Incremental Approaches for Dynamic Abstract Argumentation

Francesco Parisi

fparisi@dimes.unical.it
Department of Informatics, Modeling, Electronics and System Engineering
University of Calabria, Italy

joint work with
Gianvincenzo Alfano and Sergio Greco

3rd Workshop on
Advances in Argumentation in Artificial Intelligence (AI^3)

November 19, 2019

Rende, Italy

Dynamic Argumentation

- Argumentation frameworks are well-known formalisms for modelling and deciding disputes between two or more agents
- Several formalisms, e.g., Abstract Argumentation Frameworks (AFs), Bipolar Argumentation Frameworks (BAFs), or more expressive languages such as Defeasible Logic Programming (DeLP)
- Argumentation frameworks are often dynamic (change over the time) as a consequence of the fact that argumentation is inherently dynamic (change mind/opinion, new available knowledge)

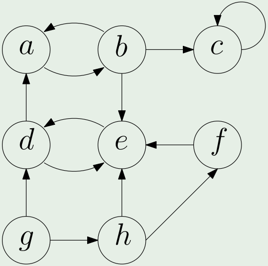
Given an argumentation framework $\mathcal{AF}_t$ at time t, the arguments' status S_t (e.g. accepted/ reject) at time t, and an update u modifying the initial framework $\mathcal{AF}_t$ into $\mathcal{AF}_{t+1}$, **should we recompute the updated arguments' status S_{t+1} from scratch?**

Outline

Argumentation Semantics

- Several semantics have been proposed to identify "reasonable" sets of arguments, called *extensions*

Example (AF $\mathcal{A}_0$)



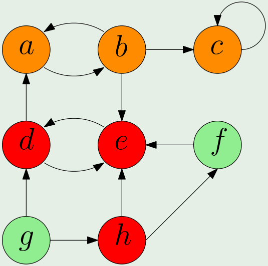
Semantic $\mathcal{S}$	Set of extensions $\mathcal{E}_\mathcal{S}(\mathcal{A}_0)$
complete (co)	$\{\{f,g\},\{a,f,g\},\{b,f,g\}\}$
preferred (pr)	$\{\{a,f,g\},\{b,f,g\}\}$
stable (st)	$\{\{b,f,g\}\}$
grounded (gr)	$\{\{f,g\}\}$

- Argumentation semantics can be also defined in terms of *labelling*
- Function $L : A \to \{\text{IN}, \text{OUT}, \text{UNDECIDED}\}$ assigns a label (accepted, rejected, undecided) to each argument

Argumentation Semantics

- Several semantics have been proposed to identify "reasonable" sets of arguments, called *extensions*

Example (AF $\mathcal{A}_0$)



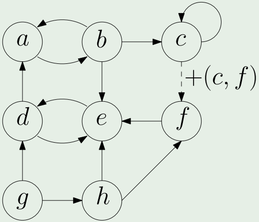
Semantic $\mathcal{S}$	Set of extensions $\mathcal{E}_{\mathcal{S}}(\mathcal{A}_0)$
complete (co)	$\{\{f,g\},\{a,f,g\},\{b,f,g\}\}$
preferred (pr)	$\{\{a,f,g\},\{b,f,g\}\}$
stable (st)	$\{\{b,f,g\}\}$
grounded (gr)	**{ {f,g} }**

- Argumentation semantics can be also defined in terms of *labelling*
- Function $L : A \rightarrow \{\text{IN}, \text{OUT}, \text{UNDECIDED}\}$ assigns a label (accepted, rejected, undecided) to each argument

Dynamic Abstract Argumentation Frameworks

- Most argumentation frameworks are dynamic systems, which are often updated by adding/removing arguments/attacks.
- For each semantics, extensions/labellings change if we update the initial AF by adding/removing arguments/attacks

Example (Updated AF $\mathcal{A} = +(c, f)(\mathcal{A}_0)$)



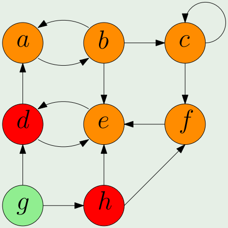
$\mathcal{S}$	$\mathcal{E}_{\mathcal{S}}(\mathcal{A}_0)$	$\mathcal{E}_{\mathcal{S}}(\mathcal{A}))$
co	$\{\{f, g\}, \{a, f, g\}, \{b, f, g\}\}$	?
pr	$\{\{a, f, g\}, \{b, f, g\}\}$	?
st	$\{\{b, f, g\}\}$	?
gr	$\{\{f, g\}\}$	?

- How do we incrementally compute the semantics of updated AFs?

Dynamic Abstract Argumentation Frameworks

- Most argumentation frameworks are dynamic systems, which are often updated by adding/removing arguments/attacks.
- For each semantics, extensions/labellings change if we update the initial AF by adding/removing arguments/attacks

Example (Updated AF $\mathcal{A} = +(c, f)(\mathcal{A}_0)$)



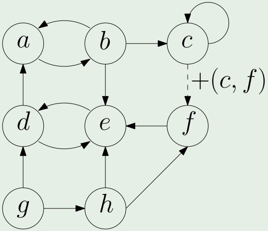
$\mathcal{S}$	$\mathcal{E}_{\mathcal{S}}(\mathcal{A}_0)$	$\mathcal{E}_{\mathcal{S}}(\mathcal{A}))$
co	$\{\{f,g\},\{a,f,g\},\{b,f,g\}\}$	$\{\{g\},\{a,g\},\{b,f,g\}\}$
pr	$\{\{a,f,g\},\{b,f,g\}\}$	$\{\{a,g\},\{b,f,g\}\}$
st	$\{\{b,f,g\}\}$	$\{\{b,f,g\}\ \}$
gr	$\{\{f,g\}\}$	$\{\ \mathbf{\{g\}}\}$

- How do we incrementally compute the semantics of updated AFs?

