Dynamics in Abstract Argumentation Frameworks with Recursive Attack and Support Relations

Gianvincenzo Alfano[1], Andrea Cohen[2], Sebastian Gottifredi[2], Sergio Greco[1], Francesco Parisi[1], and Guillermo Simari[2]

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

[2] {ac, sg, grs}@cs.uns.edu.ar
DCIC, Universidad Nacional del Sur, ICIC (CONICET-UNS), Argentina

24th European Conference on Artificial Intelligence

August 29- September 5, 2020

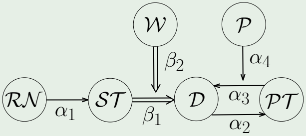
(Virtual) Santiago de Compostela, Portugal

Attack-Support Argumentation Framework (ASAF)

A general way for represent arguments and relationships between them, allowing to represent dialogues, make decisions, and handle inconsistency and uncertainty.

Example (a simple ASAF)

Suppose John is planning to spend his winter holidays in Bariloche and has to decide whether to rent a car to drive during his stay ($\mathcal{D}$) or make use of public transportation ($\mathcal{PT}$). In general, John has a preference towards driving over using public transport ($\mathcal{P}$). John needs to put a snow traction device on the car ($\mathcal{ST}$) only required during this season ($\mathcal{W}$(inter)). In addition, rental car services ran out of such devices ($\mathcal{RN}$). In this context, John will end up deciding to use public transportation.



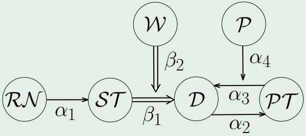
Semantics for ASAF: "reasonable" sets of arguments and relationships, called *extensions*. We focused on complete, preferred, stable and grounded.

Attack-Support Argumentation Framework (ASAF)

A general way for represent arguments and relationships between them, allowing to represent dialogues, make decisions, and handle inconsistency and uncertainty.

Example (a simple ASAF)

Suppose John is planning to spend his winter holidays in Bariloche and has to decide whether to rent a car to drive during his stay ($\mathcal{D}$) or make use of public transportation ($\mathcal{PT}$). In general, John has a preference towards driving over using public transport ($\mathcal{P}$). John needs to put a snow traction device on the car ($\mathcal{ST}$) only required during this season ($\mathcal{W}$(inter)). In addition, rental car services ran out of such devices ($\mathcal{RN}$). In this context, John will end up deciding to use public transportation.



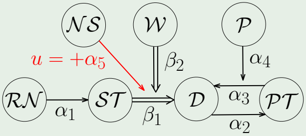
Semantics for ASAF: "reasonable" sets of arguments and relationships, called *extensions*. We focused on complete, preferred, stable and grounded.

Dynamic ASAFs

Typically an ASAF represents a temporary situation, and new arguments, attacks and supports can be added/removed to take into account new available knowledge.

Example (a simple ASAF)

Suppose it has not snowed for the last two months and will not snow during John's stay ($\mathcal{NS}$) **update +($\mathcal{NS}$, β_1)**



Should we recompute the extensions from scratch?

Contributions

1) We identify early-termination conditions.

2) We characterize an ASAF in terms of an AF, and formally show that this AF yields equivalent extensions to those of the ASAF.

3) We define an incremental algorithm for computing extensions of dynamic ASAFs by leveraging on the incremental technique proposed in [Alfano et al, 2017].

4) Experimental analysis comparing with fastest solvers from ICCMA 2017 and ICCMA 2019.

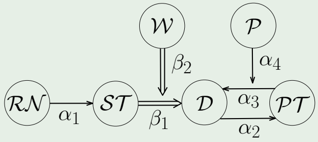
Outline

Updates

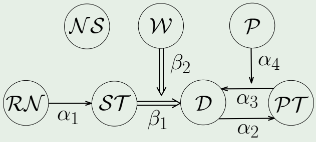
- An *update* u for an ASAF Δ allow us to change Δ into an ASAF Δ' by adding or removing an argument, an attack, or a support.
- If E_0 is an extension for Δ and Δ' is obtained by adding (resp. removing) the set S of isolated arguments, then $E = E_0 \cup S$ (resp. $E = E_0 \setminus S$)
- We focus on the addition (+) (resp., deletion (−)) of of an attack or a support not present (resp., present) in a given ASAF.
- $u(\Delta)$ denotes the application of update u to Δ.

