On the Semantics of Recursive Bipolar AFs and Partial Stable Models

Gianvincenzo Alfano, Sergio Greco, Francesco Parisi and Irina Trubitsyna

{g.alfano, greco, fparisi, trubitsyna}@dimes.unical.it
DIMES Department, University of Calabria, Italy

4^{th} Workshop on Advances In Argumentation In Artificial Intelligence

November 25 - 26, 2020

(Virtual Event)

Abstract Argumentation Framework (AF)

Abstract Argumentation Framework (AF) [Dung1995]

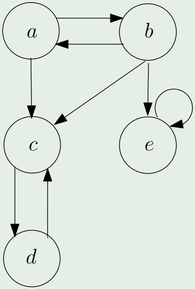
Arguments are abstract entities (no attention is paid to their internal structure) that may attack and/or be attacked by other arguments.
Formally, an AF is a pair $\mathcal{A} = \langle A, \Sigma \rangle$, where:

- A is a set of arguments, and
- $\Sigma \subseteq A \times A$ is a set of attacks.

Example (a simple AF $\mathcal{A}$)

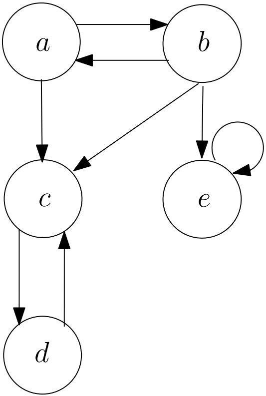
$\mathcal{A} = \langle A, \Sigma \rangle$ where
$A = \{a, b, c, d, e\}$ and
$\Sigma = \{(a, b), (b, a), (a, c), (b, c), (b, e), (e, e), (d, c), (c, d)\}$

An evaluation process is needed in order to conclude something.



Refinements of the Complete Extension

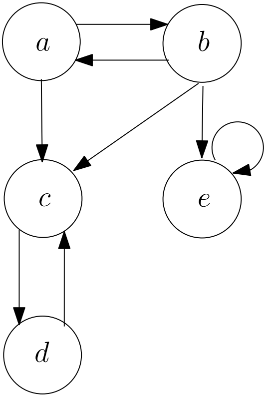
An Example

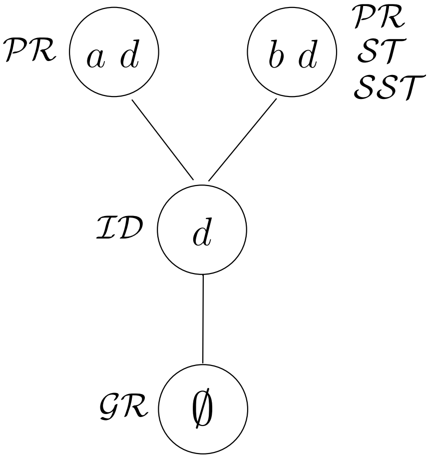


Semantic	Extensions	Refinement
complete	$\emptyset$ $\{d\}$ $\{a, d\}$ $\{b, d\}$	$\equiv$
preferred	$\{a, d\}$ $\{b, d\}$	maximal w.r.t $\subseteq$
semi-stable	$\{b, d\}$	minimal set of undecided args
stable	$\{b, d\}$	w/o UN args
ideal	$\{d\}$	maximal & contained in each $\mathtt{pr}$
grounded	$\emptyset$	minimal w.r.t $\subseteq$

Refinements of the Complete Extension

An Example





The set of complete extensions defines a meet semi-lattice

Computing Partial Stable Models (PSMs)

- A (normal) LP P is a set of rules of the form $A \leftarrow B_1 \wedge \cdots \wedge B_n$, with $n \geq 0$
- Given a (partial) interpretation $M \subseteq B_P \cup \neg B_P$, P^M is the positive instantiation of P w.r.t M obtained by replacing every negated body literal $\neg a$ with its truth value $\vartheta_M(\neg a)$ w.r.t. M

$$\vartheta_M(\neg a) \in \{True, False, Undef\}$$

- M is a Partial Stable Model (PSM) of P if it is the minimal model of P^M

Other Semantics

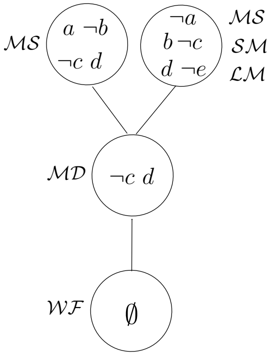
Program P:
$a \leftarrow \neg b;$
$b \leftarrow \neg a;$
$c \leftarrow \neg a, \neg b, \neg d;$
$d \leftarrow \neg c;$
$e \leftarrow \neg e, \neg b;$

