarrow \alpha \in Def(\mathbf{S})\}$.

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 $\mathbf{S} \subseteq A \cup \Sigma \cup \Pi$ and by using the definitions of defeated and acceptable sets reported above.

Example 2 For the framework Δ of Example 1, under RAF_N semantics $\mathcal{CO}(\Delta) = \{E_1 = \{o, s, p, \alpha_1, \alpha_2, \alpha_3, \beta_1\}, E_2 = \{o, r, \alpha_1, \alpha_2, \alpha_3, \beta_1\}\}$. Differently, under ASAF semantics $\mathcal{CO}(\Delta) = \{E_3 = \{o, s, p, \alpha_3, \beta_1\}, E_4 = \{o, r, \alpha_1, \alpha_2, \beta_1\}\}$. Note that α_1 and α_2 (resp., α_3) do not belong to E_3 (resp. E_4) since their source, r (resp., s), is defeated.

AFs with high-order interactions can be mapped to AFs, though the mapping is not trivial because extensions also contain attacks and supports. In particular, an equivalent AF for an ASAF can be obtained by translating it into an AFN [Cohen *et al.*, 2015] that in turns can be translated into an AF [Nouioua and Risch, 2011; Gottifredi *et al.*, 2018].

AF-based frameworks belonging to the class of Rec-BAFs. It is important to observe that different frameworks extending AF share the same structure, although they have different semantics. Thus 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 AFN and AFD are two frameworks sharing the BAF syntax; their semantics differ because they interpret supports in different ways. *Recursive AF (Rec-AF)* is another syntactic class, where AFs are extended by allowing recursive attacks: AFRA and RAF are frameworks belonging to this class, which differ only in the determination of attacks' status. Finally, ASAF and RAFN are two frameworks belonging to the general class Rec-BAF presented in this section, consisting in the extension of BAF with recursive attacks and supports.

Hereafter, we focus on necessary supports only, even if the approach discussed in next section has been extended to cope with *deductive* supports in two frameworks: *Recursive Argumentation Framework with Deductive supports (RAFD)*, which extends RAF, and *Argumentation Framework with Recursive Attacks and Deductive supports (AFRAD)*, which extends AFRA. AFDs are special cases of these frameworks. We refer the reader to [Alfano *et al.*, 2020c] for details.

Thus, the class of Rec-BAF includes ASAF, RAFN, AFRAD, and RAFD, and their specializations AF, AFRA, RAF, AFN, and AFD. We use \mathfrak{F} to denote the set of these 9 frameworks.

2.3 Propositional Logic Programs

Given a set of symbols $\Lambda = \{a_1, \dots, a_n\}$, a (*propositional*) program over Λ is a set of $|\Lambda|$ rules $a_i \leftarrow \text{body}_i$ ($1 \leq i \leq n$), where every *body_i* is a propositional formula defined over Λ . The semantics of a propositional program P , defined over a given alphabet Λ , is given in terms of the set $\mathcal{PS}(P)$ of its Partial Stable Models (PSMs) [Saccà and Zaniolo, 1990].

The set of PSMs of a logic program P , denoted by $\mathcal{PM}(P)$, define a meet semi-lattice. The *well-founded* model (denoted by $\mathcal{WF}(P)$) and the *maximal-stable* models $\mathcal{MS}(P)$ ¹, are defined by considering \subseteq -minimal and \subseteq -maximal elements. The set of (total) *stable* models (denoted by $\mathcal{SM}(P)$) is obtained by considering the maximal-stable models which are total, whereas the *least-undefined* models (denoted by $\mathcal{LM}(P)$) are obtained by considering the maximal-stable models with a \subseteq -minimal set of undefined atoms (i.e. atoms which are neither true or false). The *max-deterministic* model (denoted by $\mathcal{MD}(P)$) is the \subseteq -maximal PSM contained in every maximal-stable model [Saccà, 1997; Greco and Saccà, 1999].

¹Corresponding to the *preferred extensions* of [Dung, 1991].

