On the Semantics of Abstract Argumentation Frameworks: A Logic Programming Approach

Gianvincenzo Alfano, Sergio Greco, Francesco Parisi and Irina Trubitsyna

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

36^{th} International Conference on Logic Programming

September 18 - 24, 2020

(Virtual) Rende, Italy

What is Argumentation?

[Prakken 2011]

Argumentation is the process of supporting claims with grounds and defending them against attack.

[van Eemeren et al, 1996]

Argumentation is a verbal and social activity of reason aimed at increasing (or decreasing) the acceptability of a controversial standpoint for the listener or reader, by putting forward a constellation of propositions intended to justify (or refute) the standpoint before a rational judge.

Argumentation in AI

- Very active research area in AI.
- Useful to describe cooperating and competing systems.
- A general way for representing arguments and relationships (rebuttals) between them.
- A framework for practical and uncertain reasoning able to cope with partial and inconsistent knowledge.

Elements of an argumentation system

- The definition of argument (possibly including an underlying logical language + a notion of logical consequence)
- The notion of attack and defeat (successful attack) between arguments
- An argumentation semantics selecting acceptable (justified) arguments

Argumentation in AI

- Very active research area in AI.
- Useful to describe cooperating and competing systems.
- A general way for representing arguments and relationships (rebuttals) between them.
- A framework for practical and uncertain reasoning able to cope with partial and inconsistent knowledge.

Elements of an argumentation system

- The definition of argument (possibly including an underlying logical language + a notion of logical consequence)
- The notion of attack and defeat (successful attack) between arguments
- An argumentation semantics selecting acceptable (justified) arguments

What is argumentation (an example)

1) Constructing arguments (in favor of / against a “statement") from available information,

   A: “Tweety is a bird, so it flies"
   B: “Tweety is just a cartoon!"

2) Determining the different conflicts among the arguments.

   “Since Tweety is a cartoon, it cannot fly!" (B attacks A)

3) Evaluating the acceptability of the different arguments.

   “Since we have no reason to believe otherwise, we'll assume Tweety is a cartoon." (accept B).
   “But then, this means despite being a bird he cannot fly." (reject A).

4) Concluding, or defining the justified conclusions.

   “We conlcude that Tweety cannot fly!”

What is argumentation (an example)

1) Constructing arguments (in favor of / against a “statement") from available information,

    A: “Tweety is a bird, so it flies"
    B: “Tweety is just a cartoon!"

2) Determining the different conflicts among the arguments.

    “Since Tweety is a cartoon, it cannot fly!" (B attacks A)

3) Evaluating the acceptability of the different arguments.

    “Since we have no reason to believe otherwise, we'll assume Tweety is a cartoon." (accept B).
    “But then, this means despite being a bird he cannot fly." (reject A).

4) Concluding, or defining the justified conclusions.

    “We conlcude that Tweety cannot fly!”

What is argumentation (an example)

1) Constructing arguments (in favor of / against a “statement") from available information,

   A: “Tweety is a bird, so it flies"
   B: “Tweety is just a cartoon!"

2) Determining the different conflicts among the arguments.

   “Since Tweety is a cartoon, it cannot fly!" (B attacks A)

3) Evaluating the acceptability of the different arguments.

   “Since we have no reason to believe otherwise, we’ll assume Tweety is a cartoon." (accept B).
   “But then, this means despite being a bird he cannot fly." (reject A).

4) Concluding, or defining the justified conclusions.

   “We conlcude that Tweety cannot fly!”

What is argumentation (an example)

1) Constructing arguments (in favor of / against a “statement") from available information,

    A: “Tweety is a bird, so it flies"
    B: “Tweety is just a cartoon!"

2) Determining the different conflicts among the arguments.

    “Since Tweety is a cartoon, it cannot fly!" (B attacks A)

3) Evaluating the acceptability of the different arguments.

    “Since we have no reason to believe otherwise, we’ll assume Tweety is a cartoon." (accept B).
    “But then, this means despite being a bird he cannot fly." (reject A).

4) Concluding, or defining the justified conclusions.

    “We conlcude that Tweety cannot fly!”

Abstract Argumentation Framework (AF)

Abstract Argumentation Framework (AF) [Dung1995]

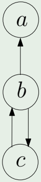
Arguments are abstract entities (no attention is paid to their internal structure) that may attack and/or be attacked by other arguments.
Formally, an AF is a pair $\mathcal{A} = \langle A, \Sigma \rangle$, where:

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

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

a = Our friends will have great fun at our party on Saturday
b = Saturday will rain (according to the weather forecasting service 1)
c = Saturday will be sunny (according to the weather forecasting service 2)

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



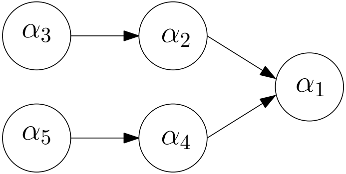
An evaluation process is needed in order to conclude something.

Collectively Evaluating Arguments

Intuitively:

A set of arguments is **conflict-free** if no argument in the set defeats another argument.

A set of arguments *defends* a given argument if it defeats all its defeaters.



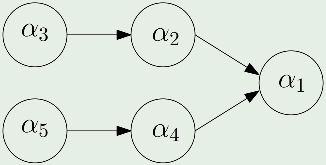
In the above graph, $\{\alpha_3, \alpha_5\}$ is conflict-free and defends α_1.

Characterizing Defense

Admissible set

A conflict-free set S is **admissible** if it defends every element in S.

Example



Sets $\emptyset$, $\{\alpha_3\}$, $\{\alpha_5\}$, and $\{\alpha_3, \alpha_5\}$ are all admissible simply because they do not have any defeaters.

