Merging Argumentation Frameworks

Lucas Leite, Thiago Alves, João Alcântara
Department of Computer Science
Federal University of Ceará
Fortaleza, Ceará 60455-760
Email: {lgoncalves,thiagoalves,jnando}@lia.ufc.br

***Abstract*—The aim of this paper is to define an arbitration merging operator for argumentation frameworks. As it is known, an argumentation framework is a collection of defeasible proofs, called arguments, and a relation attack between these arguments. Many of such arguments can be put forward by different agents and represent their different points of view. The problem is how to merge these frameworks to obtain a unique framework reflecting the arguments of the group. We overcome it by resorting to a semantic approach that selects those arguments and their attacks of an agent that vary the least from the arguments and attacks of the other agents. Then, we proved our proposal satisfy some reasonable postulates and show a procedure to build an argumentation framework resulting from this arbitration merging operator.**

I. Introduction

Often agents are urged to come to an agreement about what to believe. Then, they can put forward various arguments for (or against) a point of view aiming at establishing a rationally justifiable position. If all of these arguments are mutually consistent, then an agreement can be easily reached. Nevertheless, not rarely, because of inconsistencies among the beliefs of multiple agents, they can propose for consideration conflicting arguments. In such cases, what would be the point of view of a group of agents representing the arguments and their conflicts? Inspired by [1], we introduce in this work a semantic approach for merging argumentation frameworks to obtain the rationally justifiable position of a group of agents.

Consider the following example: John and Joanna are a lovely couple deciding what they would do this night. John, a simple-minded man, has the arguments

- a: We will will watch football because we will not have a romantic dinner.
- b: We will have a romantic dinner because we will not watch football.

On the other hand, Joanna has the following arguments:

- b: We will have a romantic dinner because we will not watch football.
- c: We will go shopping because we will not have a romantic dinner.
- d: We will watch the *Twilight* saga for the fifth time because we will not go shopping.

According to Dung [2], an argumentation framework is a pair $AF = (Ar, att)$, in which Ar is a set of arguments and att is an attack relation between two arguments in Ar. Using our terminology, they are trying to merge their argumentation frameworks to obtain their rationally justifiable position. Note that in John's framework, argument a attacks b and vice-versa; John is unaware of both arguments c and d. Regarding Joanna's framework, argument b attacks c, and c attacks d; she is unaware of argument a.

If John and Joanna want to come to an agreement about their arguments by conciliating their own arguments frameworks, they should preserve as much information as possible in a merging process known as *arbitration*. This means that the resulting arguments and their attacks of this arbitration process should represent those closest to each individual in this dispute. In the example above, what would be the most satisfactory arguments for both John and Joanna?

Due to the defeasible nature of argumentation frameworks, an arbitration operator for them demands extra efforts compared to monotonic formalisms as Propositional Logic.

In order to overcome this problem and inspired by [1], we will define the notion of SE-extension to be employed in the definition of our arbitration operator $\triangledown$ for argumentation frameworks. Indeed, $\triangledown$ will be tailored to select those SE-extensions that are closest to the SE-extensions of the argumentation frameworks to be merged. Then we show our operator satisfies all the expected properties of an arbitration operator. Furthermore, given the SE-extensions obtained from $\triangledown$, we define a procedure to build the argumentation framework resulting from this arbitration merging process.

In the next section, we focus on the fundamental notions on which our work is settled: argumentation frameworks and arbitration. Section III is devoted to the main contribution of this work: the definition of our arbitration merging operator $\triangledown$ for argumentation frameworks. Then, in the sequel, we show that $\triangledown$ satisfies eight postulates that any reasonable arbitration operator should satisfy. In Section V, we introduce a procedure to build an argumentation framework resulting from this arbitration merging process. Finally, in Section VI, we conclude our work and motivate further directions.

II. Background

A. Argumentation Framework

Argumentation theory is a reasoning process based on constructing arguments, determining conflicts between arguments and determining acceptable arguments. It is more than merely reasoning on the top of an individual belief base, but can also involve multiagent interaction. Because of this, the field of formal argumentation has found a wide range of applications in many branches of Artificial Intelligence [3], [4].

Its studies can be traced back to Pollock [5], [6], Vreeswijk [7], [8], and Simari and Loui [9]. Unlike proofs, arguments are defeasible. Thus, to be accepted, a claim will depend not only on the existence of an argument supporting this claim, but also on the existence of possible counter arguments.

Nowadays, much research on argumentation is based on Dung's abstract argumentation theory [2]. In this work, he showed that fundamental aspects of argumentation can be studied without regarding the internal structure of the arguments.

Definition 1 (Argumentation Framework)**.** *[2] Dung's argumentation framework is a pair $AF = \langle Ar, att \rangle$, where Ar is a finite set of arguments and $att \subseteq Ar \times Ar$ is a binary attack relation defined over $Ar \times Ar$. We say argument A attacks the argument B iff $(A, B) \in att$.*

Essentially Dung's argumentation framework is a directed graph in which arguments are nodes and the attack relation is represented by arrows. By exploiting this attack relation, we can define the following fundamental definitions

Definition 2. *[2] Let $AF = (Ar, att)$ be an argumentation framework, $A \in Ar$ and $S \subseteq Ar$. We define*

- $S^+ = \{B \in Ar \mid (A, B) \in att \text{ and } A \in S\}$.
- $A^- = \{B \in Ar \mid (B, A) \in att\}$.