Semantic	Extensions	Refinement
Partial Stable Model $\mathcal{PS}(M)$	$\emptyset$ $\{\neg c, d\}$ $\{a, \neg b, \neg c, d\}$ $\{\neg a, b, \neg c, d, \neg e\}$	$\equiv$
maximal-stable $\mathcal{MS}(P)$	$\{a, \neg b, \neg c, d\}$ $\{\neg a, b, \neg c, d, \neg e\}$	maximal w.r.t $\subseteq$
least-undefined $\mathcal{LM}(P)$	$\{\neg a, b, \neg c, d, \neg e\}$	minimal set of undefined atoms
total stable $\mathcal{SM}(P)$	$\{\neg a, b, \neg c, d, \neg e\}$	w/o undef atoms
max-deterministic $\mathcal{MD}(P)$	$\{\neg c, d\}$	maximal & $\in$ each $\mathcal{MS}(P)$
well-founded $\mathcal{WF}(P)$	$\emptyset$	minimal w.r.t $\subseteq$

Other Semantics

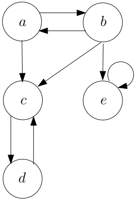
Program P:

$a \leftarrow \neg b;$
$b \leftarrow \neg a;$
$c \leftarrow \neg a, \neg b, \neg d;$
$d \leftarrow \neg c;$
$e \leftarrow \neg e, \neg b;$



The set of partial stable models of P defines a meet semi-lattice.

Analogies? Yes!



Program P:

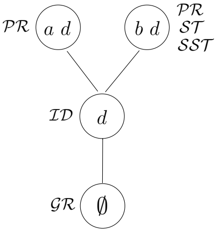
$a \leftarrow \neg b;$

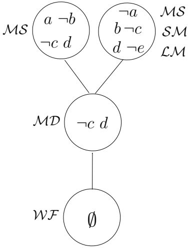
$b \leftarrow \neg a;$

$c \leftarrow \neg a, \neg b, \neg d;$

$d \leftarrow \neg c;$

$e \leftarrow \neg e, \neg b;$



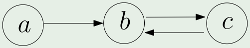


Relations with LP

A one-to-one correspondence between 3-valued stable models and complete extensions of an AF has been already proposed (Wu et al. 2009; Caminada et al. 2015). $\not\Leftarrow$ for $\mathcal{SST}$

$P_\Delta = \{a \leftarrow \bigwedge_{(b,a)\in\Omega} \neg b \mid a \in A\}$ is the propositional program derived from Δ.

Example



$a \leftarrow$

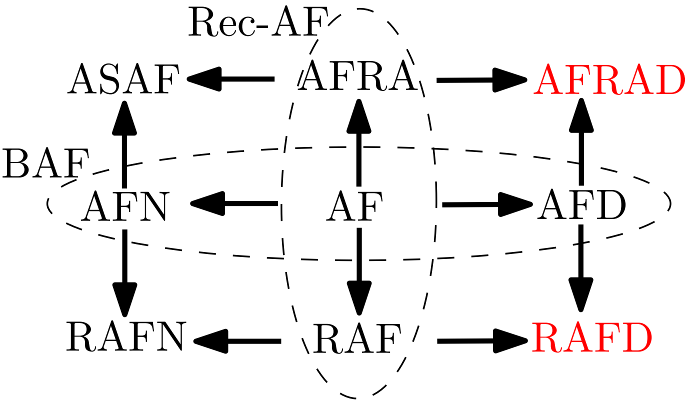
$b \leftarrow \neg c, \neg a$

$c \leftarrow \neg b$

$PSM = \widehat{\mathcal{CO}(\Delta)}$:
$\{\{a, c, \neg b\}\}$

Beyond Dung AF

Several Abstract Argumentation Frameworks extending Dung AF proposed in literature, and different ways to obtain extensions.



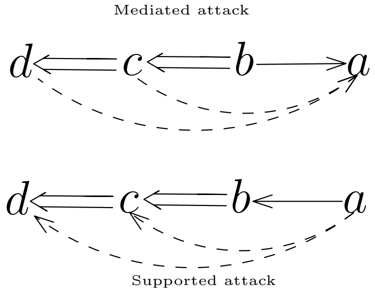
AF-based Semantics

- Sometimes the AF-based semantics are a bit difficult to understand, especially when approaching argumentation.
- The semantics can be given:

AF-based Semantics

- Sometimes the AF-based semantics are a bit difficult to understand, especially when approaching argumentation.
- The semantics can be given:

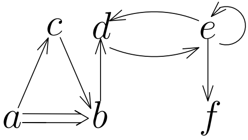
- Directly: hidden relations should be taken into account.