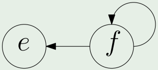
Reduced AF

- For several well-known semantics (i.e., *grounded*, *complete*, *preferred*, *stable*) an extension of the updated AF can be efficiently computed by looking only at a small part of the AF, called the *Reduced* AF, which is "influenced by" the update operation

Example (From the updated AF to the Reduced AF)



$\Longrightarrow$ Reduced AF:



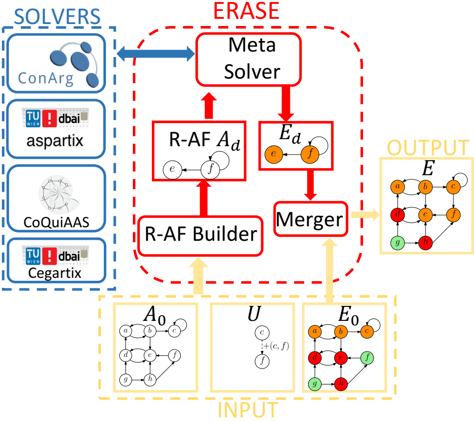
- Once computed an extension for the reduced AF, it can be combined with the initial extension of the given AF to get an extension of the updated AF

Overview of the approach

Given an initial AF $\mathcal{A}_0$, an extension E_0, and an update $u = \pm(a, b)$

Three main steps/modules:

1) Identify a sub-AF $\mathcal{A}_d = \langle A_d, \Sigma_d \rangle$, called *reduced* AF (R-AF) on the basis of the updates in U and additional information provided by the initial extension E_0
2) Compute an $\mathcal{S}$-extension E_d of the reduced AF $\mathcal{A}_d$ by using an external (non-incremental) solver
3) Merge E_d with the portion $(E_0 \setminus A_d)$ of the initial extension that does not change



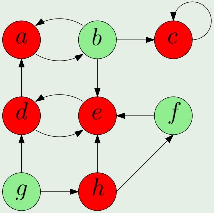
Architecture of ERASE, a system for Efficiently Recomputing Argumentation SEmantics.

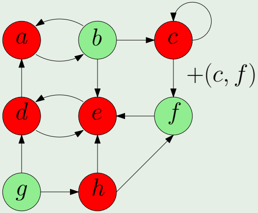
Irrelevant updates (1/2)

- Updates preserving a given initial extension/labelling
- Cases for which E_0 is still an extension of the updated AF after a *positive* update (attack addition)

update $+(a, b)$		$L_0(b)$ IN	$L_0(b)$ UNDECIDED	$L_0(b)$ OUT
$L_0(a)$	IN			co, pr, st, gr
	UNDECIDED		co, gr	co, pr, gr
	OUT	co, **pr**, st	co, gr	co, pr, st, gr

Example (For the update $+(c, f)$ the initial preferred extension $E_0 = \{b, f, g\}$ is preserved, as $L_0(c) =$ OUT and $L_0(f) =$ IN)

initial labelling: 

updated labelling: 

Irrelevant updates (2/2)

- Similar result for *negative* updates
- Cases for which E_0 is still an extension of the updated AF after a *negative* update (attack removal)

update $-(a,b)$		$L_0(b)$ IN	$L_0(b)$ UNDECIDED	$L_0(b)$ OUT
$L_0(a)$	IN	N/A	N/A	
	UNDECIDED	N/A		co, pr, gr
	OUT	co, pr, st, gr	co, pr, gr	co, pr, st, gr

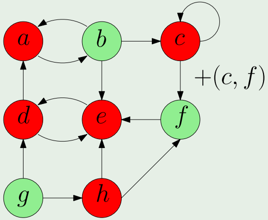
- In these cases we do not need to recompute the semantics of the updated AF: just return the initial extension

Influenced set: Intuition

- $\mathcal{I}(u, \mathcal{A}_0, E_0)$ denotes the *influenced set* of $u = \pm(a, b)$ w.r.t. $\mathcal{A}_0$ and E_0

1) $\mathcal{I}(u, \mathcal{A}_0, E_0) = \emptyset$ if u is irrelevant w.r.t. E_0 and the considered semantics
2) The status of an argument can change only if it is reachable from b: $\mathcal{I}(u, \mathcal{A}_0, E_0) \subseteq Reach_{\mathcal{A}}(b)$
3) If argument z is not reachable from b and $z \in E_0$, then also the status of the arguments attacked by z cannot change: their status remain OUT

Example (Set of arguments influenced by an update operation)



Update $+(c, f)$ is irrelevant w.r.t. the preferred extension $E_0 = \{b, f, g\}$

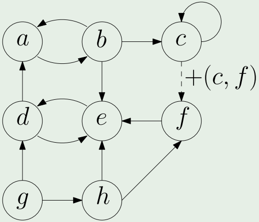
$$\Rightarrow \mathcal{I}(+(c, f), \mathcal{A}_0, \{b, f, g\}) = \emptyset$$

Influenced set: Intuition

- $\mathcal{I}(u, \mathcal{A}_0, E_0)$ denotes the *influenced set* of $u = \pm(a, b)$ w.r.t. $\mathcal{A}_0$ and E_0

1) $\mathcal{I}(u, \mathcal{A}_0, E_0) = \emptyset$ if u is irrelevant w.r.t. E_0 and the considered semantics
2) The status of an argument can change only if it is reachable from b: $\mathcal{I}(u, \mathcal{A}_0, E_0) \subseteq Reach_{\mathcal{A}}(b)$
3) If argument z is not reachable from b and $z \in E_0$, then also the status of the arguments attacked by z cannot change: their status remain OUT

Example (Set of arguments influenced by an update operation)



$\mathcal{I}(+(c, f), \mathcal{A}_0, E_0) \subseteq Reach_{\mathcal{A}}(f) = \{e, d, a, b, c\}$

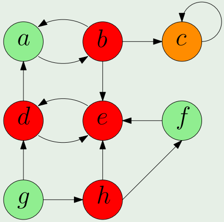
$\Rightarrow g, h \notin \mathcal{I}(+(c, f), \mathcal{A}_0, E_0)$

Influenced set: Intuition

- $\mathcal{I}(u, \mathcal{A}_0, E_0)$ denotes the *influenced set* of $u = \pm(a, b)$ w.r.t. $\mathcal{A}_0$ and E_0

1) $\mathcal{I}(u, \mathcal{A}_0, E_0) = \emptyset$ if u is irrelevant w.r.t. E_0 and the considered semantics
2) The status of an argument can change only if it is reachable from b: $\mathcal{I}(u, \mathcal{A}_0, E_0) \subseteq Reach_{\mathcal{A}}(b)$
3) If argument z is not reachable from b and $z \in E_0$, then also the status of the arguments attacked by z cannot change: their status remain OUT

Example (Set of arguments influenced by an update operation)



$d \notin \mathcal{I}(+(d, f), \mathcal{A}_0, E_0)$ since it is attacked by $g \in E_0$ and g is not reachable from f.

