Open Archive Toulouse Archive Ouverte

OATAO is an open access repository that collects the work of Toulouse researchers and makes it freely available over the web where possible

This is an author's version published in:
http://oatao.univ-toulouse.fr/22583

Official URL

DOI : https://doi.org/10.1109/ICTAI.2018.00106

To cite this version: Cayrol, Claudette and Lagasquie-Schiex, Marie-Christine *Logical Encoding of Argumentation Frameworks with Higher-order Attacks.* (2018) In: 30th International Conference on Tools with Artificial Intelligence (ICTAI 2018), 5 November 2018 - 7 November 2018 (Volos, Greece).

Any correspondence concerning this service should be sent
to the repository administrator: tech-oatao@listes-diff.inp-toulouse.fr

Logical Encoding of Argumentation Frameworks with Higher-order Attacks

Claudette Cayrol
IRIT
Université de Toulouse, CNRS
Toulouse, France
ccayrol@irit.fr

Marie-Christine Lagasquie-Schiex
IRIT
Université de Toulouse, CNRS
Toulouse, France
lagasq@irit.fr

***Abstract*—We propose a logical encoding of extended abstract argumentation frameworks, that is frameworks with higher-order attacks (*i.e.* attacks whose targets are other attacks). Our purpose is to separate the logical expression of the meaning of an attack (simple or higher-order) from the logical expression of acceptability semantics. We consider semantics which specify the conditions under which the arguments (resp. the attacks) are considered as accepted, directly on the extended framework, without translating the original framework into a Dung's argumentation framework. We characterize the output of a given framework in logical terms (namely as particular models of a logical theory). Our proposal applies to the particular case of Dung's frameworks, enabling to recover standard extensions.**

***Index Terms*—abstract argumentation, higher-order attacks, logical theory**

I. Introduction

Formal argumentation has become an essential paradigm in Artificial Intelligence, *e.g.* for reasoning from incomplete and/or contradictory information or for modeling the interactions between agents [1]. Formal abstract frameworks have greatly eased the modeling and study of argumentation. The original Dung's argumentation framework (AF) [2] consists of a collection of *arguments* interacting with each other through a relation reflecting conflicts between them, called *attack*, and enables to determine *acceptable* sets of arguments called *extensions*.

AF have been extended along different lines, *e.g.* by enriching them with positive interactions between arguments (usually expressed by a support relation), or higher-order attacks (*i.e.* attacks whose targets are other attacks). The idea of encompassing attacks to attacks in abstract argumentation frameworks has been first considered in [3] in the context of an extended framework handling argument strengths and their propagation. Then, higher-order attacks have been considered for representing preferences between arguments (second-order attacks in [4]), or for modeling situations where an attack might be defeated by an argument, without contesting the acceptability of the source of the attack [5]. Attacks to attacks and supports have been first considered in [6] with higher level networks, then in [7]; and more generally, [8] proposes an Attack-Support Argumentation Framework which allows for nested attacks and supports, *i.e.* attacks and supports whose targets can be other attacks or supports, at any level. Here is an example of higher-order attack in the legal field.

Example 1: The lawyer says that the defendant did not have intention to kill the victim (Argument b). The prosecutor says that the defendant threw a sharp knife towards the victim (Argument a). So, there is an attack from a to b. And the intention to kill should be inferred. Then the lawyer says that the defendant was in a habit of throwing the knife at his wife's foot once drunk. This latter argument (Argument c) is better considered attacking the attack from a to b, than Argument a itself. Now the prosecutor's argumentation seems no longer sufficient for proving the intention to kill.

A natural idea that has proven useful to define semantics for these frameworks, known as "flattening technique", consists in turning the original extended framework into an AF by introducing meta-arguments and a new simple (first-order) attack relation involving these meta-arguments [5], [8], [9]. More recently, alternative acceptability semantics have been defined in a direct way for argumentation frameworks with higher-order attacks [10] or for higher-order attacks and supports [11]. The idea is to specify the conditions under which the arguments (resp. the interactions) are considered as accepted directly on the extended framework, without translating the original framework into an AF.