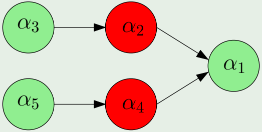
Set $\{\alpha_1, \alpha_3, \alpha_5\}$ is also admissible since it defends itself against defeaters α_2 and α_4.

Complete Extension

Complete Extension

An admissible set S of arguments in framework $\langle A, \Sigma \rangle$ is a **complete extension** if and only if *all* arguments defended by S are also in S.

Example (Complete Extension Example)

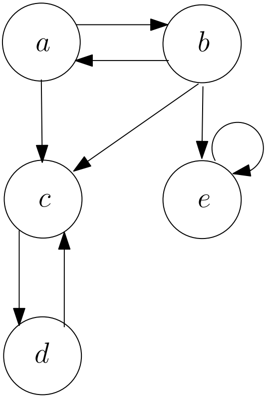


Admissible set $S_0 = \{\alpha_3, \alpha_5\}$ is not a complete extension, since it defends α_1 but does not include α_1.

Admissible set $S_3 = \{\alpha_1, \alpha_3, \alpha_5\}$ is the only complete extension.

Refinements of the Complete Extension

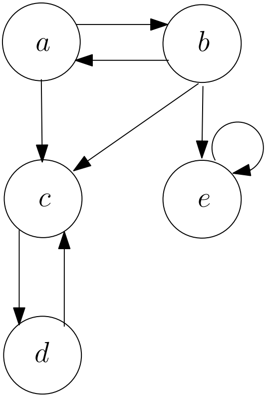
An Example

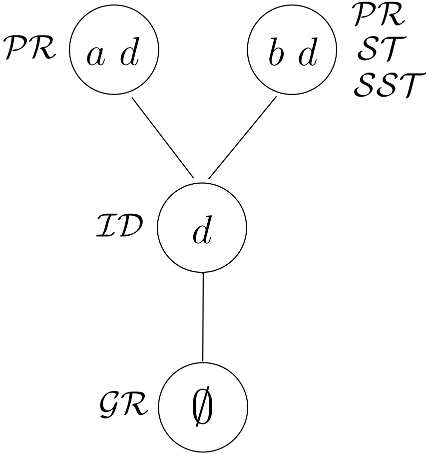


Semantic	Extensions	Refinement
complete	$\emptyset$ $\{d\}$ $\{a,d\}$ $\{b,d\}$	$\equiv$
preferred	$\{a,d\}$ $\{b,d\}$	maximal w.r.t $\subseteq$
semi-stable	$\{b,d\}$	minimal set of undecided args
stable	$\{b,d\}$	w/o UN args
ideal	$\{d\}$	maximal & contained in each $\mathtt{pr}$
grounded	$\emptyset$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

Refinements of the Complete Extension

An Example





The set of complete extensions defines a meet semi-lattice

Outline

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:

$a \leftarrow \neg b;$

$b \leftarrow \neg a;$

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

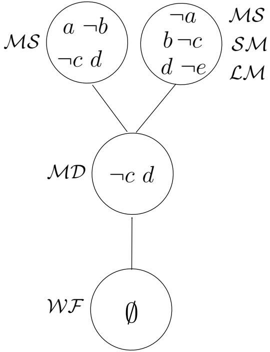
$d \leftarrow \neg c;$

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

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 $\subseteq$

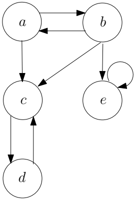
Other Semantics

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:

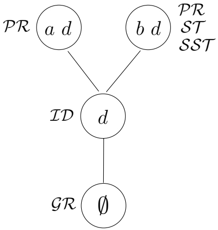
$a \leftarrow \neg b;$

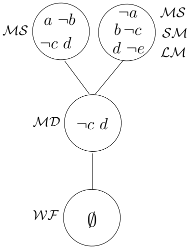
$b \leftarrow \neg a;$

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

$d \leftarrow \neg c;$

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





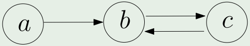
Outline

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



$a \leftarrow$

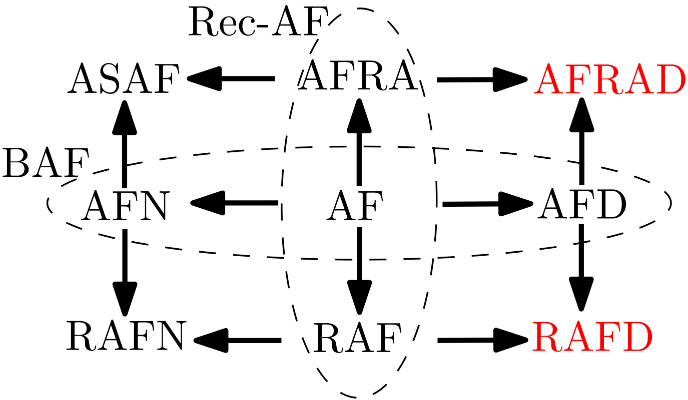
$b \leftarrow \neg c, \neg a$

$c \leftarrow \neg b$

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

Beyond Dung AF

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

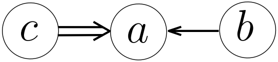


Bipolar AFs (BAFs)

- Also includes the notion of support between arguments.
- Two semantics defined: **AFN** and **AFD**.

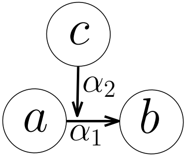
(BAF)

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*.