Thus the arguments that can be reached only using d cannot belong to $\mathcal{I}(+(c, f), \mathcal{A}_0, E_0)$.

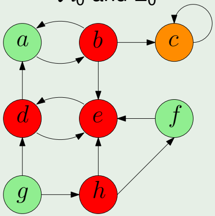
$\Rightarrow$ **The influenced set is** $\mathcal{I}(+(c, f), \mathcal{A}_0, E_0) = \{f, e\}$

Reduced AF

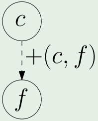
- Given an AF $\mathcal{A}_0$, an extension E_0, and an update $u = \pm(a, b)$, an extension for the updated AF is recomputed for a small part of the updated AF, called *reduced AF* and denoted $\mathcal{R}(u, \mathcal{A}_0, E_0)$
- $\mathcal{R}(u, \mathcal{A}_0, E_0)$ consists of the subgraph of $u(\mathcal{A}_0)$ induced by $\mathcal{I}(u, \mathcal{A}_0, E_0)$
- plus additional nodes/edges representing the "external context":
  1) if there is in $u(\mathcal{A}_0)$ an edge from a node $a \notin \mathcal{I}(u, \mathcal{A}_0, E_0)$ to a node $b \in \mathcal{I}(u, \mathcal{A}_0, E_0)$, we add edge (a, b) if the status of a is IN,
  2) if there is in $u(\mathcal{A}_0)$ an edge from $e \notin \mathcal{I}(u, \mathcal{A}_0, E_0)$ to $c \in \mathcal{I}(u, \mathcal{A}_0, E_0)$ such that e in UNDECIDED, we add edge (c, c) to $\mathcal{R}(u, \mathcal{A}_0, E_0)$

Example (From the influenced set to the Reduced AF)

$\mathcal{A}_0$ and E_0



$u = +(c, f)$



Sub-AF induced by $\mathcal{I}(u, \mathcal{A}_0, E_0)$

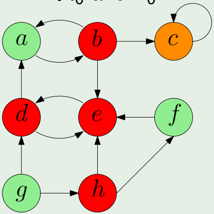
Reduced AF

Reduced AF

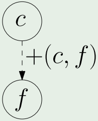
- Given an AF $\mathcal{A}_0$, an extension E_0, and an update $u = \pm(a, b)$, an extension for the updated AF is recomputed for a small part of the updated AF, called *reduced AF* and denoted $\mathcal{R}(u, \mathcal{A}_0, E_0)$
- $\mathcal{R}(u, \mathcal{A}_0, E_0)$ consists of the subgraph of $u(\mathcal{A}_0)$ induced by $\mathcal{I}(u, \mathcal{A}_0, E_0)$
- plus additional nodes/edges representing the "external context":
  1) if there is in $u(\mathcal{A}_0)$ an edge from a node $a \notin \mathcal{I}(u, \mathcal{A}_0, E_0)$ to a node $b \in \mathcal{I}(u, \mathcal{A}_0, E_0)$, we add edge (a, b) if the status of a is IN,
  2) if there is in $u(\mathcal{A}_0)$ an edge from $e \notin \mathcal{I}(u, \mathcal{A}_0, E_0)$ to $c \in \mathcal{I}(u, \mathcal{A}_0, E_0)$ such that e in UNDECIDED, we add edge (c, c) to $\mathcal{R}(u, \mathcal{A}_0, E_0)$

Example (From the influenced set to the Reduced AF)

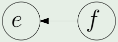
$\mathcal{A}_0$ and E_0



$u = +(c, f)$



Sub-AF induced by $\mathcal{I}(u, \mathcal{A}_0, E_0)$



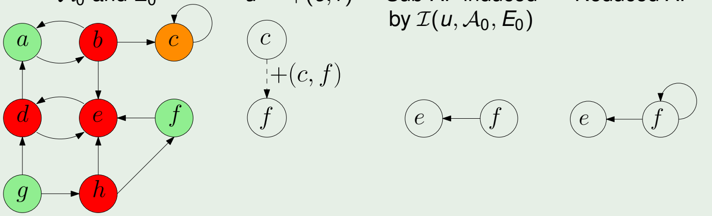
Reduced AF

Reduced AF

- Given an AF $\mathcal{A}_0$, an extension E_0, and an update $u = \pm(a, b)$, an extension for the updated AF is recomputed for a small part of the updated AF, called *reduced AF* and denoted $\mathcal{R}(u, \mathcal{A}_0, E_0)$
- $\mathcal{R}(u, \mathcal{A}_0, E_0)$ consists of the subgraph of $u(\mathcal{A}_0)$ induced by $\mathcal{I}(u, \mathcal{A}_0, E_0)$
- plus additional nodes/edges representing the "external context":
  1) if there is in $u(\mathcal{A}_0)$ an edge from a node $a \notin \mathcal{I}(u, \mathcal{A}_0, E_0)$ to a node $b \in \mathcal{I}(u, \mathcal{A}_0, E_0)$, we add edge (a, b) if the status of a is IN,
  2) if there is in $u(\mathcal{A}_0)$ an edge from $e \notin \mathcal{I}(u, \mathcal{A}_0, E_0)$ to $c \in \mathcal{I}(u, \mathcal{A}_0, E_0)$ such that e in UNDECIDED, we add edge (c, c) to $\mathcal{R}(u, \mathcal{A}_0, E_0)$

Example (From the influenced set to the Reduced AF)

$\mathcal{A}_0$ and E_0 | $u = +(c, f)$ | Sub-AF induced by $\mathcal{I}(u, \mathcal{A}_0, E_0)$ | Reduced AF



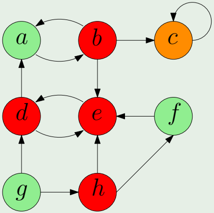
Using extensions of the reduced AF

Theorem (Merging extensions)