In this paper, we propose a logical encoding of argumentation frameworks with higher-order attacks. Our purpose is (1) to characterize in a logical way the meaning of an attack (simple or higher-order) (2) to encode the acceptance conditions for arguments and attacks proposed in [10] and (3) to characterize the outputs of the framework in logical terms, thus enabling to use logical tools for computational issues.

The connection between abstract argumentation and logics goes back to the seminal work of Dung, where a translation from an AF to a logic program was given. This line of research has been pursued with other kinds of translation *e.g.* in [12]. Other works have encoded acceptance conditions as logical formulae of a first-order theory *e.g.* [13], or defined a logical language for expressing the dynamics of a framework *e.g.* [14]. To the best of our knowledge, all these works consider only attacks between two arguments. In [15] a representation of a standard abstract argumentation network is provided by adding

DOI 10.1109/ICTAI.2018.00106

strong negation to classical logic. Then, networks with higher level attacks can be represented via a translation of the original network into a standard one by the addition of nodes.

The paper is organized as follows: the necessary background is given in Sect. II; the logical encoding is presented in Sect. III and IV; some related works are discussed in Sect. V and Sect. VI concludes the paper. Due to lack of space, the proofs are omitted but are available in a technical report [16].

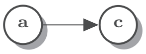
II. Background

A. The Standard Abstract Framework

The standard case handles only one kind of interaction: attacks between arguments.

Definition 1: [2] A *Dung's argumentation framework (AF)* is a tuple $\langle \mathbf{A}, \mathbf{R} \rangle$, where $\mathbf{A}$ is a finite and non-empty set of *arguments* and $\mathbf{R} \subseteq \mathbf{A} \times \mathbf{A}$ is a binary *attack relation* on the arguments, with $(a, b) \in \mathbf{R}$ indicates that a attacks b.

A graphical representation can be used for an AF: an attack $(a, c) \in \mathbf{R}$ is represented by two nodes a, c and an edge from a to c:



We recall the definitions[1] of some well-known extension-based semantics. Such a semantics specifies the requirements that a set of arguments should satisfy. The basic requirements are the following ones:

- An extension can "stand together". This corresponds to the conflict-freeness principle.
- An extension can "stand on its own", namely is able to counter all the attacks it receives. This corresponds to the defence principle.
- Reinstatement is a kind of dual principle. An attacked argument which is defended by an extension is reinstated by the extension and should belong to it.

Definition 2: [2] Let $AF = \langle \mathbf{A}, \mathbf{R} \rangle$ and $S \subseteq \mathbf{A}$.

- S is *conflict-free* iff $(a, b) \notin \mathbf{R}$ for all $a, b \in S$.
- $a \in \mathbf{A}$ is *acceptable* wrt S (or equivalently S *defends* a) iff for each $b \in \mathbf{A}$ with $(b, a) \in \mathbf{R}$, there is $c \in S$ with $(c, b) \in \mathbf{R}$.
- The *characteristic function* of AF is defined by: $\mathcal{F}(S) = \{a \in \mathbf{A}$ such that a is acceptable wrt $S\}$.
- S is *admissible* iff S is conflict-free and $S \subseteq \mathcal{F}(S)$.
- S is *a complete extension* of AF iff it is conflict-free and a fixed point of $\mathcal{F}$.
- S is *the grounded extension* of AF iff it is the minimal (wrt $\subseteq$) fixed point[2] of $\mathcal{F}$.
- S is *a preferred extension* of AF iff it is a maximal (wrt $\subseteq$) complete extension.
- S is *a stable extension* of AF iff it is conflict-free and for each $a \notin S$, there is $b \in S$ with $(b, a) \in \mathbf{R}$.

[1]Where "iff" (resp. "wrt") stands for "if and only if" (resp. "with respect to").

[2]It can be proved that the minimal fixed point of $\mathcal{F}$ is conflict-free.

Note that the complete (resp. grounded, preferred, stable) semantics satisfies the conflict-freeness, defence and reinstatement principles.