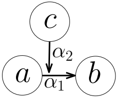
Recursive AFs

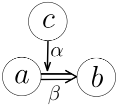
- Also includes the notion of recursive attack relations.
- Two semantics defined: **AFRA** and **RAF**.

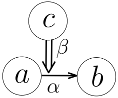


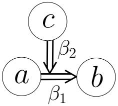
Recursive Bipolar AFs (Rec-BAFs)

- Combines the concepts of both bipolarity and recursive interactions.
- Two semantics are defined: *Recursive Argumentation Framework with Necessities (RAFN)* & *Attack-Support Argumentation Framework (ASAF)*.



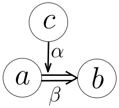






Recursive Bipolar AFs (Rec-BAFs)

- Combines the concepts of both bipolarity and recursive interactions.
- Two semantics are defined: *Recursive Argumentation Framework with Necessities (RAFN)* & *Attack-Support Argumentation Framework (ASAF)*.



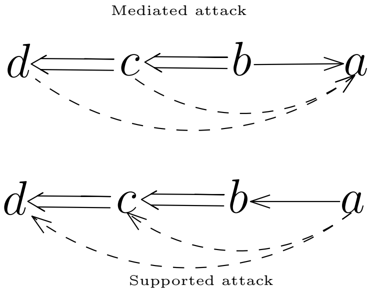
AF-based Semantics

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

AF-based Semantics

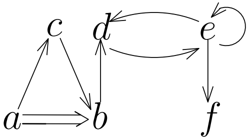
- Sometimes the AF-based semantics are a bit difficult to understand, especially when approaching argumentation.
- The semantics can be given:

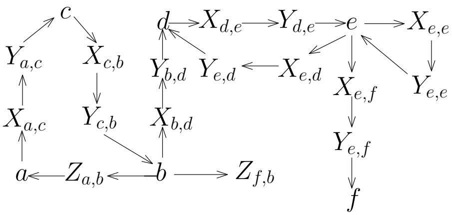
- Directly: hidden relations should be taken into account.



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)

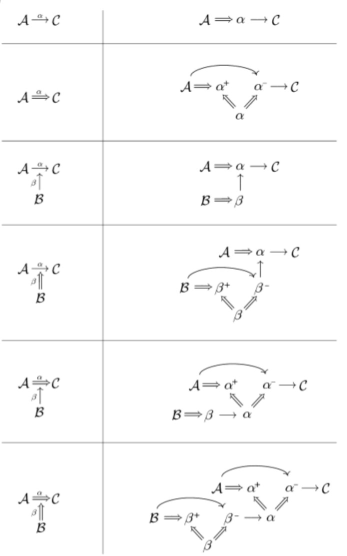
...

(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 :

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

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

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

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

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

- This is carried out by proposing novel (equivalent) definitions of acceptable and defeated arguments, for each AF-based framework.
- Intuitively, they are useful for defining different extensions (similarly to what done for AFs) as well as allowing to identify the corresponding propositional program.
- There is a correspondence between acceptable/defeated arguments and arguments appearing true/false in the PSM.
- This cannot be done w.r.t. classical definitions of defeated and acceptable arguments.

AFN: $a \Rightarrow b$ means that b is accepted only if a is accepted.

(Classical Definitions)

Given an AFN $\langle A, \Omega, \Gamma \rangle$, and a set of arguments $\mathbf{S} \subseteq A$, then
$Def(\mathbf{S}) = \{a \in A \mid \exists b \in \mathbf{S} \,.\, (b,a) \in \Omega_n\}$, and
$Acc(\mathbf{S}) = \{a \in A \mid \forall b \in A \,.\, (b,a) \in \Omega_n \Rightarrow b \in Def(\mathbf{S})\}$.

(Novel Definitions)

For any AFN $\langle A, \Omega, \Gamma \rangle$ and set of arguments $\mathbf{S} \subseteq A$,
$\textsc{Def}(\mathbf{S}) = \{a \in A \mid (\exists b \in \mathbf{S} \,.\, (b,a) \in \Omega) \vee (\exists c \in \textsc{Def}(\mathbf{S}) \,.\, (c,a) \in \Gamma)\}$;

$$\textsc{Acc}(\mathbf{S}) = \{a \in A \mid (\forall b \in A \,.\, (b,a) \in \Omega \Rightarrow b \in \textsc{Def}(\mathbf{S})) \wedge (\forall c \in A \,.\, (c,a) \in \Gamma \Rightarrow c \in \textsc{Acc}(\mathbf{S}))\}.$$

(Corresponding Prop. Program of an AFN)

$$P_\Delta = \{a \leftarrow (\bigwedge_{(b,a)\in\Omega} \neg b \ \wedge \ \bigwedge_{(c,a)\in\Gamma} c) \mid a \in A\}$$

AFN: $a \Rightarrow b$ means that b is accepted only if a is accepted.

(Classical Definitions)

Given an AFN $\langle A, \Omega, \Gamma \rangle$, and a set of arguments $\mathbf{S} \subseteq A$, then
$Def(\mathbf{S}) = \{a \in A \mid \exists b \in \mathbf{S}\,.\,(b,a) \in \Omega_n\}$, and
$Acc(\mathbf{S}) = \{a \in A \mid \forall b \in A\,.\,(b,a) \in \Omega_n \Rightarrow b \in Def(\mathbf{S})\}$.