Intuitively, the set S^+ represents the set of all arguments attacked by S, whilst, A^- represents the set of all arguments attacking A in the argumentation framework. Now we can characterize the notions of conflict-free and defense

Definition 3 (Conflict-free/Defense)**.** *[2] Let $AF = (Ar, att)$ be an argumentation framework, $A \in Ar$ and $S \subseteq Ar$.*

- *S is* conflict-free *if $S \cap S^+ = \emptyset$.*
- *S defends argument A if $A^- \subseteq S^+$.*
- *The* characteristic function *$\mathcal{F} : 2^{Ar} \rightarrow 2^{Ar}$ of AF is defined as*

$$\mathcal{F}(S) = \{A \in Ar | S \textit{ defends } A\}$$

Intuitively, a set S of arguments is conflict-free if it does not attack any argument in S; we say S defends an argument A if it attacks any argument attacking A.

The output of an argumentation framework AF is a multiset $\mathfrak{M}_{AF}$ of arguments intended to defend a viewpoint or a decision. Each set in $\mathfrak{M}_{AF}$ should satisfy the following requirements to be considered admissible:

Definition 4 (Admissible Set)**.** *Let $S \subseteq Ar$ be a conflict-free set of arguments in the argumentation framework $AF = (Ar, att)$. We say S is an admissible set of AF if S defends every element in S, i.e., $S \subseteq \mathcal{F}(S)$.*

The first condition guarantees that the point of view do not have conflicts between its arguments, that is, the arguments that form a position are not related by att. The second condition guarantees that any argument in the admissible set is defended by another argument in the same set.

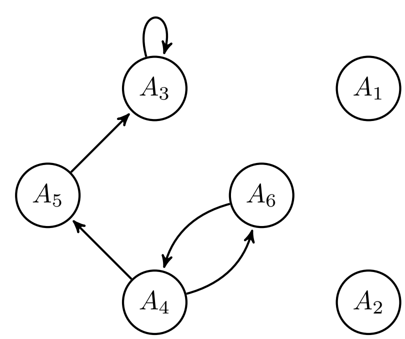
Example 1. *Let $AF = (Ar, att)$ be an argumentation framework with $Ar = \{A, B, C, D, E, F\}$ and $att = \{(A, C), (A, D), (A, F), (B, E), (F, A), (F, B), (F, C), (F, D), (F, E)\}$. The admissible sets of AF are $\{A\}$, $\{A, B\}$ and $\{F\}$.*

Traditional approaches to argumentation semantics are based on extensions of arguments. Some of them can be seen in [2]. Although our arbitration operator does not depend on any of theses semantics in particular, we will refer to stable extensions (see below) to illustrate our results:

Definition 5. *[Stable Argumentation Semantics]. Given an argumentation framework $AF = (Ar, att)$ and $S \subseteq Ar$. We say S is a stable extension of AF iff S is a conflict-free fixpoint of $\mathcal{F}$ (i.e., $S = \mathcal{F}(S)$)) such that $S \cup S^+ = Ar$.*

If S is a stable extension of AF, any argument of AF is either in S or is attacked by an argument in S.

Example 2. *Let $AF = (Ar, att)$ be an argumentation framework such that $Ar = \{A_1, \ldots, A_6\}$ and $att = \{(A_6, A_4), (A_4, A_6), (A_4, A_5), (A_5, A_3), (A_3, A_3)\}$. We depict AF below as a directed graph:*



Then AF has as its unique stable extension: $\{A_1, A_2, A_5, A_6\}$.

Now we will proceed by exploiting the field of belief merging. In Section III we will employ these techniques to merge argumentation frameworks.

B. Arbitration

In advanced knowledge-based systems, a large amount of information can come from different sources with (possibly) not all of them being completely reliable. Besides, these sources can offer not only different, but mutually contradictory views of a situation. In order to deal with such a multifaceted knowledge-base, one cannot just make the union of arguments and attacks of each framework as it may lead to loss of information on the uniqueness of agents. The best we can do is to "merge" these views in a new and consistent one, trying to preserve as much information as possible [10]. Several papers studied merging techniques applied to argumentation frameworks (see [11], [12], [13] and [14]). These merging techniques are based on merging operators and, in this work, we resort to an *arbitration* operator.

Various works on arbitration operators have been presented [15], [10], [16]. In [10], Liberatore and Schaerf address arbitration in general, while Konieczny and Pino Perez [16] consider a general approach in which merging takes place with respect to a set of global constraints, or formulas that must hold in the merging. Based on [10], [16], J. Delgrande *et al* [1] proposed a new arbitration operator for logic programs. As in [10], they consider merging two belief bases settled on the intuition that models of the merged bases should be taken from those of each belief base closest to the others. They consider a propositional language over a finite set of atoms; consequently their arbitration operator can be expressed as a binary operator on formulas. The following postulates applied to a propositional language characterize this operator:

Definition 6. *[1] We say $\diamond$ is an arbitration operator if it satisfies the following postulates.*