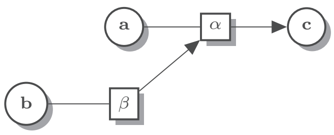
B. A Framework with Higher-Order Attacks

We consider a framework that allows representing both simple and higher-order attacks, *i.e.* attacks from an argument to either another argument or another attack. Such a framework has been usually called "recursive argumentation framework" in literature. So we keep this latter expression, even it is not completely satisfactory.

Definition 3: [10] A *recursive argumentation framework (RAF)* is a tuple $\langle \mathbf{A}, \mathbf{R}, s, t \rangle$ where $\mathbf{A}$ is a finite and non-empty set of arguments, $\mathbf{R}$ is a finite set disjunct from $\mathbf{A}$ representing attack names, s is a function from $\mathbf{R}$ to $\mathbf{A}$ mapping each interaction to its source, and t is a function from $\mathbf{R}$ to $(\mathbf{A} \cup \mathbf{R})$ mapping each interaction to its target.

Note that an AF can be viewed as a particular RAF with t being a mapping from $\mathbf{R}$ to $\mathbf{A}$.

A RAF can also be graphically represented: an attack named α (with $s(\alpha) = a$ and $t(\alpha) = c \in \mathbf{A}$) being the target of an attack β with $s(\beta) = b$ is represented by:



(arguments are in a circle and attack names are in a square)

Acceptability semantics for argumentation frameworks with higher-order attacks have been defined in a direct way in [10]. The idea is to specify the conditions under which the arguments are considered as accepted directly on the extended framework, without translating the original framework into an AF. Moreover, due to the defeasible nature of attacks (attacks may be affected by other attacks), conditions under which the attacks are accepted must also be specified. Indeed, some attacks may not be "valid", in the sense that they cannot defeat the argument or attack they are targeting. So, acceptability conditions for arguments should be given with respect to valid attacks and conversely attacks should be declared valid with respect to other arguments or attacks. For instance, the fact that two arguments may be conflicting depends on the validity of the attack between them. Hence, the notion of extension (set of arguments) is replaced by a pair of a set of arguments and a set of attacks, called a "structure".

Definition 4: [10] Consider $RAF = \langle \mathbf{A}, \mathbf{R}, s, t \rangle$. A *structure of* RAF is a pair (S, Γ) with $S \subseteq \mathbf{A}$ and $\Gamma \subseteq \mathbf{R}$.

Intuitively, given a structure $U = (S, \Gamma)$, S contains the arguments that are accepted "owing to" U and Γ contains the attacks which are valid "owing to" U (the meaning of "owing to" depending on the considered semantics).

In the following, we recall the acceptability conditions for structures, and the definitions of the semantics that are given in [10]. The key notion is the fact that a set of arguments (resp. attacks) can be "defeated" (resp. "inhibited") wrt a given structure.

Definition 5: [10] Consider $RAF = \langle \mathbf{A}, \mathbf{R}, s, t\rangle$. Given $U = (S, \Gamma)$ a structure of RAF. Let $a \in \mathbf{A}$ and $\alpha \in \mathbf{R}$.

- a is *defeated wrt* U iff there is $\beta \in \Gamma$ with $s(\beta) \in S$ and $t(\beta) = a$,
- α is *inhibited wrt* U iff there is $\beta \in \Gamma$ with $s(\beta) \in S$ and $t(\beta) = \alpha$.

$Def(U)$ (resp. $Inh(U)$) denotes the set of arguments (resp. attacks) that are defeated (resp. inhibited) wrt U.

1) Conflict-free structures: The minimal requirement for a structure (S, Γ) is that two arguments of S cannot be related by an attack of the structure, and similarly there cannot be an attack grounded in S and whose target is an element of Γ. Formally:

Definition 6: [10] Consider $RAF = \langle \mathbf{A}, \mathbf{R}, s, t\rangle$. A structure $U = (S, \Gamma)$ of RAF is *conflict-free* iff $S \cap Def(U) = \varnothing$ and $\Gamma \cap Inh(U) = \varnothing$.

2) Admissible structures: Acceptability (for an argument or an attack) is also relative to a structure.