(Novel Definitions)

For any AFN $\langle A, \Omega, \Gamma \rangle$ and set of arguments $\mathbf{S} \subseteq A$,
$\textsc{Def}(\mathbf{S}) = \{a \in A \mid (\exists b \in \mathbf{S}\,.\,(b,a) \in \Omega) \vee (\exists c \in \textsc{Def}(\mathbf{S})\,.\,(c,a) \in \Gamma)\}$;
$\textsc{Acc}(\mathbf{S}) = \{a \in A \mid (\forall b \in A\,.\,(b,a) \in \Omega \Rightarrow b \in \textsc{Def}(\mathbf{S})) \wedge (\forall c \in A\,.\,(c,a) \in \Gamma \Rightarrow c \in \textsc{Acc}(\mathbf{S}))\}$.

(Corresponding Prop. Program of an AFN)

$P_\Delta = \{a \leftarrow (\bigwedge_{(b,a)\in\Omega} \neg b \ \wedge \ \bigwedge_{(c,a)\in\Gamma} c) \mid a \in A\}$

AFN: $a \Rightarrow b$ means that b is accepted only if a is accepted.

(Classical Definitions)

Given an AFN $\langle A, \Omega, \Gamma \rangle$, and a set of arguments $\mathbf{S} \subseteq A$, then
$Def(\mathbf{S}) = \{a \in A \mid \exists b \in \mathbf{S}\,.\,(b,a) \in \Omega_n\}$, and
$Acc(\mathbf{S}) = \{a \in A \mid \forall b \in A\,.\,(b,a) \in \Omega_n \Rightarrow b \in Def(\mathbf{S})\}$.

(Novel Definitions)

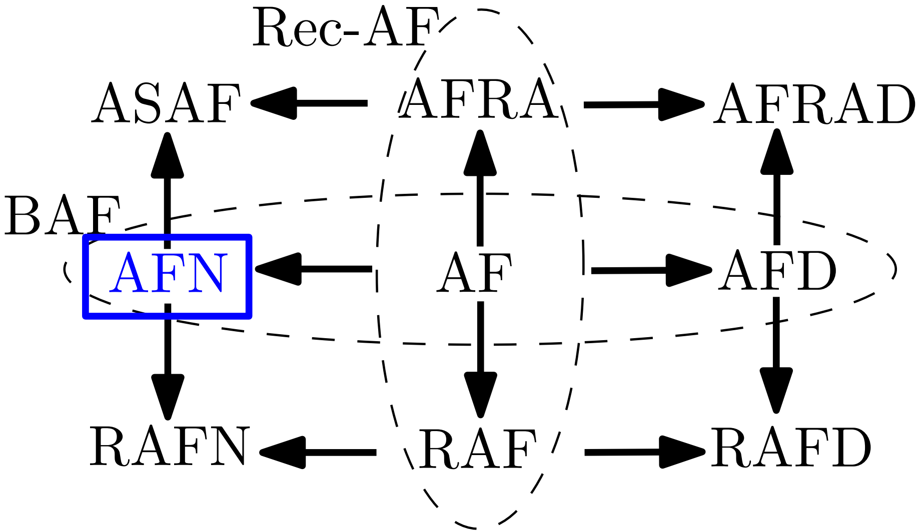
For any AFN $\langle A, \Omega, \Gamma \rangle$ and set of arguments $\mathbf{S} \subseteq A$,
$\textsc{Def}(\mathbf{S}) = \{a \in A \mid (\exists b \in \mathbf{S}\,.\,(b,a) \in \Omega) \vee (\exists c \in \textsc{Def}(\mathbf{S})\,.\,(c,a) \in \Gamma)\}$;
$\textsc{Acc}(\mathbf{S}) = \{a \in A \mid (\forall b \in A\,.\,(b,a) \in \Omega \Rightarrow b \in \textsc{Def}(\mathbf{S})) \wedge$
$(\forall c \in A\,.\,(c,a) \in \Gamma \Rightarrow c \in \textsc{Acc}(\mathbf{S}))\}$.

(Corresponding Prop. Program of an AFN)

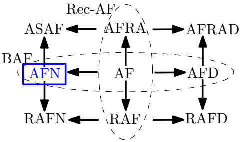
$P_\Delta = \{a \leftarrow (\bigwedge_{(b,a)\in\Omega} \neg b \wedge \bigwedge_{(c,a)\in\Gamma} c) \mid a \in A\}$

Outline

LP for AF-based frameworks: AFN



Argumentation frameworks with Necessities (AFNs)



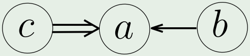
(Corresponding Prop. Program of an AFN)

$P_\Delta = \{a \leftarrow (\bigwedge_{(b,a)\in\Omega} \neg b \ \wedge \ \bigwedge_{(c,a)\in\Gamma} c) \mid a \in A\}$

(Theorem)

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

Example



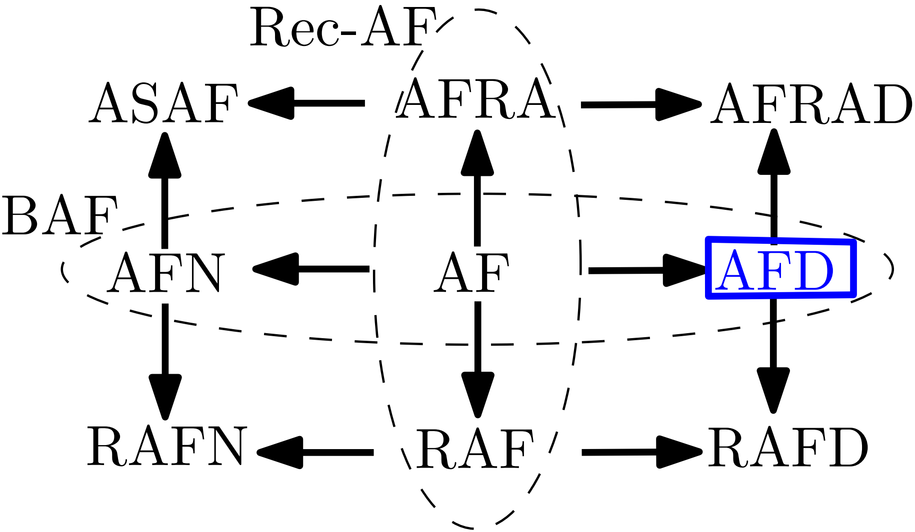
$a \leftarrow \neg b, c$

$b \leftarrow$

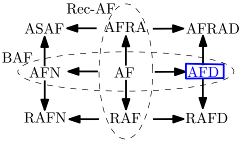
$c \leftarrow$

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

LP for AF-based frameworks: AFD



AF with Deductive Supports (AFDs)