Example (Extensions after adding the isolated argument $\mathcal{NS}$)

preferred extension:
$\{\mathcal{RN}, \alpha_1, \beta_1, \mathcal{W}, \beta_2, \mathcal{PT}, \mathcal{P}, \alpha_4\}$



preferred extension:
$\{\mathcal{RN}, \alpha_1, \beta_1, \mathcal{W}, \beta_2, \mathcal{PT}, \mathcal{P}, \alpha_4, \mathcal{NS}\}$

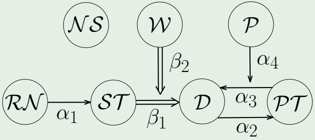


Updates

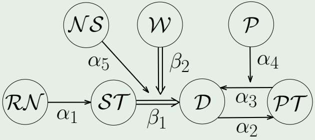
- An *update* u for an ASAF Δ allow us to change Δ into an ASAF Δ' by adding or removing an argument, an attack, or a support.
- If E_0 is an extension for Δ and Δ' is obtained by adding (resp. removing) the set S of isolated arguments, then $E = E_0 \cup S$ (resp. $E = E_0 \setminus S$)
- We focus on the addition $(+)$ (resp., deletion $(-)$) of of an attack or a support not present (resp., present) in a given ASAF.
- $u(\Delta)$ denotes the application of update u to Δ.

Example (Extensions after adding the attack $+(\mathcal{NS}, \beta_1)$)

preferred extension:
$\{\mathcal{RN}, \alpha_1, \beta_1, \mathcal{W}, \beta_2, \mathcal{PT}, \mathcal{P}, \alpha_4, \mathcal{NS}\}$



preferred extension:
$\{\mathcal{RN}, \alpha_1, \alpha_2, \alpha_5, \mathcal{NS}, \mathcal{W}, \beta_2, \mathcal{D}, \mathcal{P}, \alpha_4\}$



Overview of the Approach

(STEP 1) We check if an update is irrelevant: an extension of the updated ASAF can be easily obtained, without requiring its overall computation;

(STEP 2) For relevant updates, we compute the **AF for an ASAF**, corresponding to a Dung's AF encoding every argument/attack/support of the ASAF. We proved that there exists a one-to-one correspondence between the extensions of an ASAF and the extensions of its AF;

(STEP 3) We redefine an update u over an ASAF in terms of a(n equivalent) set of updates u' over its AF, and call an external incremental solver [Alfano et al, 2017] for computing an updated extension for the updated AF for the ASAF.

(STEP 4) Exploiting the one-to-one correspondence of extensions, we obtain an updated extension for the updated ASAF.

Overview of the Approach

(STEP 1) We check if an update is irrelevant: an extension of the updated ASAF can be easily obtained, without requiring its overall computation;

(STEP 2) For relevant updates, we compute the **AF for an ASAF**, corresponding to a Dung's AF encoding every argument/attack/support of the ASAF. We proved that there exists a one-to-one correspondence between the extensions of an ASAF and the extensions of its AF;

(STEP 3) We redefine an update u over an ASAF in terms of a(n equivalent) set of updates u' over its AF, and call an external incremental solver [Alfano et al, 2017] for computing an updated extension for the updated AF for the ASAF.

(STEP 4) Exploiting the one-to-one correspondence of extensions, we obtain an updated extension for the updated ASAF.