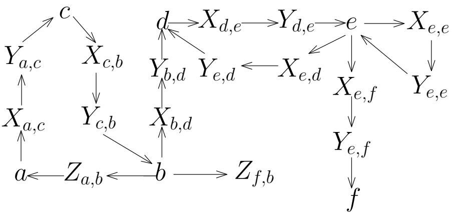


AF-based Semantics

- Sometimes the AF-based semantics are a bit difficult to understand, especially when approaching argumentation.
- The semantics can be given:

- Directly: hidden relations should be taken into account.
- Via meta-argumentation: several (fake) meta-arguments and meta-attacks are added.





AF-based Semantics

- Sometimes the AF-based semantics are a bit difficult to understand, especially when approaching argumentation.
- The semantics can be given:

- Directly: hidden relations should be taken into account.
- Via meta-argumentation: several (fake) meta-arguments and meta-attacks are added.

The same holds when moving on Rec-BAFs, but in a more complicated way due to the recursive interactions, which requires several definitions, loosing one of the key aspects of argumentation: simplicity.

Direct and Meta-AF Semantics for Rec-BAFs

(Unconditional Defeat)

...

(Support Sequence and Support Set)

...

(Conditional Defeat)

...

(Conflict-freeness)

...

(Acceptability)

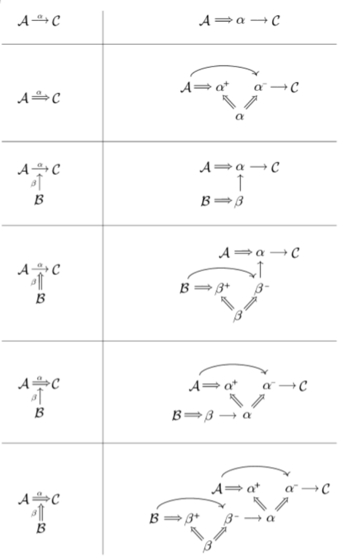
...

(Admissibility)

...

(*ASAF* Extensions)

...



What we Propose

- Sometimes the AF-based semantics are a bit difficult to understand, especially when approaching argumentation.
- The semantics can be given:

- Directly: hidden relations should be taken into account.
- Via meta-argumentation: several (fake) meta-arguments and meta-attacks are added.
- **Model semantics defined for frameworks extending AF by means of PSMs of logic programs**

Main Result

For any framework $\Delta \in \mathfrak{F}$ and a propositional program P,
whenever $\widehat{\mathcal{CO}(\Delta)} = \mathcal{PS}(P)$ it holds that :

$\widehat{\mathcal{PR}(\Delta)} = \mathcal{MS}(P)$

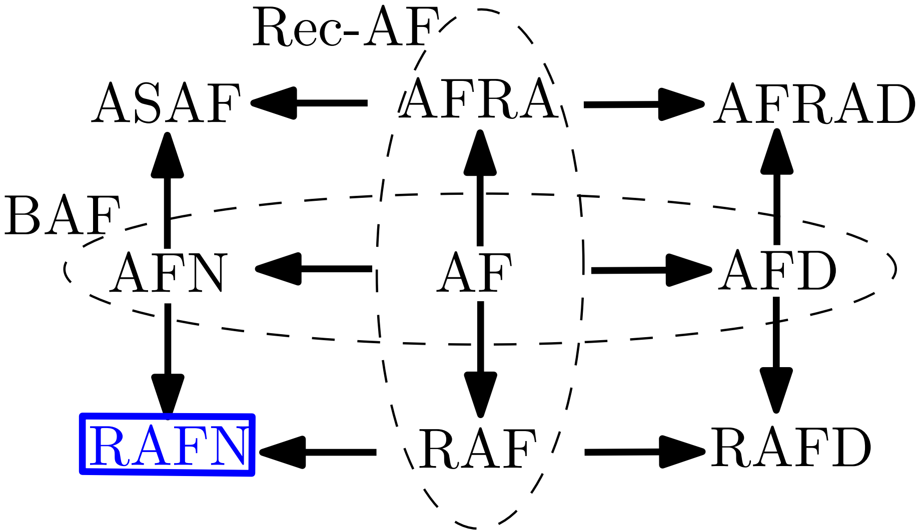
$\widehat{\mathcal{ST}(\Delta)} = \mathcal{SM}(P)$

$\widehat{\mathcal{SST}(\Delta)} = \mathcal{LM}(P)$

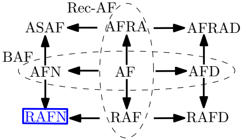
$\widehat{\mathcal{GR}(\Delta)} = \mathcal{WF}(P)$

$\widehat{\mathcal{ID}(\Delta)} = \mathcal{MD}(P)$

LP for AF-based frameworks: RAFN



Recursive AF with Necessities (RAFN)



(Corresponding Prop. Program of an RAFN)

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

(Equivalent Definition of defeated and acceptable sets)