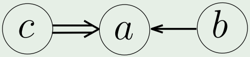
(Corresponding Prop. Program of an AFD)

$P_\Delta = \{a \leftarrow (\bigwedge_{(b,a)\in\Omega} \neg b \ \wedge \ \bigwedge_{(a,c)\in\Gamma} c) \mid a \in A\}$

(Theorem)

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

Example



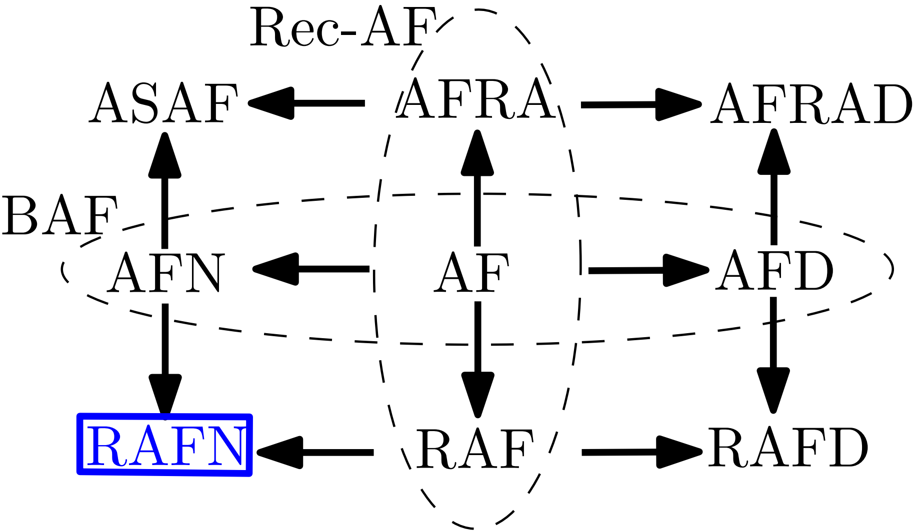
$a \leftarrow \neg b$

$b \leftarrow$

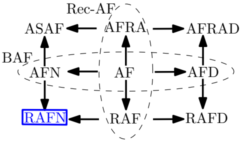
$c \leftarrow a$

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

LP for AF-based frameworks: RAFN



Recursive AF with Necessities (RAFN)



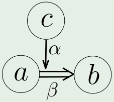
(Corresponding Prop. Program of an RAFN)

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

(Theorem)

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

Example

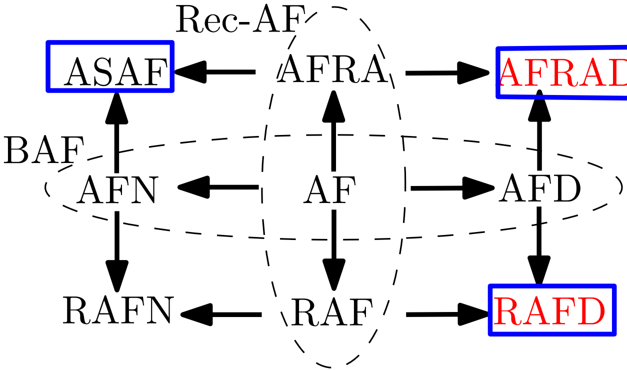


$$a \leftarrow$$
$$b \leftarrow \neg \beta \vee a$$
$$c \leftarrow$$
$$\alpha \leftarrow$$
$$\beta \leftarrow \neg \alpha \vee \neg c$$

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

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?

Outline

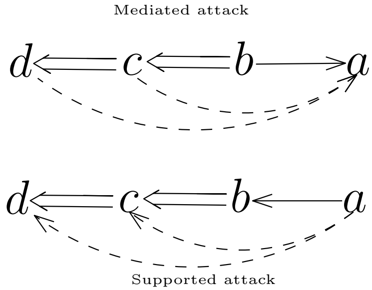
4 Appendix
- Why moving to LP?

Outline

The case of BAFs

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

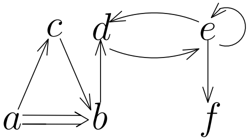
- Directly: one should first look at hidden attacks, and then remove the supports.

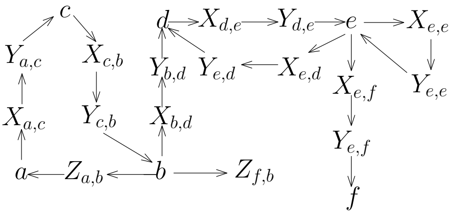


The case of BAFs

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

- Via meta-argumentation: several (fake) meta-arguments and meta-attacks are added.

