Abstract argumentation frameworks - overview and illustration

Patrick Bellositz

1 Introduction

2 Definitions

Definition 1 An *argumentation framework* F is a pair (A, R), where A is a set of arguments and R is a set of attack relations.

Definition 2 *Attack relations* $R \subseteq A \times A$ represent attacks. The pair (a, b), where $a, b \in A$ means a *attacks* b.

Example 1 Imagine we have 3 arguments a_1, a_2, and b.

$$
\begin{aligned}
a_1 &= \text{"Blue is the most beautiful of all colors."} \\
b &= \text{"No, black is much more beautiful!"} \\
a_2 &= \text{"That's wrong, black isn't even a color."}
\end{aligned}
$$

These arguments obviously contain 2 attacks. Argument b attacks argument a_1 and in turn argument a_2 attacks argument b.

This results in the framework $F = (A, R)$, where $A = \{a_1, a_2, b\}$ and $R = \{(b, a_1), (a_2, b)\}$.

As writing an argument framework in sets might become hard to read with increasing set sizes, it is also possible to write every framework as a graph (V, E), where $V = A$ and $E = R$. The graph in our example looks like this:

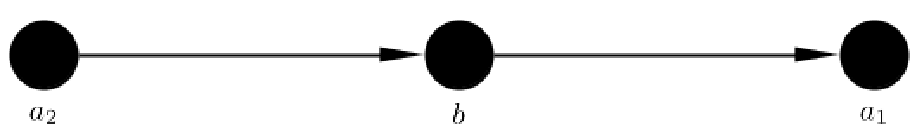

Figure 1: Example graph 1

Remark 1 Let S be a set of arguments. If $a \in S$ and there is an attack $(a, b) \in R$ we say S *attacks* b.

Definition 3 Let S be a set of arguments. It is *conflict-free*, if $\forall a \forall b\ a, b \in S, (a, b) \notin R$.

Example 2 (Continuation of Example 1) As no argument attacks itself, $\{a_1\}$, $\{a_2\}$ and $\{b\}$ each are conflict-free. $\{a_1, a_2\}$ is also a conflict-free set, since there exists no attack relation containing a_1 and a_2. The empty set is always conflict-free.
Since there is an attack relation between b and each of the other arguments, there are no other conflict-free sets.

Thus the set of conflict-free sets is $cf(F) = \{\emptyset, \{a_1\}, \{a_2\}, \{b\}, \{a_1, a_2\}\}$.

Definition 4 An argument a is *defended* by a set S, if for every attack $(b, a) \in R$ there is an attack (c, b), where $c \in S$. If this is the case S *defends* a.

Definition 5 Let S be a conflict-free set. It is called an *admissible extension* if it defends each $a \in S$.

Example 3 (Continuation of Example 2) Of the conflict-free sets only $\{a_2\}$ and $\emptyset$ don't get attacked. They are admissible. $\{a_1, a_2\}$ gets attacked via the attack relation (b, a_1), but a_1 gets defended through (a_2, b), making it also admissible. $\{b\}$ and $\{a_1\}$ are not admissible since they don't defend their arguments.

Thus the set of admissible extensions is $adm(F) = \{\emptyset, \{a_1, a_2\}, \{a_2\}\}$.

Definition 6 Let S be an admissible extension. It is called a *preferred extension* if for each $S' \subseteq A$, that is an admissible extension, $S \not\subset S'$.

Example 4 (Continuation of Example 3) $\emptyset \subset \{a_2\}$, therefore $\emptyset$ is not a preferred extension. $\{a_2\} \subset \{a_1, a_2\}$, therefore $\{a_2\}$ is not a preferred extension.
Since all other admissible extensions are proper subsets of $\{a_1, a_2\}$, it is a preferred extension.

Thus the set of preferred extensions is $prf(F) = \{\{a_1, a_2\}\}$.

Definition 7 Let S be a conflict-free set. It is called a *stable extension* if for each $a \notin S$ there is exists an attack $(b, a) \in R$ where $b \in S$.

Example 5 (Continuation of Example 2)

Definition 8 Let S be an admissible extension. It is called a *complete extension* if for each $a \notin S$, $S \cup \{a\}$ is not an admissible extension.

Example 6 (Continuation of Example 3)

Definition 9 The (unique) *grounded extension* is defined by $\bigcap_{i=1}^{n} S_i$, where

$\{S_1, ..., S_n\}$ is the set of all complete extensions.

Example 7 (Continuation of Example 6)

