An Efficient Algorithm for Skeptical Preferred Acceptance in Dynamic Argumentation Frameworks

GIANVINCENZO ALFANO, SERGIO GRECO, FRANCESCO PARISI

Department of Informatics, Modeling, Electronics and System Engineering, University of Calabria, ITALY

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

ABSTRACT ARGUMENTATION

An *(abstract) argumentation framework* (*AF*) is a pair $\langle A, \Sigma \rangle$, where A is a set of *arguments* and $\Sigma \subseteq A \times A$ is a set of *attacks*.

- It allows representing dialogues, making decisions, and handling inconsistency;
- An AF can be viewed as a direct graph, whose nodes are arguments and whose edges are attacks.

SEMANTICS FOR AFS

An argumentation semantics specifies the criteria for identifying "reasonable" sets of arguments, called *extensions*.

- A *preferred extension* (pr) is a maximal (w.r.t. $\subseteq$) admissible set.
- An *ideal extension* (id) is the biggest (w.r.t. $\subseteq$) admissible set which is contained in every preferred extension.

An argument is **skeptically accepted** under the preferred semantics iff it belongs to every preferred extension.

UPDATES

An *update* u for an AF $\mathcal{A}_0$ consists in modifying $\mathcal{A}_0$ into an AF $\mathcal{A}$ by adding or removing arguments or attacks.

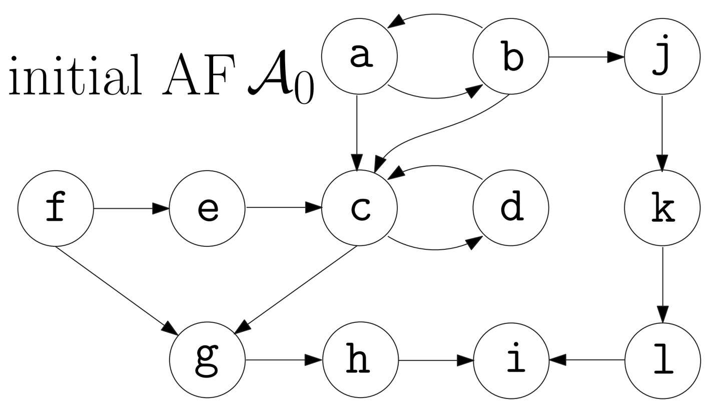
- $+(a,b)$ (resp. $-(a,b)$) denotes the addition (resp. deletion) of an attack (a,b);
- $u(\mathcal{A}_0)$ means applying $u = \pm(a,b)$ to $\mathcal{A}_0$;
- **multiple (attacks) updates** can be simulated by a single attack update.

DYNAMIC ARGUMENTATION FRAMEWORKS

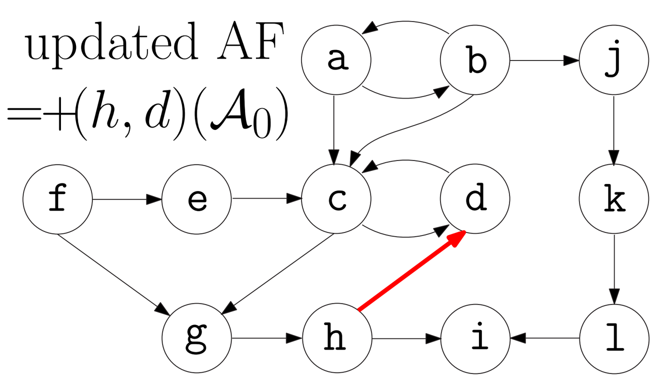
- An argumentation framework models a temporary situation as **new arguments and attacks can be added/removed** to take into account new available knowledge.
- The set of arguments skeptically accepted under the preferred semantics may change if we update an initial AF $\mathcal{A}_0$ by adding/removing arguments/attacks. For instance, the skeptical acceptance under the preferred semantics of goal argument d is *true* in $\mathcal{A}_0$ but false in the updated AF $\mathcal{A} = +(h,d)(\mathcal{A}_0)$ obtained from $\mathcal{A}_0$ by adding attack (h,d). This is due to the change of the set of the preferred extensions.

$\mathcal{S}$	Set of initial extensions	Set of updated extensions
pr	$\{\{a,d,f,h,j,l\},\{b,d,f,h,k\}\}$	$\{\{a,f,h,j,l\},\{b,f,h,k\}\}$
id	$\{\{d,f,h\}\}$	$\{\{f,h\}\}$

initial AF $\mathcal{A}_0$



updated AF $\mathcal{A}=+(h,d)(\mathcal{A}_0)$



- **Should we recompute the skeptical acceptance of updated AFs from scratch?**

CONTRIBUTIONS

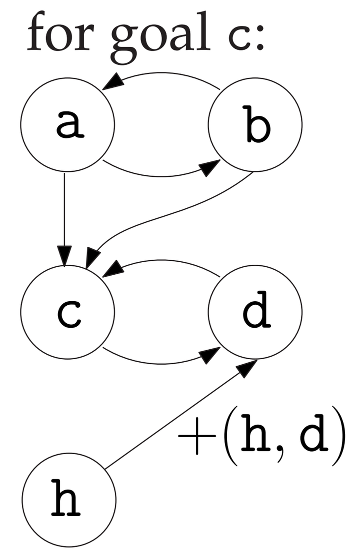
We show that computing a small portion of the input AF is sufficient to determine the skeptical acceptance of a goal argument in the updated AF.

We introduce SPA, an incremental algorithm for computing the Skeptical Preferred Acceptance of a goal within a dynamic AF. It consists of the following main steps:

- Identify a sub-AF called *context-based* AF on the basis of updates and additional information provided by the ideal extension.
- Give as input the context-based AF to an external (non-incremental) solver to compute (i) the skeptical preferred acceptance of the goal argument, and (ii) the ideal extension for the updated AF.

We provide a thorough experimental analysis showing the effectiveness of our approach.

Context-based AF for goal c:



EXPERIMENTS

Datasets: ICCMA'17 benchmarks.

For each AF in the dataset, we compared the performance of our technique with that of *ArgSemSAT*, the solver that won the last ICCMA competition for the computational task DS-pr: Given an AF, determine the skeptical preferred acceptance of a given argument.

Results: The figures report the improvements (running time of the competitor over running time of our approach) of SPA and SPA-base versus the number of attacks. SPA-base is a version of SPA not using the ideal extension.

- Considering the averages of the improvements, SPA and SPA-base turn out to be 5 and 4 orders of magnitude faster than *ArgSemSAT*, respectively. However, as this can be skewed by extremely large values of improvements (e.g. 10^6), we also considered the medians of improvements for SPA (32 on $A2$, 134 on $A3$) and SPA-base (27 on $A2$, 40 on $A3$) (see dashed line), which confirm the significance of the gain in efficiency.
- SPA is generally faster than SPA-base, except for a few AFs whose initial ideal extension is empty.
- The performance gets worse when the ratio between the size of the context-based AF and that of the initial AF becomes very large because of the increasing density of the initial AFs.
- For sets of updates, results show that SPA remains faster than the competitor even when 10 or 100 updates are performed simultaneously.
- Finally, applying updates simultaneously is faster than applying them sequentially.