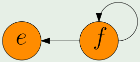
Let $\mathcal{A}_0$ be an AF, and $\mathcal{A} = u(\mathcal{A}_0)$ be the AF resulting from performing update $u = \pm(a, b)$ on $\mathcal{A}_0$. Let $E_0 \in \mathcal{E}_\mathcal{S}(\mathcal{A}_0)$ be an extension for $\mathcal{A}_0$ under a semantics $\mathcal{S} \in \{\texttt{co}, \texttt{pr}, \texttt{st}, \texttt{gr}\}$. Then, if $\mathcal{E}_\mathcal{S}(\mathcal{R}(u, \mathcal{A}_0, E_0))$ is not empty, then there is an extension $E \in \mathcal{E}_\mathcal{S}(\mathcal{A})$ for the updated AF $\mathcal{A}$ such that
$E = (E_0 \setminus \mathcal{I}(u, \mathcal{A}_0, E_0)) \cup E_d$ where
E_d is an S-extension for reduced AF $\mathcal{R}(u, \mathcal{A}_0, E_0)$.

Example (Merging an initial extension with that of the reduced AF)

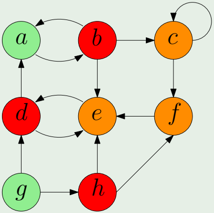
$E_0 \in \mathcal{E}_{\texttt{pr}}(\mathcal{A}_0)$



$E_d \in \mathcal{E}_{\texttt{pr}}(\mathcal{R}(u, \mathcal{A}_0, E_0))$



$E \in \mathcal{E}_{\texttt{pr}}(u(\mathcal{A}_0))$



Incremental Algorithm

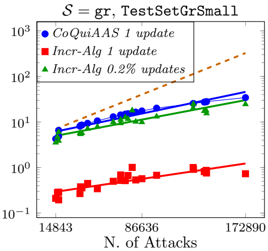
Algorithm Incr-Alg($\mathcal{A}_0, u, \mathcal{S}, E_0$, Solver$_\mathcal{S}$)

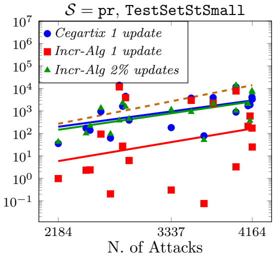
Input: AF $\mathcal{A}_0 = \langle A_0, \Sigma_0 \rangle$, update $u = \pm(a, b)$,
semantics $\mathcal{S} \in \{\texttt{co}, \texttt{pr}, \texttt{st}, \texttt{gr}\}$, extension $E_0 \in \mathcal{E}_\mathcal{S}(\mathcal{A}_0)$,
function Solver$_\mathcal{S}(\mathcal{A})$ returning an $\mathcal{S}$-extension for AF $\mathcal{A}$ if it exists, $\perp$ otherwise;

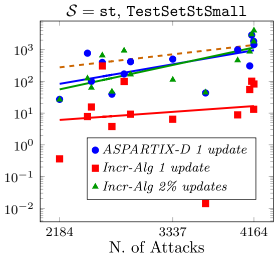
Output: An $\mathcal{S}$-extension $E \in \mathcal{E}_\mathcal{S}(u(\mathcal{A}_0))$ if it exists, $\perp$ otherwise;

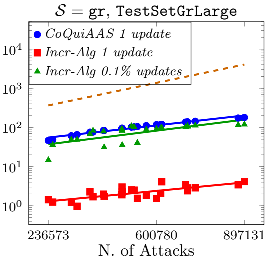
1: $S = \mathcal{I}(u, \mathcal{A}_0, E_0)$; // Compute the influenced set
2: **if** ($S = \emptyset$) **then**
3: **return** E_0; // If the influenced set is empty, return the initial extension E_0
4: $\mathcal{A}_d = \mathcal{R}(u, \mathcal{A}_0, E_0)$; // Otherwise, compute the reduced AF
5: Let $E_d =$ Solver$_\mathcal{S}(\mathcal{A}_d)$; // Compute an extension for the reduced AF using an external solver
6: **if** ($E_d \neq \perp$) **then**
7: **return** $E = (E_0 \setminus S) \cup E_d$; // Merge E_0 with extension E_d of the reduced AF
8: **else**
9: **return** Solver$_\mathcal{S}(u(\mathcal{A}_0))$; // If an extension for the reduced AF doesn't exist (it can happen for stable semantics), compute an extension from scratch

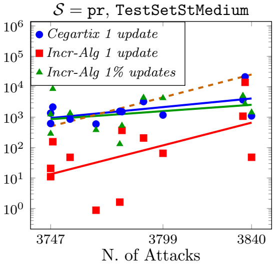
Experimental Results

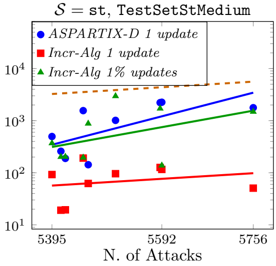












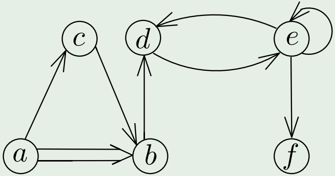
Outline

Bipolar Argumentation Frameworks

- An *abstract Bipolar Argumentation Framework* (*BAF*) is a triple $\langle A, \Sigma, \Pi \rangle$, where
  - $A \subseteq Arg$ is a set of *arguments*,
  - $\Sigma \subseteq A \times A$ is a set of *attacks*,
  - $\Pi \subseteq A \times A$ is set of **supports** ($\Sigma \cap \Pi = \emptyset$)
- Dung's AF is a BAF of the form $\langle A, \Sigma, \emptyset \rangle$.