3 AF-based Frameworks and PSMs

We present a new way to define the semantics of AF-based frameworks by considering propositional programs and PSMs. To compare extensions E of a given framework Δ (containing acceptable elements) with PSMs of a given program P (containing true and false atoms), we denote by $\widehat{E} = E \cup \{\neg a \mid a \in \text{Def}(E)\}$ the *completion* of E . For a collection of extensions \mathbf{E} , $\widehat{\mathbf{E}}$ denotes the set $\{\widehat{E} \mid E \in \mathbf{E}\}$.

A first relationship between AF-based frameworks (e.g. BAF, Rec-BAF) and logic programs with PSMs is as follows.

Proposition 1 *For any framework $\Delta \in \mathfrak{F}$ and propositional program P , whenever $\widehat{\mathcal{CO}(\Delta)} = \mathcal{PS}(P)$ it holds that $\widehat{\mathcal{PR}(\Delta)} = \mathcal{MS}(P)$, $\widehat{\mathcal{ST}(\Delta)} = \mathcal{ST}(P)$, $\widehat{\mathcal{SST}(\Delta)} = \mathcal{LM}(P)$, $\widehat{\mathcal{GR}(\Delta)} = \mathcal{WF}(P)$, and $\widehat{\mathcal{ID}(\Delta)} = \mathcal{MD}(P)$.*

The result of Proposition 1 derives from the fact that preferred, stable, semi-stable, grounded, and ideal extensions are defined by selecting a subset of the complete extensions satisfying given criteria (see Section 2). On the other side, the maximal, stable, least-undefined, well-founded, and max-deterministic (partial) stable models are obtained by selecting a subset of the PSMs satisfying criteria coinciding with those used to restrict the set of complete extensions.

Given a framework Δ and an extension E , for any element a which could occur in some extension of Δ , the truth value $v_E(a)$, or simply $v(a)$ whenever E is understood, is equal to **true** if $a \in E$, **false** if $a \in \text{Def}(E)$, **undec** (*undecided*) otherwise. Hereafter, we assume that **false** < **undec** < **true** and $\neg\text{undec} = \text{undec}$.

The relationship between the semantics of AFs (given in terms of subset of complete extensions) and the semantics of logic programs (given in terms of subset of PSMs) has been shown in [Wu *et al.*, 2009; Caminada *et al.*, 2015]. The relationship is based on the observation that the meaning of an attack $a \rightarrow b$ is that the condition $v(b) \leq \neg v(a)$ must hold. On the other side, the satisfaction of a logical rule $a \leftarrow b_1, \dots, b_n$ implies that $v(a) \geq \min\{v(b_1), \dots, v(b_n), \text{true}\}$.

Definition 1 *Given AF $\Delta = \langle A, \Omega \rangle$, $P_\Delta = \{a \leftarrow \bigwedge_{(b,a) \in \Omega} \neg b \mid a \in A\}$ is the program derived from Δ .*

The semantics of an AF Δ can be obtained by considering PSMs of the logic program P_Δ . In fact, $\widehat{\mathcal{CO}(\Delta)} = \mathcal{PS}(P_\Delta)$.

In the rest of this section, we show how the semantics defined for frameworks extending AF can be captured by means of PSMs of logic programs. We propose a general method that can be applied to all the discussed frameworks. Specifically, to model frameworks extending Dung's framework by logic programs under PSM semantics, we provide new definitions of defeated and acceptable sets that, for a given set \mathbf{S} , will be denoted by $\text{DEF}(\mathbf{S})$ and $\text{ACC}(\mathbf{S})$, respectively. These definitions will be used to derive rules in P_Δ , the propositional program for $\Delta \in \mathfrak{F}$. For AFs we have that, for every set $\mathbf{S} \subseteq A$, $\text{DEF}(\mathbf{S}) = \text{Def}(\mathbf{S})$ and $\text{ACC}(\mathbf{S}) = \text{Acc}(\mathbf{S})$.

3.1 Recursive BAFs with Necessary Supports

We now study the relationship between PSMs and the semantics of Rec-BAFs. We first present results for RAFN semantics, and then we discuss results for the ASAF framework.