Overview of the Approach

(STEP 1) We check if an update is irrelevant: an extension of the updated ASAF can be easily obtained, without requiring its overall computation;

(STEP 2) For relevant updates, we compute the **AF for an ASAF**, corresponding to a Dung's AF encoding every argument/attack/support of the ASAF. We proved that there exists a one-to-one correspondence between the extensions of an ASAF and the extensions of its AF;

(STEP 3) We redefine an update u over an ASAF in terms of a(n equivalent) set of updates u' over its AF, and call an external incremental solver [Alfano et al, 2017] for computing an updated extension for the updated AF for the ASAF.

(STEP 4) Exploiting the one-to-one correspondence of extensions, we obtain an updated extension for the updated ASAF.

Overview of the Approach

(STEP 1) We check if an update is irrelevant: an extension of the updated ASAF can be easily obtained, without requiring its overall computation;

(STEP 2) For relevant updates, we compute the **AF for an ASAF**, corresponding to a Dung's AF encoding every argument/attack/support of the ASAF. We proved that there exists a one-to-one correspondence between the extensions of an ASAF and the extensions of its AF;

(STEP 3) We redefine an update u over an ASAF in terms of a(n equivalent) set of updates u' over its AF, and call an external incremental solver [Alfano et al, 2017] for computing an updated extension for the updated AF for the ASAF.

(STEP 4) Exploiting the one-to-one correspondence of extensions, we obtain an updated extension for the updated ASAF.

(STEP 1) Irrelevant Updates

Cases in which the update $u = +\alpha$, with $\alpha \in (\mathbb{R}_{\mathbb{U}} \backslash \mathbb{R})$ is irrelevant. If $L_0(\text{src}(\alpha)) = \text{IN}$, then $E_0 \cup \{\alpha\} \in \mathcal{E}_{\mathcal{S}}(u(\Delta))$; otherwise, $E_0 \in \mathcal{E}_{\mathcal{S}}(u(\Delta))$.

Update $u = +\alpha$, with $\alpha \in (\mathbb{R}_{\mathbb{U}} \backslash \mathbb{R})$		$L_0(\text{trg}(\alpha))$ IN	$L_0(\text{trg}(\alpha))$ UN	$L_0(\text{trg}(\alpha))$ OUT
$L_0(\text{src}(\alpha))$	IN			co,pr,st,gr
	UN		co,gr	co,pr,gr
	OUT	co,pr,st	co,pr,gr	co,pr,st,gr

(Theorem 1)

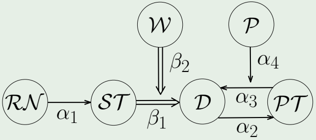
Let $\Delta = \langle \mathbb{A}, \mathbb{R}, \mathbb{S} \rangle$ be an ASAF, $u = \pm X$ an update for Δ, and $u(\Delta)$ the updated ASAF. Also, let E_0 be an $\mathcal{S}$-extension of Δ and L_0 the labelling corresponding to E_0, where $\mathcal{S} \in \{\text{co}, \text{pr}, \text{st}, \text{gr}\}$. Then, for each type of update considered in Tables $1 - 4$, a semantics $\mathcal{S}$ occurs in the cell $\langle L_0(\text{src}(X)), L_0(\text{trg}(X)) \rangle$ of Table i ($i \in [1, 4]$) (in the paper) iff u is irrelevant for Δ w.r.t. $\mathcal{S}$ and an $\mathcal{S}$-extension E of $u(\Delta)$ can be obtained directly from E_0 as described in the caption of Table i.

(STEP 2) The AF for an ASAF

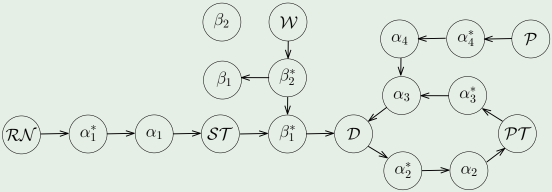
The set of arguments of the AF consists of the arguments of Δ plus a pair of arguments, α and α^*, for each attack α in Δ and a pair of arguments, β and β^*, for each support β in Δ.