- (LS1) $\vdash \alpha \diamond \beta \equiv \beta \diamond \alpha$
- (LS2) $\vdash \alpha \wedge \beta \supset \alpha \diamond \beta$
- (LS3) *If* $\alpha \wedge \beta$ *is satisfiable then* $\vdash \alpha \diamond \beta \supset \alpha \wedge \beta$
- (LS4) $\alpha \diamond \beta$ *is unsatisfiable iff* α *is unsatisfiable and* β *is unsatisfiable*
- (LS5) *If* $\vdash \alpha_1 \equiv \alpha_2$ *and* $\vdash \beta_1 \equiv \beta_2$ *then* $\vdash \alpha_1 \diamond \beta_1 \equiv \alpha_2 \diamond \beta_2$
- (LS6) $\alpha \diamond (\beta_1 \vee \beta_2) = \alpha \diamond \beta_1$ *or* $\alpha \diamond \beta_2$ *or* $(\alpha \diamond \beta_1) \vee (\alpha \diamond \beta_2)$
- (LS7) $\vdash (\alpha \diamond \beta) \supset (\alpha \vee \beta)$
- (LS8) *If* α *is satisfiable then* $\alpha \wedge (\alpha \diamond \beta)$ *is satisfiable*

As emphasized in [1], (LS1) asserts that merging is commutative, while the next two guarantee that, for mutually consistent formulas, merging corresponds to their conjunction. (LS4) guarantees that the result of merging will be inconsistent if and only if both of the formulas to be merged are inconsistent themselves. (LS5) ensures the operator is independent of syntax, while (LS6) provides a “factoring” postulate. As for (LS7), it asserts the result of merging implies the disjunction of the original formulas. The last postulate informally constrains the result of merging so that each operator “contributes to” (i.e. is consistent with) the final result.

III. Arbitration Merging of Argumentation Frameworks

A. Arbitration Merging

We will present the main results of our work; namely, the definition of an arbitration operator for argumentation frameworks in terms of a semantic approach. We also prove that it satisfies those postulates introduced in Definition 6.

Due to the defeasible nature of argumentation frameworks, an arbitration operator for them demands extra efforts compared to monotonic formalisms as Propositional Logic. Indeed, as far as we are aware, the problem of merging via arbitration under a semantic perspective has not been addressed for argumentation frameworks.

A key challenge here is how to merge non-monotonic knowledge bases. In [1], the authors have resorted to what they called SE-models (as ordinary models are not suitable) to merge normal logic programs. It is known that normal logic programs have also a defeasible semantics. Furthermore, there is a correspondence between some logic programs semantics and argumentation frameworks (see [2]).

Considering this relation involving semantics of normal logic programs and extensions of argumentation frameworks and inspired by [1], we will define a new extension to be employed in the definition of our arbitration operator for argumentation frameworks: SE-Extension.

Definition 7 (SE-extension)**.** *Let $AF = (Ar, att)$ be an argumentation framework, and X and Y be sets of arguments. We say a pair (X, Y) is an SE-extension of AF if*

$$Ar \setminus Y^+ \subseteq X \subseteq Y$$

An SE-extension is a pair (X, Y) of arguments (with $X \subseteq Y$) such that X and Y should at least contain the arguments not attacked by Y. It has been tailored as above to guarantee the following immediate result:

Theorem 1. *Let $AF = (Ar, att)$ be an argumentation framework, and Y a subset of Ar. Then Y is a stable extension of AF if and only if (Y, Y) is an SE-extension of AF and there is no SE-extension (X, Y) such that $X \subset Y$.*

We will employ this notion of SE-extension to define our arbitration operator. Before, however, we will introduce some fundamental notation:

Definition 8 (Argumentation Profile)**.** *We define an argumentation profile Ψ as the sequence $\langle AF_1, \ldots, AF_n \rangle$ of argumentation frameworks $AF_i = (Ar_i, att_i)$.*

Definition 9 (Extensions of an Argumentation Profile)**.** *The set of the extensions of an Argumentation Profile is defined by $Ext(\Psi) = Ext(AF_1) \times \cdots \times Ext(AF_n)$, where each $Ext(AF_i)$, $1 \leq i \leq n$, is the set $\{Y \mid Ar_i \setminus Y^+ \subseteq Y\}$. We use X_i to denote the i^{th} component of $\overline{X} \in Ext(\Psi)$.*

Definition 10 (SE-extensions of an Argumentation Profile)**.** *The set of SE-extensions of Ψ is given by $SE(\Psi) = SE(AF_1) \times \cdots \times SE(AF_n)$, where each $SE(AF_i)$, $1 \leq i \leq n$, is the set of all SE-extensions of AF_i. For each $\overline{S} = \langle S_1, \ldots, S_n \rangle \in SE(\Psi)$, we use S_i to denote the i^{th} component of $\overline{S}$, i.e., $S_i \in SE(AF_i)$.*

Furthermore, for any argumentation frameworks AF_1 and AF_2, by $AF_1 \sqcup AF_2$ we mean an argumentation framework whose SE-extensions are $SE(AF_1) \cup SE(AF_2)$; similarly, by $AF_1 \sqcap AF_2$, we mean an argumentation framework whose SE-extensions are $SE(AF_1) \cap SE(AF_2)$. We say AF_1 and AF_2 are strongly equivalent ($AF_1 \equiv_s AF_2$) iff $SE(AF_1) = SE(AF_2)$. We also write $AF_1 \models_s AF_2$ to mean that $SE(AF_1) \subseteq SE(AF_2)$.