RAFNS. We next provide the definitions of defeated and acceptable sets for an RAFN.

Definition 2 For any RAFN $\langle A, \Sigma, \Pi, s, t \rangle$ and set $S \subseteq A \cup \Sigma \cup \Pi$, we have that:

- $\text{DEF}(S) = \{X \in A \cup \Sigma \cup \Pi \mid$
 $(\exists \alpha \in \Sigma \cap S . s(\alpha) \in S \wedge t(\alpha) = X) \vee$
 $(\exists \beta \in \Pi \cap S . s(\beta) \in \text{DEF}(S) \wedge t(\beta) = X)\}$;
- $\text{ACC}(S) = \{X \in A \cup \Sigma \cup \Pi \mid$
 $(\forall \alpha \in \Sigma . t(\alpha) = X \Rightarrow (\alpha \in \text{DEF}(S) \vee s(\alpha) \in \text{DEF}(S))) \wedge$
 $(\forall \beta \in \Pi . t(\beta) = X \Rightarrow (\beta \in \text{DEF}(S) \vee s(\beta) \in \text{ACC}(S)))\}$.

Notice that $\text{DEF}(S)$ and $\text{ACC}(S)$ are defined recursively. If S is a complete extension we obtain the following result.

Theorem 1 Given an RAFN Δ and an extension $S \in \mathcal{CO}(\Delta)$, then $\text{Def}(S) = \text{DEF}(S)$ and $\text{Acc}(S) = \text{ACC}(S)$.

Theorem 1 states that in order to define the semantics for an RAFN Δ we can use acceptable sets $S = \text{ACC}(S)$. This is captured by the following definition, that shows how to derive a propositional program from an RAFN from Definition 2.

Definition 3 Given an RAFN $\Delta = \langle A, \Sigma, \Pi, s, t \rangle$, then P_Δ (the propositional program derived from Δ) contains, for each $X \in A \cup \Sigma \cup \Pi$, a rule

$$X \leftarrow \bigwedge_{\alpha \in \Sigma \wedge t(\alpha) = X} (\neg \alpha \vee \neg s(\alpha)) \wedge \bigwedge_{\beta \in \Pi \wedge t(\beta) = X} (\neg \beta \vee s(\beta)).$$

Intuitively, starting from the definition of $\text{ACC}(S)$ in Definition 2, the rationale of the above definition is as follows. An element $X \in A \cup \Sigma \cup \Pi$ is true if (i) every attack α toward X is false or originates from a source $s(\alpha)$ which is false, and (ii) every support β toward X is false or originates from a source $s(\beta)$ which is true. These conditions resemble the conditions $(\forall \alpha \in \Sigma . t(\alpha) = X \Rightarrow (\alpha \in \text{DEF}(S) \vee s(\alpha) \in \text{DEF}(S)))$ and $(\forall \beta \in \Pi . t(\beta) = X \Rightarrow (\beta \in \text{DEF}(S) \vee s(\beta) \in \text{ACC}(S)))$, respectively, of Definition 2 after interpreting elements in $\text{DEF}(S)$ as false and elements in $\text{ACC}(S)$ as true.

As stated below, the set of complete extensions of an RAFN Δ coincides with the set of PSMs of P_Δ .

Theorem 2 For any RAFN Δ , $\widehat{\mathcal{CO}(\Delta)} = \mathcal{PS}(P_\Delta)$.

Thus, using Proposition 1, also the others argumentation semantics turns out to be characterized in terms of subsets of PSMs. Notice that, previous results also apply to restricted frameworks such as RAF, where $\Pi = \emptyset$, and AFN, where $t : \Sigma \cup \Pi \rightarrow A$.

ASAF. We next provide definitions of defeated and acceptable sets for an ASAF. They will be used in a way similar to that described above to obtain a semantically-equivalent propositional program for ASAFs.