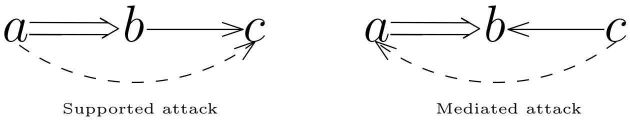
Example (BAF)

$A = \{a, b, c, d, e, f\}$
$\Sigma = \{(a, c),\ (c, b),\ (b, d),\ (d, e)$
$\quad (e, d),\ (e, e),\ (e, f)\}$
$\Pi = \{(a, b)\}$



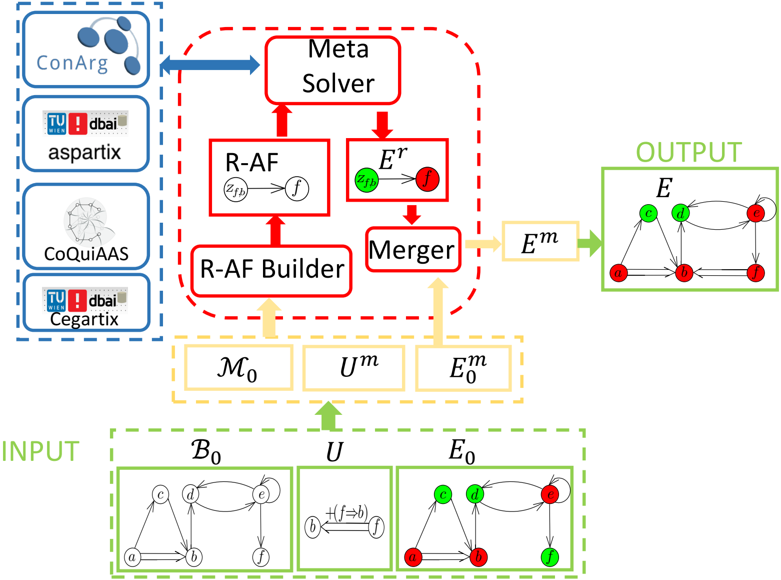
Implicit Attacks

The coexistence of the support and attack relations in BAFs entails that new kinds of "implicit" attacks should be considered



- A set $S \subseteq A$ *set-attacks* an argument $b \in A$ iff there exists a supported or mediated attack for b by an argument $a \in S$
- $S \subseteq A$ *defends* an argument $a \in A$ iff for each $b \in A$ such that $\{b\}$ set-attacks a, it is the case that S set-attacks b
- BAFs semantics (e.g. stable and preferred) can be defined as in the Dung's framework using the above notions of set-attack and defence

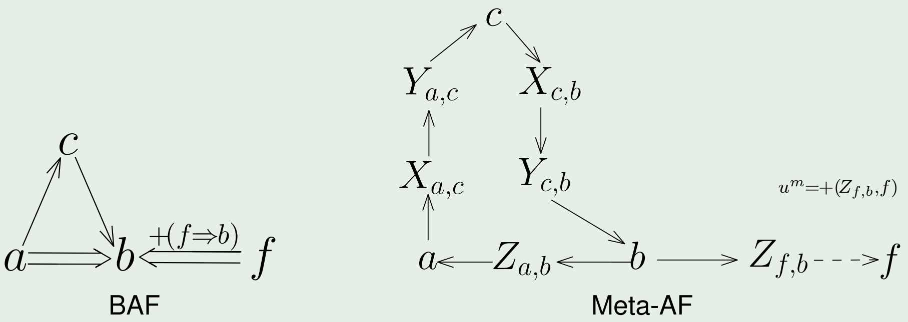
Overview of the approach



The Meta-Argumentation Framework

- The definition of meta-AF builds on that proposed in [Boella et al., 2010]
- consider additional (meta)arguments (e.g., $Z_{f,b}$) and attacks (e.g., $(b, Z_{f,b})$) that will allow us to *simulate updates* to be performed on BAF $\mathcal{B}_0$ by *means of updates performed on the corresponding the meta-AF* $\mathcal{M}_0$.

Example (Meta AF $\mathcal{M}_0$ for the BAF $\mathcal{B}_0$ w.r.t. the update $u = +(f \Rightarrow b)$)



Incremental Algorithm

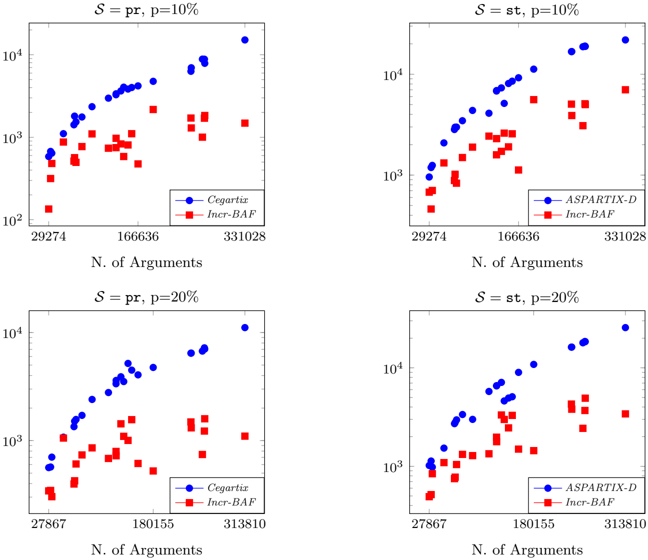
Algorithm Incr-BAF($\mathcal{B}_0, u, E_0, \mathcal{S}$, Solver$_\mathcal{S}$)

Input: BAF $\mathcal{B}_0 = \langle A_0, \Sigma_0\ \Pi_0 \rangle$,
update u of the form $u = \pm(a \Rightarrow b)$ or $u = \pm(a \rightarrow b)$,
an initial $\mathcal{S}$-extension E_0,
semantics $\mathcal{S} \in \{\texttt{pr}, \texttt{st}\}$,
function Solver$_\mathcal{S}(\mathcal{A})$ returning an $\mathcal{S}$-extension for AF $\mathcal{A}$ if it exists, $\perp$ otherwise

Output: An $\mathcal{S}$-extension E for $u(\mathcal{B}_0)$ if it exists, $\perp$ otherwise;

```
if checkProp(B_0, u, E_0, S) then
    return E_0 // Check if the initial extension is preserved (the update is irrelevant)
Let M_0 = ⟨A^m, Σ^m⟩ be the the meta-AF for B_0 w.r.t. u  // Compute the meta AF
Let u^m be the update for M_0 corresponding to u // Translate u into u^m
Let E_0^m be the initial S-extension for M_0 corresponding to E_0 // Convert the initial extension for the BAF into an extension for the meta AF
Let E^m = Incr-Alg(M_0, u^m, S, E_0^m, Solver_S) // Compute an S-extension for the meta AF
if (E^m ≠ ⊥) then
    return  E = (E^m ∩ A_0);
else
    return  ⊥; // An extension for the the meta AF could not exists (e.g., stable semantics)
```