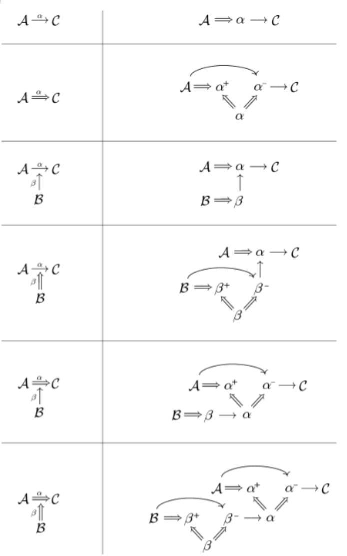




The case of Rec-BAFs

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.

Also when approaching at the direct semantics...



Bipolar Argumentation Frameworks (BAFs)

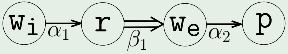
Also includes the notion of support between arguments.
AFN: The necessary interpretation of a support $a \Rightarrow b$ is that b is accepted only if a is accepted. (Dually for **AFDs**).

(BAF)

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*.

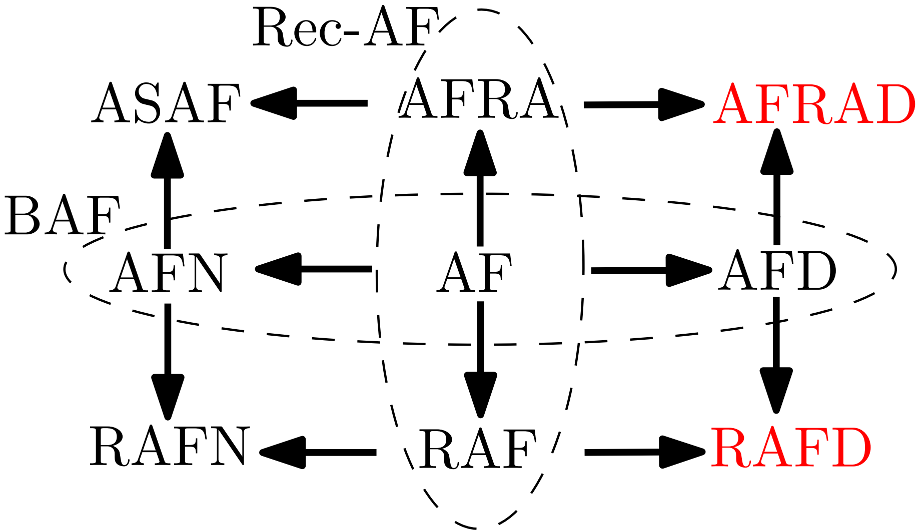
Example

$\mathtt{w_i}$	it is windy
$\mathtt{r}$	it is raining
$\mathtt{w_e}$	the court is wet
$\mathtt{p}$	play tennis



$$\mathcal{CO}(\Delta) = \{\{\mathtt{w_i}, \mathtt{p}\}\}$$

Moving to Rec-BAFs: the corners



Recursive BAFs (Rec-BAFs)

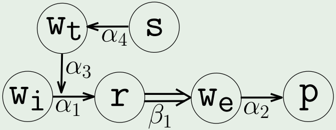
- Combines the concepts of both bipolarity and recursive interactions.
- Two semantics are defined: ASAF & RAFN.

(Rec-BAF)

A *Recursive Bipolar Argumentation Framework (Rec-BAF)* is a tuple $\langle A, \Sigma, \Pi, \mathbf{s}, \mathbf{t} \rangle$, where A is a set of arguments, Σ is a set of attack names, Π is a set of necessary 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).

Example (con't)

$\mathtt{w_t}$	winter
$\mathtt{s}$	it is sunny



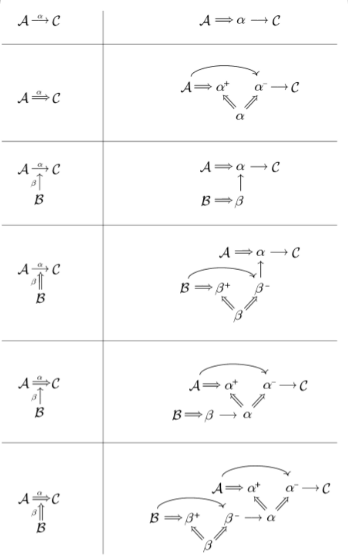
$\mathcal{CO}(\Delta) = \{\{\mathtt{s}, \mathtt{w_i}, \mathtt{p}, \beta_1, \alpha_1, \alpha_2, \alpha_3, \alpha_4\}\}$

Beyond Dung AF

- Sometimes the semantics are a bit difficult to understand, especially when approaching argumentation.
- The semantics for BAFs can be given
  - directly: one should first look at hidden attacks, and then remove the supports.
  - via meta-argumentation: several (fake) meta-arguments and meta-attacks are added.

Beyond Dung AF

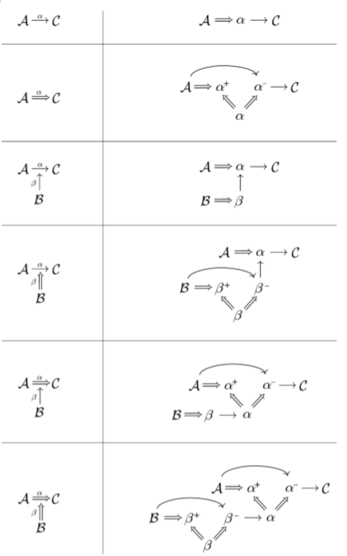
The same holds when moving on Rec-BAFs, but in a more complicated way due to the recursive interactions, which requires several definitions, loosing one of the key aspects of argumentation: simplicity.



Beyond Dung AF

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.

Also when approaching at the direct semantics...