Definition 4 Given an ASAF $\langle A, \Sigma, \Pi, s, t \rangle$ and a set $S \subseteq A \cup \Sigma \cup \Pi$, we define:

- $\text{DEF}(S) = \{X \in A \cup \Sigma \cup \Pi \mid (X \in \Sigma \wedge s(X) \in \text{DEF}(S)) \vee$
 $(\exists \alpha \in \Sigma \cap S . t(\alpha) = X) \vee$
 $(\exists \beta \in \Pi \cap S . t(\beta) = X \wedge s(\beta) \in \text{DEF}(S))\}$;
- $\text{ACC}(S) = \{X \in A \cup \Sigma \cup \Pi \mid (X \in \Sigma \Rightarrow s(X) \in \text{ACC}(S)) \wedge$
 $(\forall \alpha \in \Sigma . t(\alpha) = X \Rightarrow \alpha \in \text{DEF}(S)) \wedge$
 $(\forall \beta \in \Pi . t(\beta) = X \Rightarrow (\beta \in \text{DEF}(S) \vee s(\beta) \in \text{ACC}(S)))\}$.

Analogously to what has been done RAFNs, since it can be shown that for any complete extension S of an ASAF Δ it is the case that $\text{Acc}(S) = \text{ACC}(S)$, the program P_Δ for Δ can be derived by looking at the definition $\text{ACC}(S)$ of acceptable elements. In this case, the three conjuncts in the acceptance condition for an element X will correspond to three (group of) conjuncts, respectively, in the rule for X of P_Δ . Specifically, the last two conjuncts in the definition of $\text{ACC}(S)$ can be mapped by reasoning similarly to the RAFN case, whereas the first one entails a rule's body conjunction stating that if X is an attack then all of its sources must be true.

Definition 5 For any ASAF $\Delta = \langle A, \Sigma, \Pi, s, t \rangle$, P_Δ (the propositional program derived from Δ) contains, for each $X \in A \cup \Sigma \cup \Pi$, a rule of the form

$$X \leftarrow \varphi(X) \wedge \bigwedge_{\alpha \in \Sigma \wedge t(\alpha) = X} \neg \alpha \wedge \bigwedge_{\beta \in \Pi \wedge t(\beta) = X} (\neg \beta \vee s(\beta))$$

where: $\varphi(X) = s(X)$ if $X \in \Sigma$; otherwise, $\varphi(X) = \text{true}$.

The set of complete extensions of an ASAF Δ coincides with the set of PSMs of P_Δ , meaning that using Proposition 1 also the others argumentation semantics turns out to be characterized in terms of subsets of PSMs.

Theorem 3 For any ASAF Δ , $\widehat{\mathcal{CO}(\Delta)} = \mathcal{PS}(P_\Delta)$.

The above results also apply to restricted frameworks such as AFRA, where $\Pi = \emptyset$, and AFN, where $t : \Sigma \cup \Pi \rightarrow A$.

Example 3 Consider Δ of Example 1, whose complete extensions under RAFN and ASAF semantics are given in Example 2. The program under the RAFN semantics is $P_\Delta = \{(o \leftarrow), (r \leftarrow \neg \alpha_3 \vee \neg s), (p \leftarrow \neg \alpha_1 \vee \neg r), (s \leftarrow \neg \alpha_2 \vee \neg r), (\alpha_1 \leftarrow \neg \beta_1 \vee o), (\alpha_2 \leftarrow), (\alpha_3 \leftarrow), (\beta_1 \leftarrow)\}$, whose set of PSM is $\mathcal{PS}(P_\Delta) = \{M_1 = \{o, \neg r, s, p, \alpha_1, \alpha_2, \alpha_3, \beta_1\}, M_2 = \{o, r, \neg s, \neg p, \alpha_1, \alpha_2, \alpha_3, \beta_1\}\}$. For ASAF semantics, $P_\Delta = \{(o \leftarrow), (r \leftarrow \neg \alpha_3), (p \leftarrow \neg \alpha_1), (s \leftarrow \neg \alpha_2), (\alpha_1 \leftarrow r \wedge (\neg \beta_1 \vee o)), (\alpha_2 \leftarrow r), (\alpha_3 \leftarrow s), (\beta_1 \leftarrow)\}$, whose set of PSMs is $\mathcal{PS}(P_\Delta) = \{M_3 = \{o, \neg r, s, p, \neg \alpha_1, \neg \alpha_2, \alpha_3, \beta_1\}, M_4 = \{o, r, \neg s, \neg p, \alpha_1, \alpha_2, \neg \alpha_3, \beta_1\}\}$. Note that M_1 (resp., M_2) differs from M_3 (resp., M_4) in the status of α_1 and α_2 (resp., α_3).