Experimental Results (p is the percentage of supports)

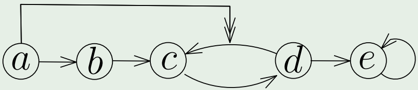


Extended Abstract Argumentation Frameworks

- An *Extended Argumentation Framework* (*EAF*) is a triple $\langle A, \Sigma, \Delta \rangle$, where
  - $A \subseteq Arg$ is a set of arguments
  - $\Sigma \subseteq A \times A$ is a set of attacks
  - Δ is a binary relation over $A \times \Sigma$: **second-order attacks**
- A Dung's AF is an EAF of the form $\langle A, \Sigma, \emptyset \rangle$

Example (EAF)

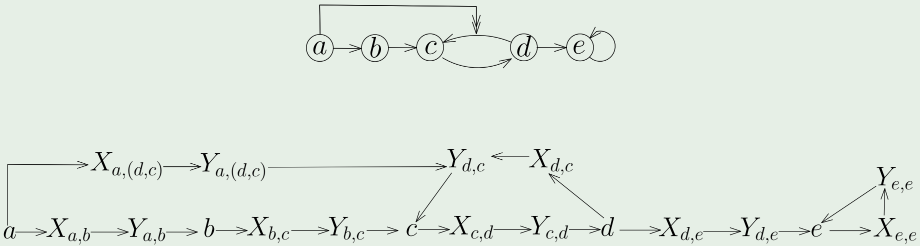
$A = \{a, b, c, d, e\}$
$\Sigma = \{(a, b),\ (b, c),\ (c, d),\ (d, c),\ (d, e),\ (e, e)\}$
$\Delta = \{(a, (d, c))\}$



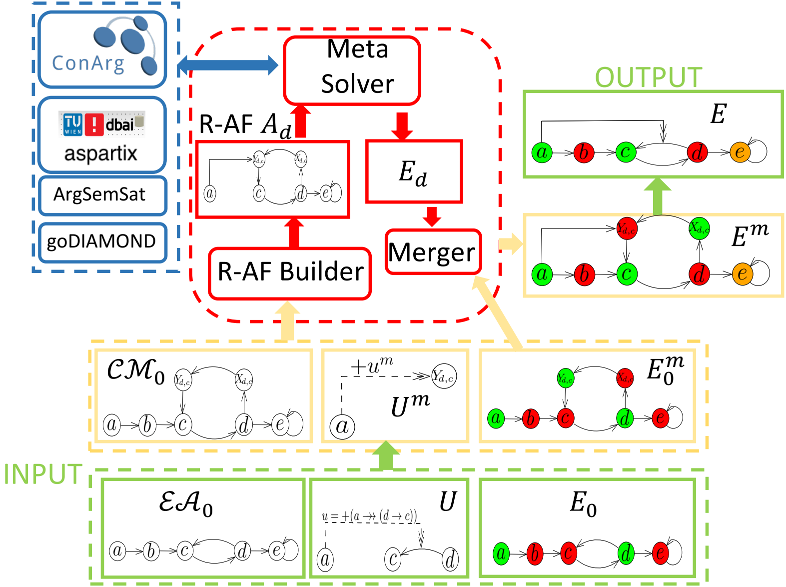
Semantics for Extended Abstract Argumentation

The semantics of EAFs can be given in terms of meta-argumentation frameworks (i.e., Dung's AFs) where additional (meta-)arguments and attacks are considered to model second-order attacks

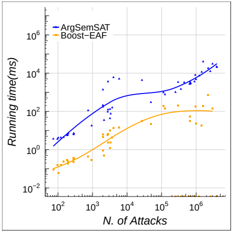
Example (EAF and corresponding Meta-AF)



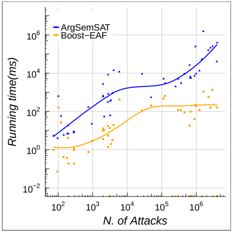
Overview of the approach



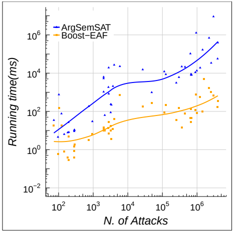
Results (s is the percentage of second-order attacks)



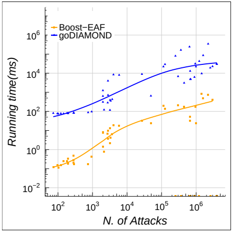
(a) $\mathcal{S} = \texttt{pr}$, $s = 0\%$.



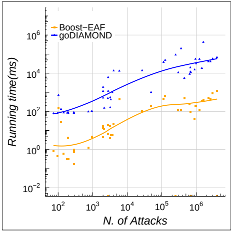
(b) $\mathcal{S} = \texttt{pr}$, $s = 10\%$.



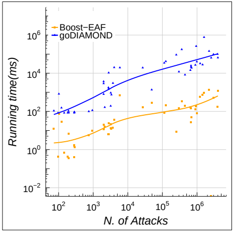
(c) $\mathcal{S} = \texttt{pr}$, $s = 20\%$.



(d) $\mathcal{S} = \texttt{st}$, $s = 0\%$.



(e) $\mathcal{S} = \texttt{st}$, $s = 10\%$.



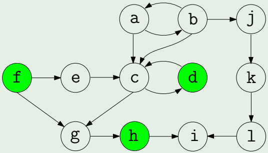
(f) $\mathcal{S} = \texttt{st}$, $s = 20\%$.

Outline

Skeptically Preferred Acceptance

- An argument *g* (goal) is skeptically preferred accepted w.r.t. $\mathcal{A}$ (denoted as $SA_{\mathcal{A}}(\mathtt{g}) = true$) iff it appears in every $\mathtt{pr}$-extension of $\mathcal{A}$.