In an SE-Extension (X, Y) of an argumentation framework AF, the intuition is that if one conjecture every argument in Y is possibly true, X represents a candidate to be a stable extension of AF. In this case, X should at least contain those arguments not attacked by Y. With that motivation in mind, we define an argumentation framework AF is *satisfiable* if

$SE(AF) \neq \emptyset$. Every argumentation framework (Ar, att) is satisfiable and has at least (Ar, Ar) as SE-extension.

The argumentation frameworks in Ψ are those to be merged. Our aim is to define the SE extensions of the merged argumentation framework as the least different than the SE extensions of frameworks in Ψ. As in [1], this will be captured in our approach by resorting to the symmetric difference operator:

Definition 11 (Symmetric Difference Operator)**.** *Let X and Y be two sets. By $X \ominus Y$, we mean the set*

$$(X \setminus Y) \cap (Y \setminus X)$$

For every pair of sets (X_1, Y_1) and (X_2, Y_2),

$$(X_1, Y_1) \ominus (X_2, Y_2) = (X_1 \ominus X_2, Y_1 \ominus Y_2)$$

We say $(X_1, Y_1) \subseteq (X_2, Y_2)$ if and only if $X_1 \subseteq X_2$ and $Y_1 \subseteq Y_2$. Besides, $(X_1, Y_1) \subset (X_2, Y_2)$ if and only if$(X_1, Y_1) \subseteq (X_2, Y_2)$ and $X_1 \subset X_2$ or $Y_1 \subset Y_2$.

The output of our arbitration operator will be the sets of SE-extensions closest to the SE-extensions of each framework to be merged. With this aim in mind, we introduce a notion of distance between SE-extensions of Ψ:

Definition 12. *Let $\Psi = \langle AF_1, \ldots, AF_n \rangle$ be an argumentation profile and let $\overline{S}, \overline{T}$ be two SE-extensions of Ψ (or two sets of arguments in $Ext(\Psi)$). Then, define*

$$\overline{S} \leq_a \overline{T} \textit{ if } S_i \ominus S_j \subseteq T_i \ominus T_j \textit{ for every } 1 \leq i < j \leq n$$

Considering that $\leq_a$ is a partial pre-order, we can define

$$Min_a(N) = \left\{ \overline{S} \in N \mid \overline{T} \leq_a \overline{S} \text{ implies } \overline{S} \leq_a \overline{T} \text{ for } \overline{T} \in N \right\}$$

Note that $Min_a(N)$ represents the set of all minimal elements (with respect to $\leq_a$) of a set N of tuples. For a set N of n-tuples, we will also resort to the definition below when characterizing our arbitration merging.

$$\cup N = \left\{ S_i \mid \overline{S} \in N \text{ and } i \in \{1, \ldots, n\} \right\}.$$

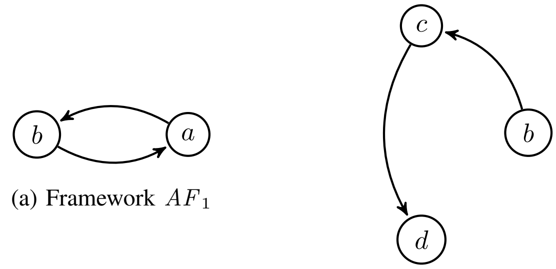
Then, the argumentation arbitration merging of a profile Ψ, denoted by $\triangledown(\Psi)$, can be defined as follows:

Definition 13 (Argumentation arbitration)**.** *Let Ψ be an argumentation profile. The arbitration merging of Ψ is an argumentation framework such that*

$$\begin{aligned} SE(\triangledown(\Psi)) = \{(X, Y) \mid Y \in \cup Min_a(Ext(\Psi)), X \subseteq Y, \textit{ and} \\ \textit{if } X \subset Y \textit{ then } (X, Y) \in \cup Min_a(SE(\Psi))\} \end{aligned}$$

Note $SE(\triangledown(\Psi))$ selects those SE-extensions of the corresponding merged framework closest (according to Definitions 11 and 12) to the SE-extensions of frameworks in Ψ. We emphasize $SE(\triangledown(\Psi))$ is not an argumentation framework, but the set of all SE-extensions of the corresponding merged framework. In Section V, we propose an algorithm to obtain the corresponding argumentation. We proceed by recalling the example (involving John and Joanna) exhibited in the Introduction, which can be translated into the following argumentation frameworks. Then, we will employ $\triangledown$ to merge them:

Example 3. *Let $\Psi = \langle AF_1, AF_2 \rangle$, where $AF_1 = (Ar_1, att_1)$ and $AF_2 = (Ar_2, att_2)$ are shown in Fig. 1.*



(a) Framework AF_1

(b) Framework AF_2

Fig. 1: Frameworks of Argumentation to Be Merged

From these frameworks, we obtain

- $Ext(AF_1) = \{\{a\}, \{b\}, \{a, b\}, \{a, c\}, \{a, d\} \{a, b, d\}, \{b, c\}, \{b, d\}, \{b, c, d\}, \{a, b, c\}, \{a, b, d\}, \{a, b, c, d\}\}$
- $Ext(AF_2) = \{\{b, c\}, \{b, d\}, \{b, c, d\}, \{a, b, c\} \{a, b, d\}, \{a, b, c, d\}\}$