4 Conclusions and Future Work

We have proposed a general framework that can be used i) for better understanding the semantics of several AF-based frameworks, ii) to easily define new semantics for extended frameworks, and iii) to provide additional tools for computing stable semantics using answer set solvers [Gebser *et al.*, 2018], as well as other complete-based semantics using classical program rewriting [Janhunen *et al.*, 2006; Sakama and Rienstra, 2017; Gaggl *et al.*, 2015].

Future work will be devoted to generalize our approach to also deal with AF-based framework with probabilities [Fazzinga *et al.*, 2015; Alfano *et al.*, 2020a], weights [Bistarelli *et al.*, 2018], preferences [Amgoud and Vesic, 2011; Modgil, 2009], and constraints [Alfano *et al.*, 2021b]. Finally, we plan to investigate incremental techniques tailored at using our approach to compute extensions of dynamic AF-based frameworks, where the sets of arguments and interactions change over the time [Alfano *et al.*, 2020b; Alfano *et al.*, 2021a].

References

- [Alfano *et al.*, 2020a] G. Alfano, M. Calautti, S. Greco, F. Parisi, and I. Trubitsyna. Explainable acceptance in probabilistic abstract argumentation: Complexity and approximation. In *Proc. of KR*, pages 33–43, 2020.
- [Alfano *et al.*, 2020b] G. Alfano, A. Cohen, S. Gottifredi, S. Greco, F. Parisi, and G. R. Simari. Dynamics in abstract argumentation frameworks with recursive attack and support relations. In *ECAI*, pages 577–584, 2020.
- [Alfano *et al.*, 2020c] G. Alfano, S. Greco, F. Parisi, and I. Trubitsyna. On the semantics of abstract argumentation frameworks: A logic programming approach. *TPLP*, 20(5):703–718, 2020.
- [Alfano *et al.*, 2021a] G. Alfano, S. Greco, and F. Parisi. Incremental computation in dynamic argumentation frameworks. *IEEE Intelligent Systems*, 2021.
- [Alfano *et al.*, 2021b] G. Alfano, S. Greco, F. Parisi, and I. Trubitsyna. Argumentation frameworks with strong and weak constraints: Semantics and complexity. In *Proc. of AAAI*, page (to appear), 2021.
- [Amgoud and Vesic, 2011] L. Amgoud and S. Vesic. A new approach for preference-based argumentation frameworks. *Ann. Math. Artif. Intell.*, 63(2):149–183, 2011.
- [Baroni *et al.*, 2011] P. Baroni, F. Cerutti, M. Giacomin, and G. Guida. AFRA: Argumentation Framework with Recursive Attacks. *IJAR*, 52(1):19–37, 2011.
- [Bistarelli *et al.*, 2018] S. Bistarelli, F. Rossi, and F. Santini. A novel weighted defence and its relaxation in abstract argumentation. *IJAR*, 92:66–86, 2018.
- [Caminada *et al.*, 2015] M. Caminada, S. Sá, J. F. L. Alcântara, and W. Dvorák. On the equivalence between logic programming semantics and argumentation semantics. *IJAR*, 58:87–111, 2015.
- [Cayrol and Lagasque-Schiex, 2013] C. Cayrol and M.-C. Lagasque-Schiex. Bipolarity in argumentation graphs: Towards a better understanding. *IJAR*, 54(7):876–899, 2013.
- [Cayrol *et al.*, 2017] C. Cayrol, J. Fandinno, L. Fariñas del Cerro, and M.-C. Lagasque-Schiex. Valid attacks in argumentation frameworks with recursive attacks. In *Proc. of COMMONSENSE*, 2017.