3 Observations

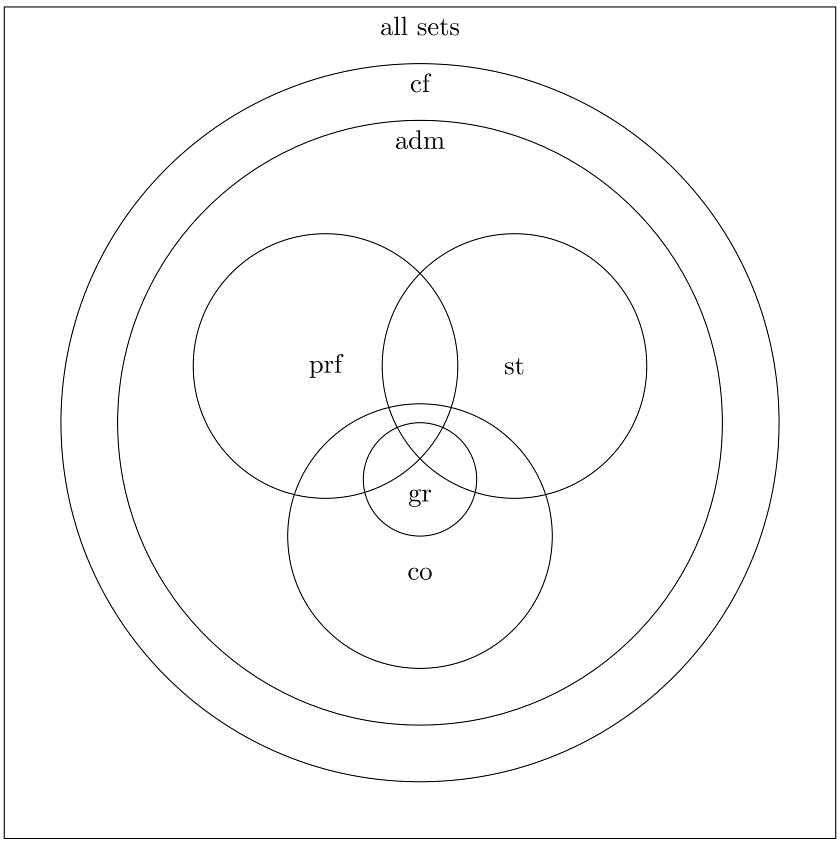

Figure 2: Relation between extension types

4 Application

4.1 Introduction

In this section the usage and implementation details of the aforementioned program illustrating the computation of the different extension types is provided.

4.2 Creation of a framework

On starting the application the user is presented with an input mask. It consists of ten rows, each representing an argument and a button labeled ”show graph”.

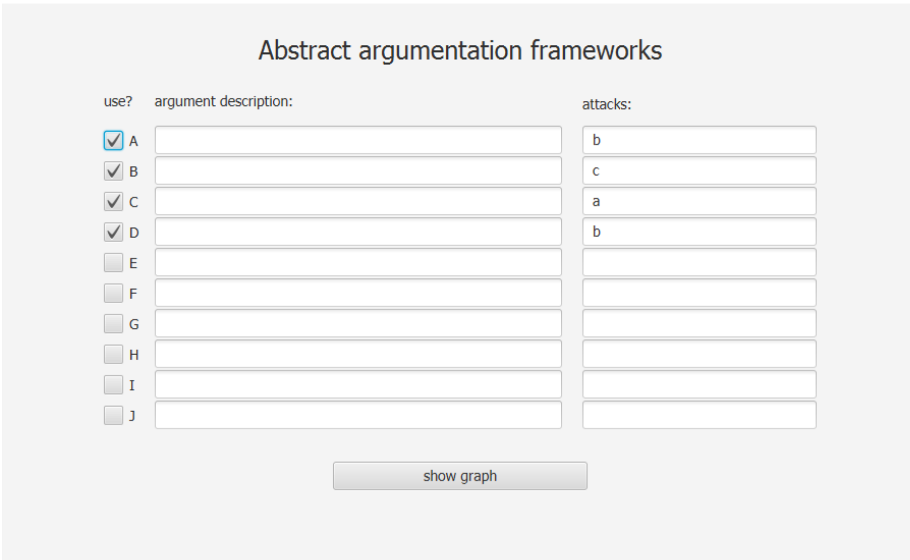

Figure 3: Input mask

4.3 Argumentation graph

Once a framework is created and ”show graph” was clicked a graphical representation of it is shown alongside buttons to compute extensions and an area dedicated to showing computation results.

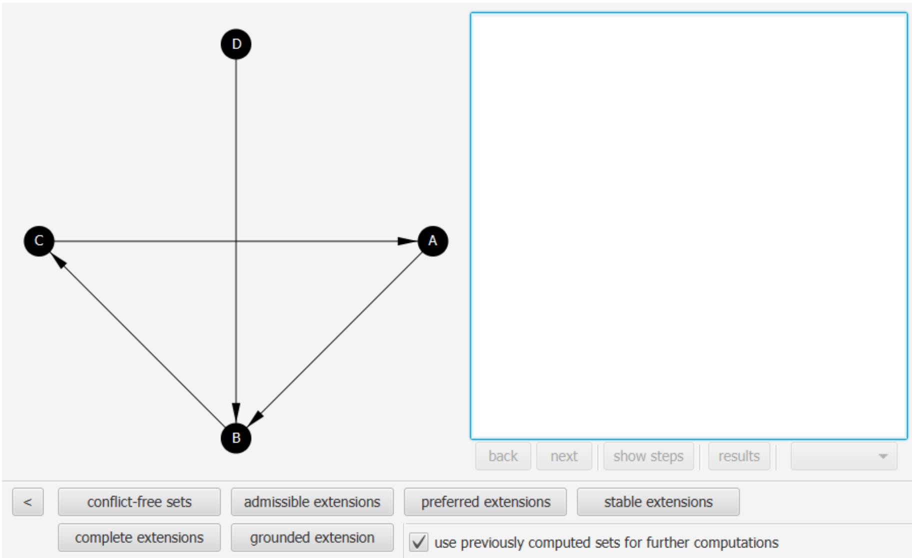

Figure 4: Demonstration view

4.4 stuff comes here