Example (AF $\mathcal{A}_0$)



Semantic $\mathcal{S}$	Set of extensions of $\mathcal{A}_0$
preferred ($\mathtt{pr}$)	$\{\{\mathtt{a,d,f,h,j,l}\},\{\mathtt{b,d,f,h,k}\}\}$
ideal ($\mathtt{id}$)	$\{\{\mathtt{d,f,h}\}\}$

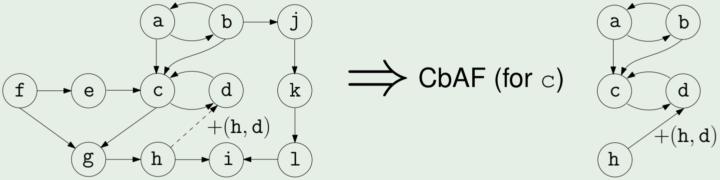
- $SA_{\mathcal{A}}(\mathtt{d}) = SA_{\mathcal{A}}(\mathtt{f}) = SA_{\mathcal{A}}(\mathtt{h}) = true$

- A preferred extension of an AF $\mathcal{A}$ is a maximal admissible set of $\mathcal{A}$
- The ideal extension of $\mathcal{A}$ is the biggest admissible set of $\mathcal{A}$ which is contained in every preferred extension of $\mathcal{A}$
- If an argument is in the ideal extension then it is skeptically (preferred) accepted, but the converse does not hold

Updates and Context-based AF (CbAF)

- The skeptical preferred acceptance of goal argument w.r.t an updated AF can be efficiently computed by looking only at a small part of the AF, called the *Context-based* AF

Example (From the updated AF to the CbAF)

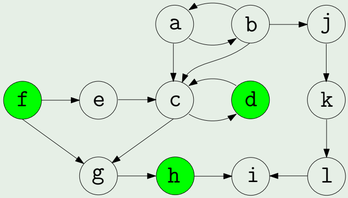


- The Context-based AF depends on the update, the initial *ideal* extension, and the *goal argument*
- It is useful to incrementally maintain the ideal extension

Supporting set: Intuition

- $Sup(u, \mathcal{A}, E, g)$ is the set of arguments whose status may change after performing update u and s.t. they may imply a change of the status of the goal argument g
- $Sup(u, \mathcal{A}, E, g)$ for update $u = \pm(a, b)$, goal g, and ideal extension E consists of the arguments that
  - (*i*) can be reached from b without using any argument attacked by $z \in E$ that is not reachable from b, and
  - (*ii*) allow to reach the goal g by using only the arguments selected as above (if not possible it is empty)

Example (For update $u = +(\mathrm{h}, \mathrm{d})$)

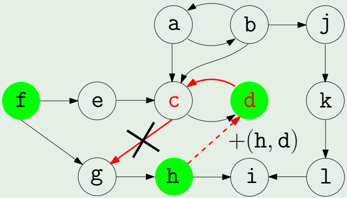


Don't use g and e as they are attacked by some argument in the ideal extensions (i.e., f) which is not reachable from the target node d of the update $+(\mathrm{h}, \mathrm{d})$

Supporting set: Intuition

- *Sup*$(u, \mathcal{A}, E, g)$ is the set of arguments whose status may change after performing update u and s.t. they may imply a change of the status of the goal argument g
- *Sup*$(u, \mathcal{A}, E, g)$ for update $u = \pm(a, b)$, goal g, and ideal extension E consists of the arguments that
  - (*i*) can be reached from b without using any argument attacked by $z \in E$ that is not reachable from b, and
  - (*ii*) allow to reach the goal g by using only the arguments selected as above (if not possible it is empty)

Example (For update $u = +(\mathrm{h}, \mathrm{d})$)



For the goal c, the supporting set is: *Sup*$(u, \mathcal{A}_0, E_{id}, \mathrm{c}) = \{\mathrm{c}, \mathrm{d}\}$

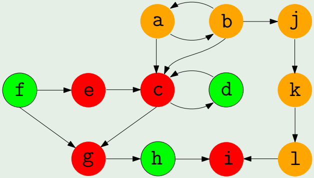
For the goal h, the supporting set is empty: the goal cannot be reached from d without using g

Context-based AF for skeptical acceptance

- A restriction of the AF to incrementally compute the status of the goal
- Similar to the reduced AF for $\mathtt{gr}$, $\mathtt{co}$, $\mathtt{st}$, and $\mathtt{pr}$
- But here we use as input the ideal extension and $Sup(u, \mathcal{A}, E, g)$
- In addition to the arguments of the reduced AF, we need to consider the arguments and attacks of the updated AF such that:
  (*a*) they occur in a path ending in $Sup(u, \mathcal{A}, E, g)$, and
  (*b*) the arguments of the path outside the supporting set are undecided

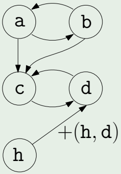
Example (From the updated AF to the CbAF)

$u = +(\mathrm{h}, \mathrm{d})$



$Sup(u, \mathcal{A}_0, E_{id}, \mathrm{c}) = \{\mathrm{c}, \mathrm{d}\}$

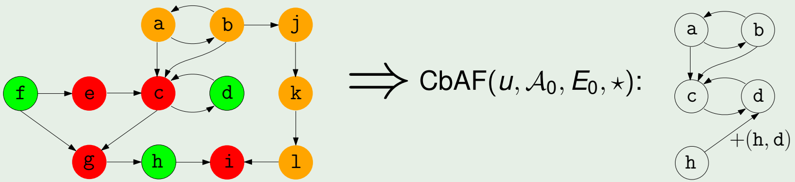
$\Longrightarrow$ CbAF$(u, \mathcal{A}_0, E_0, \mathrm{c})$:



CbAF for the incremental computation of the ideal ext.