Argumentation frameworks with Necessities (AFNs)

AFN: The necessary interpretation of a support $a \Rightarrow b$ is that b is accepted only if a is accepted.

For any AFN $\langle A, \Omega, \Gamma \rangle$ and set of arguments $\mathbf{S} \subseteq A$,
•$\textsc{Def}(\mathbf{S}) = \{a \in A \mid (\exists b \in \mathbf{S} \,.\, (b, a) \in \Omega) \ \vee \ (\exists c \in \textsc{Def}(\mathbf{S}) \,.\, (c, a) \in \Gamma)\}$;
•$\textsc{Acc}(\mathbf{S}) = \{a \in A \mid (\forall b \in A \,.\, (b, a) \in \Omega \Rightarrow b \in \textsc{Def}(\mathbf{S})) \wedge (\forall c \in A \,.\, (c, a) \in \Gamma \Rightarrow c \in \textsc{Acc}(\mathbf{S}))\}$.

They are useful for defining different extensions (similarly to what done for AFs) as well as allowing to identify the corresponding propositional program.

(Corresponding Prop. Program of an AFN)

Given an AFN $\Delta = \langle A, \Omega, \Gamma \rangle$, then $P_\Delta = \{a \leftarrow (\bigwedge_{(b,a) \in \Omega} \neg b \ \wedge \ \bigwedge_{(c,a) \in \Gamma} c) \mid a \in A\}$ denotes the propositional program derived from Δ

Argumentation frameworks with Necessities (AFNs)

AFN: The necessary interpretation of a support $a \Rightarrow b$ is that b is accepted only if a is accepted.

(Corresponding Prop. Program of an AFN)

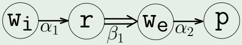
Given an AFN $\Delta = \langle A, \Omega, \Gamma \rangle$, then $P_\Delta = \{a \leftarrow (\bigwedge_{(b,a)\in\Omega} \neg b \ \wedge \ \bigwedge_{(c,a)\in\Gamma} c) \mid a \in A\}$ denotes the propositional program derived from Δ

Example

$$(\mathtt{w_i} \leftarrow) \qquad (\mathtt{r} \leftarrow \neg \mathtt{w_i})$$
$$(\mathtt{w_e} \leftarrow \mathtt{r}) \qquad (\mathtt{p} \leftarrow \neg \mathtt{w_e})\}$$

Clearly

$$\widehat{\mathcal{CO}(\Delta)} = \mathcal{PS}(P_\Delta) = \{\{\mathtt{w_i}, \neg \mathtt{r}, \neg \mathtt{w_e}, \mathtt{p}\}\}$$



AF with Deductive supports (AFDs)

AFD: The deductive interpretation of a support $a \Rightarrow b$ is that b is accepted whenever a is accepted (and a is defeated whenever b is defeated).

For any AFD $\Delta = \langle A, \Omega, \Gamma \rangle$ and set of arguments $\mathbf{S} \subseteq A$,
•$\textsc{Def}(\mathbf{S}) = \{a \in A \mid (\exists\, b \in \mathbf{S}\,.\,(b, a) \in \Omega) \ \vee \ (\exists\, c \in \textsc{Def}(\mathbf{S})\,.\,(a, c) \in \Gamma)\}$;
•$\textsc{Acc}(\mathbf{S}) = \{a \in A \mid (\forall\, b \in A\,.\,(b, a) \in \Omega \Rightarrow b \in \textsc{Def}(\mathbf{S})) \wedge (\forall c \in A\,.(a, c) \in \Gamma \Rightarrow c \in \textsc{Acc}(\mathbf{S}))\}$.

(Corresponding Prop. Program of an AFD)

Given an AFD $\Delta = \langle A, \Omega, \Gamma \rangle$, then $P_\Delta = \{a \leftarrow (\bigwedge_{(b,a)\in\Omega} \neg b \ \wedge \ \bigwedge_{(a,c)\in\Gamma} c) \mid a \in A\}$ denotes the propositional program derived from Δ.

AF with Deductive supports (AFDs)

AFD: The deductive interpretation of a support $a \Rightarrow b$ is that b is accepted whenever a is accepted (and a is defeated whenever b is defeated).

(Corresponding Prop. Program of an AFD)

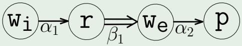
Given an AFD $\Delta = \langle A, \Omega, \Gamma \rangle$, then $P_\Delta = \{a \leftarrow (\bigwedge_{(b,a)\in\Omega} \neg b \ \wedge \ \bigwedge_{(a,c)\in\Gamma} c) \mid a \in A\}$ denotes the propositional program derived from Δ.

Example

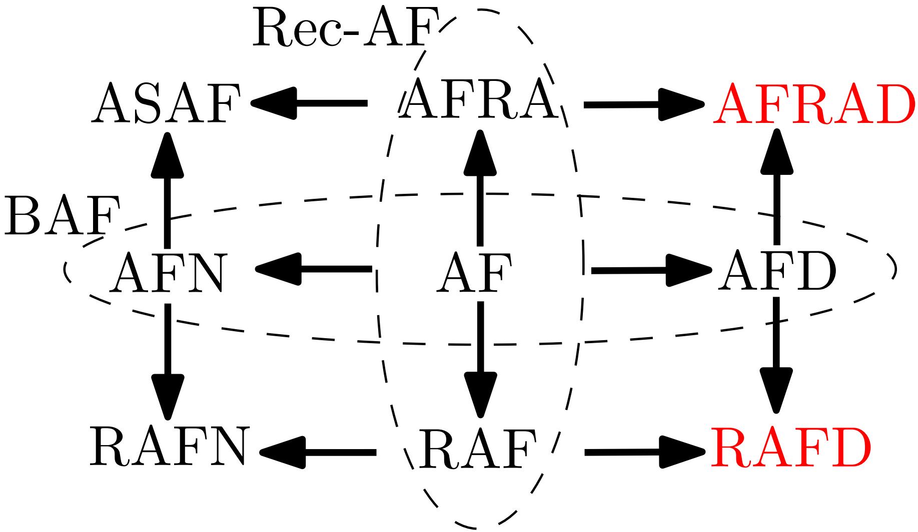
$$(\mathtt{w_i} \leftarrow) \qquad (\mathtt{r} \leftarrow \neg \mathtt{w_i} \wedge \mathtt{w_e})$$
$$(\mathtt{w_e} \leftarrow) \qquad (\mathtt{p} \leftarrow \neg \mathtt{w_e})\}$$