Example (The AF for the ASAF Δ_1)

ASAF Δ_1



Δ_{AF}: the AF for Δ_1



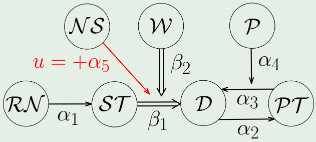
(Theorem 2)

Let $\Delta = \langle \mathbb{A}, \mathbb{R}, \mathbb{S} \rangle$ be an ASAF, $\Delta_{AF} = \langle A, R \rangle$ its AF, $E \subseteq (\mathbb{A} \cup \mathbb{R} \cup \mathbb{S})$, $E' \subseteq A$ and $\mathcal{S} \in \{\texttt{co}, \texttt{pr}, \texttt{st}, \texttt{gr}\}$. It holds that E is an $\mathcal{S}$-extension of Δ iff $\texttt{ASAFtoAF}(E)$ is an $\mathcal{S}$-extension of Δ_{AF}. Equivalently, it holds that E' is an $\mathcal{S}$-extension of Δ_{AF} iff $\texttt{AFtoASAF}(E')$ is an $\mathcal{S}$-extension of Δ.

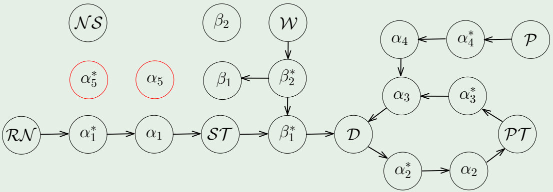
(STEP 3) Set of Updates Corresponding to *u*

Example (Addition update)

ASAF Δ_1 and $u(\Delta_1)$



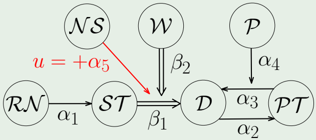
1st: adding meta-arguments



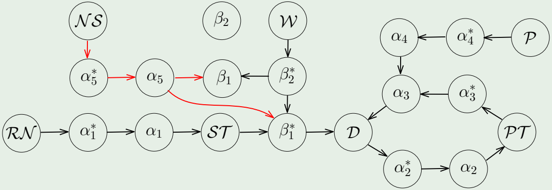
(STEP 3) Set of Updates Corresponding to u

Example (Addition update)

ASAF Δ_1 and $u(\Delta_1)$



2nd: adding meta-attacks



Incremental Algorithm

Algorithm *DynamicASAF*$(\Delta_0, u, \mathcal{S}, E_0, \text{Solver}_{\mathcal{S}})$

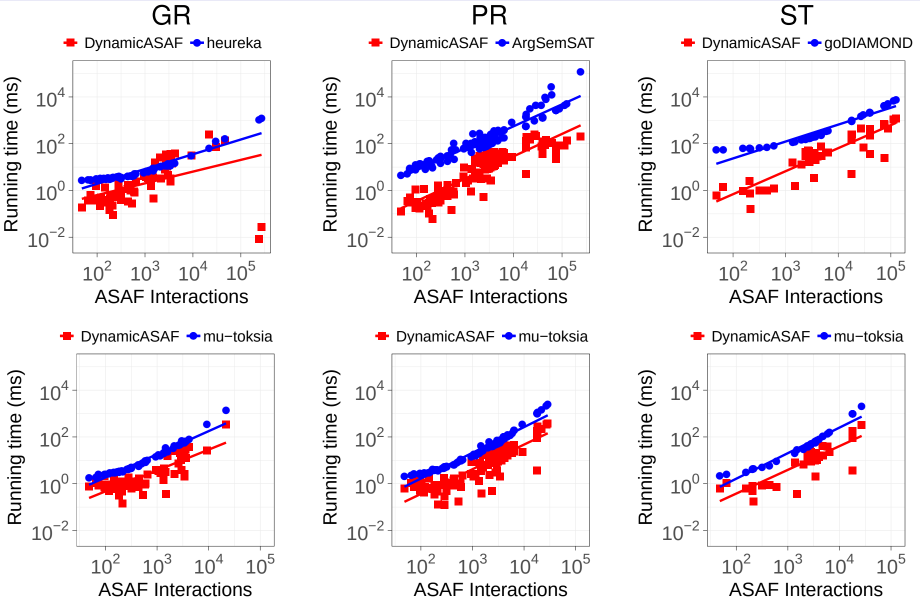
Input: ASAF $\Delta_0 = \langle \mathbb{A}, \mathbb{R}, \mathbb{S} \rangle$, update $u = \pm X$ (X is either an attack or a support), semantics $\mathcal{S} \in \{\texttt{co}, \texttt{pr}, \texttt{st}, \texttt{gr}\}$, initial $\mathcal{S}$-extension E_0 of Δ_0, function $\text{Solver}_{\mathcal{S}}(\langle \text{A}, \text{R} \rangle)$ returning an $\mathcal{S}$-extension for an AF $\langle \text{A}, \text{R} \rangle$ if it exists, $\perp$ otherwise.