- [Cayrol *et al.*, 2018] C. Cayrol, J. Fandinno, L. Fariñas del Cerro, and M.-C. Lagasque-Schiex. Structure-based semantics of argumentation frameworks with higher-order attacks and supports. In *COMMA*, pages 29–36, 2018.
- [Cohen *et al.*, 2014] A. Cohen, S. Gottifredi, A. J. Garcia, and G. R. Simari. A survey of different approaches to support in argumentation systems. *The Know. Eng. Rev.*, 29(5):513–550, 2014.
- [Cohen *et al.*, 2015] A. Cohen, S. Gottifredi, A. J. Garcia, and G. R. Simari. An approach to abstract argumentation with recursive attack and support. *J. Appl. Log.*, 13(4):509–533, 2015.
- [Dung, 1991] P. M. Dung. Negations as hypotheses: An abductive foundation for logic programming. In *ICLP*, pages 3–17, 1991.
- [Dung, 1995] P. M. Dung. On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games. *AI*, 77:321–358, 1995.
- [Fazzinga *et al.*, 2015] B. Fazzinga, S. Flesca, and F. Parisi. On the complexity of probabilistic abstract argumentation frameworks. *TOCL*, 16(3):22:1–22:39, 2015.
- [Gaggl *et al.*, 2015] S. A. Gaggl, N. Manthey, A. Ronca, J. P. Wallner, and S. Woltran. Improved answer-set programming encodings for abstract argumentation. *TPLP*, 15(4-5):434–448, 2015.
- [Gebser *et al.*, 2018] M. Gebser, N. Leone, M. Maratea, S. Perri, F. Ricca, and T. Schaub. Evaluation techniques and systems for answer set programming: a survey. In *IJCAI*, pages 5450–5456, 2018.
- [Gelfond and Lifschitz, 1988] M. Gelfond and V. Lifschitz. The stable model semantics for logic programming. In *ICSLP*, pages 1070–1080, 1988.
- [Gottifredi *et al.*, 2018] S. Gottifredi, A. Cohen, A. Javier Garcia, and G. R. Simari. Characterizing acceptability semantics of argumentation frameworks with recursive attack and support relations. *AI*, 262:336–368, 2018.
- [Greco and Saccà, 1999] S. Greco and D. Saccà. Complexity and expressive power of deterministic semantics for datalog \neg . *Inf. Comput.*, 153(1):81–98, 1999.
- [Janhunen *et al.*, 2006] T. Janhunen, I. Niemelä, D. Seipel, P. Simons, and J.-H. You. Unfolding partiality and disjunctions in stable model semantics. *ACM Trans. Comput. Logic*, 7(1), 2006.
- [Modgil, 2009] Sanjay Modgil. Reasoning about preferences in argumentation frameworks. *AI*, 173(9-10):901–934, 2009.
- [Nouioua and Risch, 2011] F. Nouioua and V. Risch. Argumentation frameworks with necessities. In *SUM*, pages 163–176, 2011.
- [Saccà and Zaniolo, 1990] D. Saccà and C. Zaniolo. Stable models and non-determinism in logic programs with negation. In *PODS*, pages 205–217, 1990.
- [Saccà, 1997] D. Saccà. The expressive powers of stable models for bound and unbound DATALOG queries. *J. Comput. Syst. Sci.*, 54(3):441–464, 1997.
- [Sakama and Rienstra, 2017] C. Sakama and T. Rienstra. Representing argumentation frameworks in answer set programming. *Fundam. Inform.*, 155(3):261–292, 2017.
- [Villata *et al.*, 2012] S. Villata, G. Boella, D. M. Gabbay, and L. W. N. van der Torre. Modelling defeasible and prioritized support in bipolar argumentation. *AMAI*, 66(1-4):163–197, 2012.
- [Wu *et al.*, 2009] Y. Wu, M. Caminada, and D. M. Gabbay. Complete extensions in argumentation coincide with 3-valued stable models in logic programming. *Studia Logica*, 93(2-3):383–403, 2009.