Clearly

$$\widehat{\mathcal{CO}(\Delta)} = \mathcal{PS}(P_\Delta) = \{\{\mathtt{w_i}, \neg\mathtt{r}, \mathtt{w_e}, \neg\mathtt{p}\}\}$$



Moving to Rec-BAFs: the corners

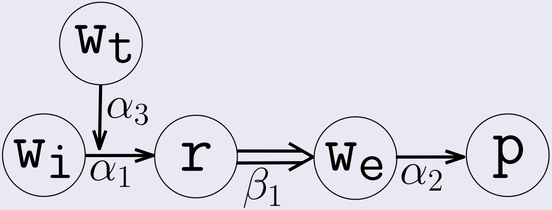


Recursive BAFs (Rec-BAFs)

- Combines the concepts of both bipolarity and recursive interactions.
- Two semantics are defined: *Recursive Argumentation Framework with Necessities (RAFN)* & *Attack-Support Argumentation Framework (ASAF)*.

(Rec-BAF)

A *Recursive Bipolar Argumentation Framework (Rec-BAF)* is a tuple $\langle A, \Sigma, \Pi, \mathbf{s}, \mathbf{t} \rangle$, where A is a set of arguments, Σ is a set of attack names, Π is a set of necessary support names, $\mathbf{s}$ (resp., $\mathbf{t}$) 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).



Recursive BAFs (Rec-BAFs)

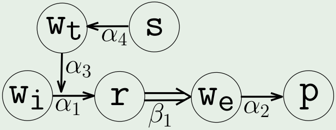
(Corresponding Prop. Program of an RAFN)

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

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

Example

$\{(\mathtt{w_i} \leftarrow), (\mathtt{r} \leftarrow \neg\alpha_1 \vee \neg\mathtt{w_i}),$
$(\mathtt{w_e} \leftarrow \neg\beta_1 \vee \mathtt{r}), (\mathtt{p} \leftarrow \neg\alpha_2 \vee \neg\mathtt{w_e}),$
$(\mathtt{w_t} \leftarrow \neg\alpha_4 \vee \neg\mathtt{s}), (\alpha_1 \leftarrow \neg\alpha_3 \vee \neg\mathtt{w_t}),$
$(\mathtt{s} \leftarrow), (\alpha_2 \leftarrow), (\alpha_3 \leftarrow), (\alpha_4 \leftarrow),$
$(\beta_1 \leftarrow)\}$
PSM: $\{\{\mathtt{s}, \mathtt{w_i}, \neg\mathtt{r}, \neg\mathtt{w_e}, \neg\mathtt{w_t},$
$\mathtt{p}, \beta_1, \alpha_1, \alpha_2, \alpha_3, \alpha_4\}\} = \widehat{\mathcal{CO}}(\Delta)$



Recursive BAFs (Rec-BAFs)

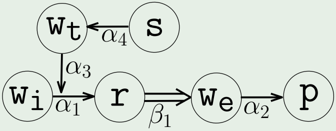
(Corresponding Prop. Program of an ASAF)

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

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

Example

$\{(\mathtt{w_i} \leftarrow), (\mathtt{r} \leftarrow \neg\alpha_1), (\mathtt{w_e} \leftarrow \neg\beta_1 \vee \mathtt{r}),$
$(\mathtt{p} \leftarrow \neg\alpha_2), (\mathtt{w_t} \leftarrow \neg\alpha_4), (\alpha_1 \leftarrow \neg\alpha_3 \wedge \mathtt{w_i}),$
$(\mathtt{s} \leftarrow), (\alpha_2 \leftarrow \mathtt{w_e}), (\alpha_3 \leftarrow \mathtt{w_t}), (\alpha_4 \leftarrow \mathtt{s}), (\beta_1 \leftarrow)\}$
$PSMs(\Delta) = \widehat{\mathcal{CO}(\Delta)} =$
$\{\{\mathtt{s}, \mathtt{w_i}, \neg\mathtt{r}, \neg\mathtt{w_e}, \neg\mathtt{w_t}, \mathtt{p}, \beta_1, \alpha_1, \neg\alpha_2, \neg\alpha_3, \alpha_4\}\}$
differs from RAFN in the status of α_2 and α_3



Recursive BAFs with Deductive Supports

However, no Rec-BAFs under deductive supports were proposed. Then, we study two **new** frameworks both belonging to the Rec-BAF class and both extending AFD by allowing recursive attacks and deductive supports

- *Recursive Argumentation Framework with Deductive supports (RAFD)*, extends RAF:

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

- *Argumentation Framework with Recursive Attacks and Deductive supports (AFRAD)*, extends AFRA.

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

Main Result

(Prop. 1)

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

This result 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. 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.

Thank you!

... any ~~question~~ argument?