Output: $\mathcal{S}$-extension E for $u(\Delta_0)$ if it exists, $\perp$ otherwise.

if *checkIrrelevantUpdate*$(\Delta_0, u, E_0, \mathcal{S})$ **then**
    Obtain E from E_0 as per Theorem 1; // Simply obtain the extension to return
    return E**;**
Let $\Delta_{AF} = \langle \text{A}, \text{R} \rangle$ be the AF for Δ_0; // Build the AF for the ASAF
Let $\Delta'_{AF} = \langle \text{A} \cup \{X, X^*\}, \text{R} \rangle$ if $u = +X$, otherwise $\Delta'_{AF} = \Delta_{AF}$; // Add to it meta arguments if positive update
Let u' be the set of updates over Δ'_{AF} corresponding to u; // Identify the set of updates of the AF for the ASAF, corresponding to u
Let $E'_0 = \texttt{ASAFtoAF}(E_0) \cup E_X$, where $E_X = \{X, X^*\}$ if $u = +X$, otherwise $E_X = \emptyset$; // Obtain the initial extension of the AF for the ASAF
Let $E' = \text{Incr-Alg}(\Delta'_{AF}, u', \mathcal{S}, E'_0, \text{Solver}_{\mathcal{S}})$; // Call external incremental solver
if $(E' \neq \perp)$ **then**
    return $E = \texttt{AFtoASAF}(E')$; // Return an extension of the updated ASAF
else
    return $\perp$; // It can be the case only for stable semantics

Outline

Experimental Results



Outline

Conclusions and future work

We introduced a technique for the incremental computation of extensions of dynamic ASAFs

Given the generality of the ASAF, our technique can be also applied to AFRAs and AFNs

We identified a tighter portion of the updated ASAF to be examined for recomputing the extensions

Our experiments showed that the incremental technique outperforms the computation from scratch

As future work we plan to extend our technique to deal with other computational problems, such as enumerating extensions and deciding credulous/skeptical acceptance in dynamic ASAFs, as well as extend our technique to cope with other AFs with higher-order interactions such as *Recursive Argumentation Framework with Necessities*.

Conclusions and future work

We introduced a technique for the incremental computation of extensions of dynamic ASAFs

Given the generality of the ASAF, our technique can be also applied to AFRAs and AFNs

We identified a tighter portion of the updated ASAF to be examined for recomputing the extensions

Our experiments showed that the incremental technique outperforms the computation from scratch

As future work we plan to extend our technique to deal with other computational problems, such as enumerating extensions and deciding credulous/skeptical acceptance in dynamic ASAFs, as well as extend our technique to cope with other AFs with higher-order interactions such as *Recursive Argumentation Framework with Necessities*.

Thank you!

... any ~~question~~ argument?