The set $\cup Min_a(Ext(\Psi))$ was tailored to select the closest sets of arguments regarding $Ext(AF_1)$ and $Ext(AF_2)$. As $Ext(AF_1) \cap Ext(AF_2) \neq \emptyset$, $\cup Min_a(Ext(\Psi)) = Ext(AF_1) \cap Ext(AF_2) = Ext(AF_2)$

Similarly, $\cup Min_a(SE(\Psi))$ was tailored to select the closest sets of SE-extensions of AF_1 and AF_2. As $SE(AF_1) \cap SE(AF_2) \neq \emptyset$, $\cup Min_a(SE(\Psi)) = SE(AF_1) \cap SE(AF_2)$

Indeed, $SE(\triangledown(\Psi)) = \cup Min_a(SE(\Psi)) =$

$$\left\{ \begin{array}{ll} \boldsymbol{(\{b,d\},\{b,d\})}, & (\{b,c\},\{b,c\}), \\ (\{b\},\{b,c\}), & (\{a,b,c\},\{a,b,c\}), \\ (\{a,b\},\{a,b,c\}), & (\{b,c\},\{a,b,c\}), \\ (\{b\},\{a,b,c\}), & (\{a,b,d\},\{a,b,d\}), \\ (\{b,d\},\{a,b,d\}), & (\{b,c,d\},\{b,c,d\}), \\ (\{b,c\},\{b,c,d\}), & (\{b,d\},\{b,c,d\}), \\ (\{b\},\{b,c,d\}), & (\{a,b,c,d\},\{a,b,c,d\}), \\ (\{a,b,c\},\{a,b,c,d\}), & (\{a,b,d\},\{a,b,c,d\} \\ (\{b,c,d\},\{a,b,c,d\}), & (\{b,c\},\{a,b,c,d\}), \\ (\{b,d\},\{a,b,c,d\}), & (\{a,b\},\{a,b,c,d\}), \\ (\{b\},\{a,b,c,d\})\}), & \end{array} \right\} \quad (1)$$

These are the SE-extensions of the argumentation framework obtained by merging AF_1 and AF_2. One can check that $SE(\triangledown(\Psi))$ has a unique SE-extension (the element bold-typed above) representing its unique stable model extension $\{b, d\}$. Then, according to this extension, the most satisfactory arguments for both John and Joanna are b and d.

Now we will prove $\triangledown$ satisfies all those postulates showed in [1] to identify a good arbitration operator (Definition 6).

IV. Showing Conformity of $\triangledown$ with the Arbitration Postulates

Denoting $AF_1 \diamond AF_2$ as $\triangledown(AF_1, AF_2)$, we will adapt slightly the postulates in Definition 6 to deal with argumentation framework and prove that $\triangledown$ satisfies all of them.

Theorem 2. *Let Ψ be an argumentation profile. Then ∇ satisfies the following postulates:*
(LS1') $\nabla(AF_1, AF_2) \equiv_s \nabla(AF_2, AF_1)$.

Proof. Let $(X_1, Y_1) \in SE(\nabla(AF_1, AF_2))$. Then $Y_1 \in \cup Min_a(Ext(AF_1, AF_2))$. By definition of $\cup Min_a$, $Y_1 \in \cup Min_a(Ext(AF_2, AF_1))$. If $X_1 = Y_1$, then $(X_1, Y_1) \in SE(\nabla(AF_2, AF_1))$. If $X_1 \neq Y_1$, then $(X_1, Y_1) \in \cup Min_a(SE(AF_1, AF_2))$ and by definition of $\cup Min_a$, $(X_1, Y_1) \in \cup Min_a(SE(AF_2, AF_1))$. Thus, $(X_1, Y_1) \in SE(\nabla(AF_2, AF_1))$. The converse is analogous. $\square$

(LS2')$AF_1 \sqcap AF_2 \models_s \nabla(AF_1, AF_2)$.

Proof. If $(X_1, Y_1) \in SE(AF_1) \cap SE(AF_2)$, then $(X_1, Y_1) \in SE(AF_1)$ and $(X_1, Y_1) \in SE(AF_2)$. It follows that $Y_1 \models AF_1$ and $Y_1 \models AF_2$. Thus, $(Y_1, Y_1) \in Ext(AF_1, AF_2)$ and as consequence, $Y_1 \in \cup Min_a(Ext(AF_1, AF_2))$. From $(X_1, Y_1) \in SE(AF_1)$ and $(X_1, Y_1) \in SE(AF_2)$ we have that $((X_1, Y_1), (X_1, Y_1)) \in SE(AF_1, AF_2)$. As $(X_1, Y_1) \ominus (X_1, Y_1) = (\emptyset, \emptyset)$, $(X_1, Y_1) \in \cup Min_a(SE(AF_1, AF_2))$. Thus, $(X_1, Y_1) \in SE(\nabla(AF_1, AF_2))$. $\square$

(LS3') If $AF_1 \sqcap AF_2$ is satisfiable then $\nabla(AF_1, AF_2) \models_s AF_1 \sqcap AF_2$.