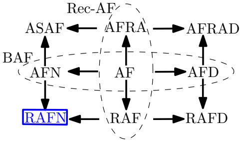
$$\begin{aligned}
&\textsc{Def}(S) = \{X \in A \cup \Sigma \cup \Pi \mid \\
&(\exists \alpha \in \Sigma \cap S \,.\, \mathbf{s}(\alpha) \in S \ \wedge \ \mathbf{t}(\alpha) = X) \vee \\
&(\exists \beta \in \Pi \cap S \,.\, \mathbf{s}(\beta) \in \textsc{Def}(S) \ \wedge \ \mathbf{t}(\beta) = X)\ \};
\end{aligned}$$

$$\begin{aligned}
&\textsc{Acc}(S) = \{X \in A \cup \Sigma \cup \Pi \mid \\
&(\forall \alpha \in \Sigma \,.\, \mathbf{t}(\alpha) = X \Rightarrow (\alpha \in \textsc{Def}(S) \vee \mathbf{s}(\alpha) \in \textsc{Def}(S))) \wedge \\
&(\forall \beta \in \Pi \,.\, \mathbf{t}(\beta) = X \Rightarrow (\beta \in \textsc{Def}(S) \vee \mathbf{s}(\beta) \in \textsc{Acc}(S)))\}.
\end{aligned}$$

Recursive AF with Necessities (RAFN)

(Corresponding Prop. Program of an RAFN)

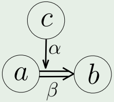
$$X \leftarrow \bigwedge_{\alpha \in \Sigma \wedge \mathbf{t}(\alpha)=X} (\neg\alpha \vee \neg \mathbf{s}(\alpha)) \wedge \bigwedge_{\beta \in \Pi \wedge \mathbf{t}(\beta)=X} (\neg\beta \vee \mathbf{s}(\beta)).$$



(Theorem)

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

Example

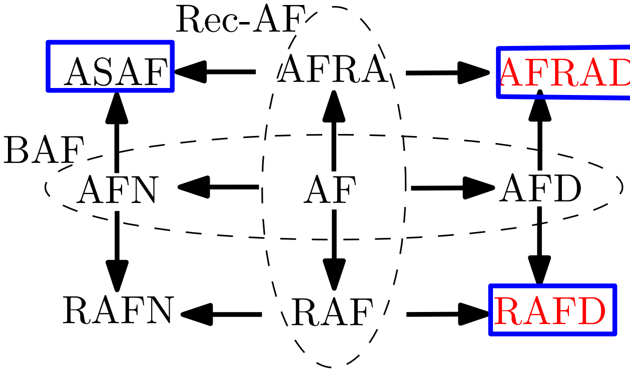


$$a \leftarrow$$
$$b \leftarrow \neg\beta \vee a$$
$$c \leftarrow$$
$$\alpha \leftarrow$$
$$\beta \leftarrow \neg\alpha \vee \neg c$$

$\widehat{\mathcal{CO}(\Delta)} = \mathcal{PS}(P_\Delta)$:
$\{\{a, b, c, \alpha, \neg\beta\}\}$

LP for (other) AF-based frameworks (1/2)

Same is done for the other AF-based frameworks.



LP for (other) AF-based frameworks (2/2)

(Corresponding Prop. Program of an ASAF)

$$X \leftarrow \varphi(X) \wedge \bigwedge_{\alpha\in\Sigma\wedge\mathbf{t}(\alpha)=X} \neg\alpha \wedge \bigwedge_{\beta\in\Pi\wedge\mathbf{t}(\beta)=X} (\neg\beta \vee \mathbf{s}(\beta)) \text{ where } \varphi(X) = \begin{cases} \mathbf{s}(X) \text{ if } X \in \Sigma \\ \texttt{true} \text{ otherwise} \end{cases}.$$

(Corresponding Prop. Program of an AFRAD)

$$X \leftarrow \varphi(X) \wedge \bigwedge_{\alpha\in\Sigma\wedge\mathbf{t}(\alpha)=X} \neg\alpha \wedge \bigwedge_{\beta\in\Pi\wedge\mathbf{s}(\beta)=X} (\neg\beta \vee \mathbf{t}(\beta)) \text{ where } \varphi(X) = \begin{cases} \mathbf{s}(X) \text{ if } X \in \Sigma \\ \texttt{true} \text{ otherwise.} \end{cases}$$

(Corresponding Prop. Program of an RAFD)

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

Conclusions and Future Work

- A simple & general logical framework able to capture in a systematic and succinct way different features of several AF-based frameworks under different argumentation semantics.
- The proposed approach can be used for better understanding the semantics of extended AF frameworks (sometimes a bit involved), and is flexible enough for encouraging the study of other extensions.
- Enabling the computation at the LP level: using ASP solvers for computing extensions in extended AFs.

FW) Generalize our logical approach to deal also with Probabilistic AF-based frameworks, weights, preferences, and considering supports with multiple sources.

Thank you!

... any ~~question~~ argument?