- Same as the Context-based AF for the skeptical acceptance
- But build from a supporting set where the goal argument is not specified (it can be any argument): $Sup(u, \mathcal{A}_0, E_0, \star)$
- $Sup(u, \mathcal{A}_0, E_0, \star)$ is defined as $Sup(u, \mathcal{A}_0, E_0, g)$ but without considering the conditions that the arguments selected must allow to reach the goal g

Example (From the updated AF to the CbAF)



- in our example, since $Sup(u, \mathcal{A}_0, E_0, \star) = Sup(u, \mathcal{A}_0, E_0, \mathtt{c})$ then $\text{CbAF}(u, \mathcal{A}_0, E_0, \star) = \text{CbAF}(u, \mathcal{A}_0, E_0, \mathtt{c})$
- in general $\text{CbAF}(u, \mathcal{A}_0, E_0, \star)$ includes $\text{CbAF}(u, \mathcal{A}_0, E_0, g)$ for each g

Incremental Algorithm

Algorithm SPA($\mathcal{A}_0, g, SA_{\mathcal{A}_0}(g), u, E_0$)

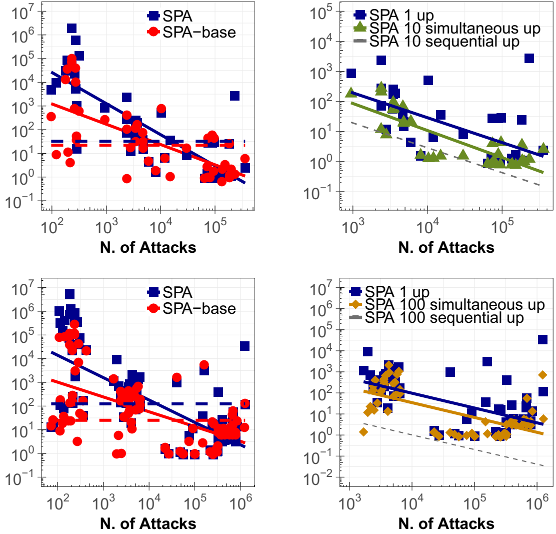
Input: AF $\mathcal{A}_0 = \langle A_0, \Sigma_0 \rangle$, argument $g \in A_0$,
skeptical acceptance $SA_{\mathcal{A}_0}(g)$ of g w.r.t. $\mathcal{A}_0$,
update $u = \pm(a, b)$,
ideal extension E_0 of $\mathcal{A}_0$;

Output: skeptical acceptance $SA_{u(\mathcal{A}_0)}(g)$ of g w.r.t. $u(\mathcal{A}_0)$,
ideal extension E of $u(\mathcal{A}_0)$;

Let $S_\star = Sup(u, \mathcal{A}_0, E_0, \star)$ // supporting set for computing the updated ideal extension
Let $\mathcal{A}_{id} = \text{CbAF}(u, \mathcal{A}_0, E_0, \star)$// CbAF for computing the updated ideal extension
Let $E = (E_0 \setminus S_\star) \cup \text{ID-Solver}(\mathcal{A}_{id})$// Incremental computation of the ideal extension
if $g \in E$ **then**
    return $\langle true, E \rangle$// if the goal is in the ideal extension, then it is skeptical accepted
if $g \in E^+$ **then**
    return $\langle false, E \rangle$// g is attacked by the ideal extension, thus it is not skeptically accepted
Let $S_g = Sup(u, \mathcal{A}_0, E_0, g)$ // supporting set for the skeptical acceptance of g
if S_g is empty **then**
    return $\langle SA_{\mathcal{A}_0}(g), E \rangle$ // skeptical acceptance preserved
Let $\mathcal{A}_{sa} = \text{CbAF}(u, \mathcal{A}_0, E_0, g)$// CbAF for skeptical acceptance of g
return $\langle \text{SA-Solver}(\mathcal{A}_{sa}, g), E \rangle$// If the supporting set is not empty, it suffices to compute the skeptical acceptance only on the CbAF

Experimental Results: improvement of SPA and SPA-base over the computation from scratch

1. Incremental algorithm SPA, where ID-Solver is pyglaf [Alviano, 2017] and SA-Solver is ArgSemSAT [Cerutti et al., 2014], the solver that won the ICCMA'17 DS-pr track;
2. SPA-base, a version of SPA where the ideal extension is not used;
3. ArgSemSAT for the computation from scratch.



Outline

Conclusions and Future Work

- We discussed a general incremental approach based on identifying a tighter portion of the updated framework (e.g. AF, BAF) to be examined for recomputing the status of arguments
- The approach uses both the initial arguments' status, the update, as well as the structure of the given framework
- The incremental approach enables any non-incremental algorithm to be used as an incremental one for boosting computation
- It can be applied to the incremental computation of skeptical acceptance
- Current and Future work:
  - (i) applying the technique to more general argumentation frameworks (e.g., ASAF, DeLP) and other semantics (e.g., semi-stable)
  - (ii) enumerating all the extensions and deciding credulous/sceptical acceptance for other semantics
  - (iii) devising heuristics to take advantages of different algorithms (incremental or not, depending on the input framework)

Thank you for your attention!

... question?

Selected References

Gianvincenzo Alfano, Sergio Greco, Francesco Parisi.
Efficient Computation of Extensions for Dynamic Abstract Argumentation Frameworks: An Incremental Approach.
In *IJCAI*, pages 49–55, 2017.

Sergio Greco, Francesco Parisi.
Incremental Computation of Deterministic Extensions for Dynamic Argumentation Frameworks.
In *IJCAI*, pages 288–304, 2016.

Sergio Greco, Francesco Parisi.
Efficient Computation of Deterministic Extensions for Dynamic Abstract Argumentation Frameworks.
In *ECAI*, pages 1668–1669, 2016.

Gianvincenzo Alfano, Sergio Greco, Francesco Parisi.
A meta-argumentation approach for the efficient computation of stable and preferred extensions in dynamic bipolar argumentation frameworks.
Intelligenza Artificiale, 2018.

Gianvincenzo Alfano, Sergio Greco, Francesco Parisi.
Computing Stable and Preferred Extensions of Dynamic Bipolar Argumentation Frameworks.
*AI*IA Workshop on Advances in Argumentation in Artificial Intelligence (AI3)*, 2017.

Gianvincenzo Alfano, Sergio Greco, Francesco Parisi.
Computing Extensions of Dynamic Abstract Argumentation Frameworks with Second-Order Attacks.
IDEAS, 2018.

Gianvincenzo Alfano, Sergio Greco, Francesco Parisi.
Gianvincenzo Alfano, Sergio Greco, Francesco Parisi: An Efficient Algorithm for Skeptical Preferred Acceptance in Dynamic Argumentation Frameworks
In *IJCAI*, 2019.