Proof. Assume that there is an SE-extension $(X_1, Y_1) \in SE(AF_1) \cap SE(AF_2)$. Then $Min_a(SE(AF_1, AF_2)) = \{((X, Y), (X, Y)) \mid (X, Y) \in SE(AF_1) \cap SE(AF_2)\}$ and $Min_a(Ext(AF_1, AF_2)) = \{(Y, Y) \mid Y \in Ext(AF_1) \cap Ext(AF_2)\}$.

Assume that $(X_2, Y_2) \in SE(\nabla(AF_1, AF_2))$. Then, $Y_2 \in \cup Min_a(Ext(AF_1, AF_2))$. If $X_2 = Y_2$, $(Y_2, Y_2) \in Min_a(Ext(AF_1, AF_2))$. Thus, $Y_2 \in Ext(AF_1)$ and $Y_2 \in Ext(AF_2)$. Hence, $(Y_2, Y_2) \in SE(AF_1)$ and $(Y_2, Y_2) \in SE(AF_2)$, i.e., $(Y_2, Y_2) \in SE(AF_1 \sqcap AF_2)$.

If $X_2 \neq Y_2$, $(X_2, Y_2) \in \cup Min_a(SE(AF_1, AF_2))$. Thus, $((X_2, Y_2), (X_2, Y_2)) \in Min_a(SE(AF_1, AF_2))$. It follows that $(X_2, Y_2) \in SE(AF_1)$ and $(X_2, Y_2) \in SE(AF_1)$, i.e., $(X_2, Y_2) \in SE(AF_1 \cap AF_2)$. $\square$

(LS4')$\nabla(AF_1, AF_2)$ is satisfiable iff AF_1 and AF_2 are satisfiable.

Proof. Assume that there is an SE-extension $(X_1, Y_1) \in SE(AF_1)$ and an SE-extension $(X_2, Y_2) \in SE(AF_2)$. Then, $((X_1, Y_1), (X_2, Y_2)) \in SE(AF_1, AF_2)$ and $(Y_1, Y_2) \in Ext(AF_1, AF_2)$. Thus, there is an SE-extension $(X_3, Y_3) \in SE(\nabla(AF_1, AF_2))$.

Assume that there is an SE-extension $(X_3, Y_3) \in SE(\nabla(AF_1, AF_2))$. Then, for some Y_1, we have $(Y_1, Y_3) \in Ext(AF_1, AF_2)$ or $(Y_3, Y_1) \in Ext(AF_1, AF_2)$. If $(Y_1, Y_3) \in Ext(AF_1, AF_2)$, by definition of SE-extension, $(Y_1, Y_1) \in SE(AF_1)$ and $(Y_3, Y_3) \in SE(AF_2)$, i.e., AF_1 and AF_2 are both satisfiable. Similarly if $(Y_3, Y_1) \in Ext(AF_1, AF_2)$. $\square$

(LS5') If $AF_1 \equiv_s AF_2$ and $AF_3 \equiv_s AF_4$ then $\nabla(AF_1, AF_3) \equiv_s \nabla(AF_2, AF_4)$.

Proof. Assume that $SE(AF_1) = SE(AF_2)$ and $SE(AF_3) = SE(AF_4)$. Then, $Ext(AF_1) = Ext(AF_2)$ and $Ext(AF_3) = Ext(AF_4)$. If $(X_1, Y_1) \in SE(\nabla(AF_1, AF_3))$ with $X_1 = Y_1$, then $Y_1 \in Min_a(Ext(AF_1, AF_3))$. Thus, $Y_1 \in Min_a(Ext(AF_2, AF_4))$, i.e., $(Y_1, Y_1) \in SE(\nabla(AF_2, AF_4))$. If $X_1 \neq Y_1$, then $(X_1, Y_1) \in \cup Min_a(SE(AF_1, AF_3))$ and $(X_1, Y_1) \in \cup Min_a(SE(AF_2, AF_4))$. Thus, $(X_1, Y_1) \in SE(\nabla(AF_2, AF_4))$. The converse is analogous. $\square$

(LS6')$\nabla(AF_1, AF_2 \sqcup AF_3) \equiv_s \nabla(AF_1, AF_2)$ *or* $\nabla(AF_1, AF_2 \sqcup AF_3) \equiv_s \nabla(AF_1, AF_3)$ *or* $\nabla(AF_1, AF_2 \sqcup AF_3) \equiv_s \nabla(AF_1, AF_2) \sqcup \nabla(AF_1, AF_3)$.

Proof. We have to prove that $SE(\nabla(AF_1, AF_2 \sqcup AF_3)) = SE(\nabla(AF_1, AF_2))$ or $SE(\nabla(AF_1, AF_2 \sqcup AF_3)) = SE(\nabla(AF_1, AF_3))$ or $SE(\nabla(AF_1, AF_2 \sqcup AF_3)) = SE(\nabla(AF_1, AF_2)) \cup SE(\nabla(AF_1, AF_2))$.
$Ext(AF_1, AF_2 \cup AF_3) = Ext(AF_1) \times Ext(AF_2) \cup Ext(AF_1) \times Ext(AF_3)$ and $SE(AF_1, AF_2 \sqcup AF_3) = SE(AF_1) \times SE(AF_2) \cup SE(AF_1) \times SE(AF_3)$.
We have three cases:
1) $Min_a(Ext(AF_1) \times Ext(AF_2) \cup Ext(AF_1) \times Ext(AF_3)) \subseteq Ext(AF_1) \times Ext(AF_2)$ and $Min_a(Ext(AF_1) \times Ext(AF_2) \cup Ext(AF_1) \times Ext(AF_3)) \cap Ext(AF_1) \times Ext(AF_3) = \emptyset$.
Then, $Min_a(SE(AF_1) \times SE(AF_2) \cup SE(AF_1) \times SE(AF_3)) \subseteq SE(AF_1) \times SE(AF_2)$ and $Min_a(SE(AF_1) \times SE(AF_2) \cup SE(AF_1) \times SE(AF_3)) \cap SE(AF_1) \times SE(AF_3) = \emptyset$.
Thus, $SE(\nabla(AF_1, AF_2 \sqcup AF_3)) = SE(\nabla(AF_1, AF_2))$.
2) Analogous to 1).
3) $Min_a(Ext(AF_1) \times Ext(AF_2) \cup Ext(AF_1) \times Ext(AF_3)) \cap Ext(AF_1) \times Ext(AF_2) \neq \emptyset$ and $Min_a(Ext(AF_1) \times Ext(AF_2) \cup Ext(AF_1) \times Ext(AF_3)) \cap Ext(AF_1) \times Ext(AF_3) \neq \emptyset$.
Then, $Min_a(SE(AF_1) \times SE(AF_2) \cup SE(AF_1) \times SE(AF_3)) \cap SE(AF_1) \times SE(AF_2) \neq \emptyset$ and $Min_a(SE(AF_1) \times SE(AF_2) \cup SE(AF_1) \times SE(AF_3)) \cap SE(AF_1) \times SE(AF_3) \neq \emptyset$.
Thus, $SE(\nabla(AF_1, AF_2 \sqcup AF_3)) = SE(\nabla(AF_1, AF_2)) \cup SE(\nabla(AF_1, AF_3))$. $\square$

(LS7')$\nabla(AF_1, AF_2) \models_s AF_1 \sqcup AF_2$.

Proof. Assume that $(X_1, Y_1) \in SE(\nabla(AF_1, AF_2))$. Then, $(Y_1, Y_1) \in Ext(AF_1) \cup Ext(AF_2)$. If $X_1 = Y_1$, then $Y_1 \in SE(AF_1) \cup SE(AF_2)$. If $X_1 \neq Y_1$, then $(X_1, Y_1) \in \cup Min_a(SE(AF_1, AF_2))$.It follows that $(X_1, Y_1) \in SE(AF_1) \cup SE(AF_2)$. $\square$

(LS8') If AF_1 and AF_2 are satisfiable, then $AF_1 \sqcap (\nabla(AF_1, AF_2))$ is satisfiable.

Proof. Assume that AF_1 and AF_2 are satisfiable. It follows that $SE(AF_1, AF_2) \neq \emptyset$ and also $Min_a(SE(AF_1, AF_2)) \neq \emptyset$. Let $(S_1, S_2) \in Min_a(SE(AF_1, AF_2))$ and then $S_1 \in SE(\nabla(AF_1, AF_2))$. $S_1 \in SE(AF_1)$ because $(S_1, S_2) \in Min_a(SE(AF_1, AF_2))$. As $S_1 \in SE(AF_1)$ and $S_1 \in SE(\nabla(AF_1, AF_2))$ we have $S_1 \in SE(AF_1) \cap SE(\nabla(AF_1, AF_2))$. Hence, $S_1 \in SE(AF_1 \sqcap \nabla(AF_1, AF_2))$. Thus, $AF_1 \sqcap \nabla(AF_1, AF_2)$ is satisfiable. $\square$

V. Obtaining the Merged Argumentation Framework

In the previous section, we showed how to obtain the SE-extensions of the argumentation framework resulting from the arbitration merging operator we defined; namely, we show how to determine $SE(\nabla(\psi))$. Now we will show a procedure to generate an argumentation framework whose SE-extensions are $SE(\nabla(\psi))$, i.e., this procedure will have as output the argumentation framework resulting from our arbitration merging operator: $\nabla(\psi)$. We start with some auxiliary definitions:

Definition 14. *Let S be the set of all SE-extensions of an argumentation framework $AF = (Ar, att)$. We define $Min(S) = \{(X,Y) \in S \mid X = Ar \setminus Y^+\}$.*

Note that given a set S of all SE-extensions of AF, if $(X,Y) \in Min(S)$, there is no SE-extension (X',Y) of AF such that $X' \subset X$, i.e., X is the set of all arguments in Ar not attacked by Y. Then, we define $NotAtt(S)$ to collect all pair (A,B) such that A does not attack B according to S:

Definition 15. *Let S be the set of all SE-extensions of an argumentation framework $AF = (Ar, att)$. We define $NotAtt(S) = \{(A,B) \mid$ there exists $(X,Y) \in Min(S)$, such that $A \in Y$ and $B \in X\}$*

In the sequel, we will build an argumentation framework resulting from the arbitration merging operator:

Definition 16 (Building $\nabla(\psi)$)**.** *Let $\Psi = \langle AF_1, \ldots, AF_n \rangle$ be an argumentation profile, in which $AF_i = (Ar_i, att_i)$, $1 \leq i \leq n$. We define the argumentation framework $\nabla(\psi) = (Ar, att)$ such that $Ar = Ar_1 \cup \cdots \cup Ar_n$ and $att = \{(A,B) \mid A, B \in Ar$ and $(A,B) \notin NotAtt(SE(\nabla(\psi)))\}$.*

Recalling $\Psi = \langle AF_1, AF_2 \rangle$ from Example 3, we will show how to obtain $\nabla(\psi)$. Firstly, Let S be the set $SE(\nabla(\psi))$ of all SE-extensions of $\nabla(\psi)$ (see (1)). In order to determine $NotAtt(S)$, we have to obtain the set $Min(S)$:

$$(\{b\},\{a,b,c,d\}), (\{b\},\{b,c\}), (\{b\},\{a,b,c\}), \\ (\{b,d\},\{b,d\}), (\{b\},\{b,c,d\}), (\{b,d\},\{a,b,d\})$$

Hence, $NotAtt(S)$ is

$$\{(a,b),(a,d),(b,b),(b,d),(c,b),(d,b),(d,d)\}.$$

Then the resulting framework is $\nabla(\psi) = (Ar, att)$ such that $Ar = \{a,b,c,d\}$ and att is the set

$$\{(a,a),(a,c),(b,a),(b,c),(c,a),(c,c),(c,d),(d,a),(d,c)\}$$

VI. Conclusion

We have defined an arbitration merging operator ∇ for argumentation frameworks. Following a semantic approach, we have showed how to obtain all of what we called SE-extensions of a merged argumentation framework. Then, from these extensions, we have presented a procedure to generate an argumentation framework whose SE-extensions are those obtained by ∇, i.e., this procedure has as output the argumentation framework resulting from ∇. In addition, we have shown that our proposal satisfies all the postulates characterizing any reasonable arbitration merging operator.

Our work follows the approach presented in [1] to define an arbitration merging operator for logic programs. Nonetheless, in the field of Argumentation Theory, as far as we known, it is the first time that an arbitration operator is introduced under a semantic perspective.

As a future work, we want to compare our operator with other previously defined merging operators such as majority and define merging operators based on other kinds of extension, including grounded and semi-stable semantics.

Acknowledgment

This research is supported by CNPq (Universal 2012 - Proc. n° 473110/2012-1), and CNPq Casadinho/PROCAD (Project n° 552578/2011-8).

References

[1] J. Delgrande, T. Schaub, H. Tompits, and S. Woltran, "Merging logic programs under answer set semantics," in *Proceedings of the 25th International Conference on Logic Programming (ICLP)*, ser. Lecture Notes in Computer Science, P. M. Hill and D. S. Warren, Eds., vol. 5649. Springer, 2009, pp. 160–174.

[2] P. Dung, "On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games," *Artificial Intelligence*, vol. 77, pp. 321–357, 1995.

[3] T. J. M. Bench-Capon and P. E. Dunne, "Argumentation in artificial intelligence," *Artificial Intelligence*, vol. 171, no. 10–15, pp. 619–641, 2007.

[4] I. Rahwan and P. McBurney, "Guest editors' introduction: Argumentation technology," *IEEE Intelligent Systems*, vol. 22, no. 6, pp. 21–23, 2007.

[5] J. Pollock, "How to reason defeasibly," *Artificial Intelligence*, vol. 57, no. 1, pp. 1–42, 1992.

[6] ——, *Cognitive Carpentry: A Manual for How to Build a Person*. Cambridge, Massachusetts: The MIT Press, 1995.

[7] G. Vreeswijk, "Studies in defeasible argumentation," Ph.D. dissertation, Free University of Amsterdam, 1993.

[8] ——, "Abstract argumentation systems," *Artificial Intelligence*, vol. 90, no. 1-2, pp. 225–279, 1997.

[9] Simari and Loui, "A mathematical treatment of defeasible reasoning and its implementation," *Artificial Intelligence*, vol. 53, pp. 125–157, 1992.

[10] P. Liberatore and M. Schaerf, "Arbitration (or how to merge knowledge bases)," *IEEE Transactions on Knowledge and Data Engineering*, vol. 10, no. 1, pp. 76–90, 1998.

[11] C. Cayrol and M.-C. Lagasquie-Schiex, "Weighted argumentation systems: A tool for merging argumentation systems," in *Tools with Artificial Intelligence (ICTAI), 2011 23rd IEEE International Conference on*. IEEE, 2011, pp. 629–632.

[12] S. Coste-Marquis, C. Devred, S. Konieczny, M.-C. Lagasquie-Schiex, and P. Marquis, "On the merging of Dung's argumentation systems," *Artificial Intelligence*, vol. 171, no. 10, pp. 730–753, 2007.

[13] ——, "Merging argumentation systems," in *Proceedings of the National Conference on Artificial Intelligence*, vol. 20, no. 2. Menlo Park, CA; Cambridge, MA; London; AAAI Press; MIT Press; 1999, 2005, p. 614.

[14] M. A. Falappa, G. Kern-Isberner, and G. R. Simari, "Belief revision and argumentation theory," in *Argumentation in artificial intelligence*. Springer, 2009, pp. 341–360.

[15] P. Z. Revesz, "On the semantics of theory change: Arbitration between old and new information," in *Proc. ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems*, 1993, p. 71.

[16] S. Konieczny and R. P. Perez, "Merging information under constraints: A logical framework," *Journal of Logic and Computation*, vol. 12, no. 5, pp. 773–808, 2002.