Definition 7: [10] Consider $RAF = \langle \mathbf{A}, \mathbf{R}, s, t\rangle$. Given a structure $U = (S, \Gamma)$ of RAF. Let $a \in \mathbf{A}$ and $\alpha \in \mathbf{R}$.

- a (resp. α) is *acceptable wrt* U iff for each $\beta \in \mathbf{R}$ with $t(\beta) = a$ (resp. $t(\beta) = \alpha$), either $\beta \in Inh(U)$ or $s(\beta) \in Def(U)$.
- U is *admissible* iff it is conflict-free and for each $x \in (S \cup \Gamma)$, x is acceptable wrt U.

$Acc(U)$ denotes the set containing all acceptable arguments and attacks wrt U.

Remark: Let $\alpha \in \mathbf{R}$ with $t(\alpha) = b \in \mathbf{A}$ (resp. $t(\alpha) = \beta \in \mathbf{R}$). If α and $s(\alpha)$ are unattacked, there is no admissible structure $U = (S, \Gamma)$ such that $b \in S$ (resp. $\beta \in \Gamma$).

3) Complete, stable, preferred and grounded structures: For any pair of structures $U = (S, \Gamma)$ and $U' = (S', \Gamma')$, $U \subseteq U'$ means that $(S \cup \Gamma) \subseteq (S' \cup \Gamma')$. The structure U is $\subseteq$-maximal iff every structure U' that satisfies $U \subseteq U'$ also satisfies $U' \subseteq U$. Similarly, U is $\subseteq$-minimal iff every structure U' that satisfies $U' \subseteq U$ also satisfies $U \subseteq U'$.

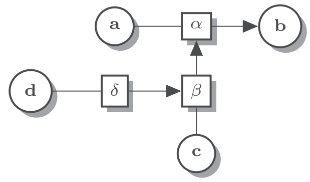
Definition 8: [10] Consider $RAF = \langle \mathbf{A}, \mathbf{R}, s, t\rangle$. A structure $U = (S, \Gamma)$ of RAF is:

- *complete* iff it is conflict-free and $Acc(U) = S \cup \Gamma$.
- *stable* iff it is conflict-free and satisfies $\mathbf{A} \setminus S \subseteq Def(U)$ and $\mathbf{R} \setminus \Gamma \subseteq Inh(U)$.
- *preferred* iff it is a $\subseteq$-maximal admissible structure.
- *grounded* iff it is the $\subseteq$-minimal conflict-free structure $U = (S, \Gamma)$ satisfying $Acc(U) \subseteq S \cup \Gamma$.

It has been proved in [10] that usual properties of Dung's extensions also hold for structures:

- A complete structure contains all the unattacked arguments and all the unattacked attacks.
- Every complete structure is admissible, every preferred structure is also complete and every stable structure is also preferred.
- The grounded structure is the $\subseteq$-minimal complete structure. It is unique.

Example 2: Consider the RAF depicted by the following figure:



There is only one complete (resp. preferred, stable, grounded structure): $(\{a, c, d\}, \{\alpha, \delta\})$.

4) D-structures: The notion of structure has been strengthened in order to obtain a conservative generalization of Dung's frameworks for the conflict-free, admissible, complete, stable and preferred semantics. It is worth to note that in an AF, each attack is considered as valid, in the sense that it may affect its target. The next definition strengthens the notion of structure by adding a condition on attacks that will force every acceptable attack to be valid.

Definition 9: [10] Given $RAF = \langle \mathbf{A}, \mathbf{R}, s, t\rangle$.

1) A *d-structure on* RAF is a structure $U = (S, \Gamma)$ such that $(Acc(U) \cap \mathbf{R}) \subseteq \Gamma$.
2) A *conflict-free (resp. admissible, complete, preferred, stable) d-structure* is a conflict-free (resp. admissible, complete, preferred, stable) structure which is also a d-structure.

It follows from Def. 8 that every complete (resp. stable, preferred) structure of a RAF is a d-structure of this RAF. However it is not the case for admissible and conflict-free structures.

The conservative generalization proved in [10] relies upon a correspondence between a Dung's framework (and its extensions) and a "nonrecursive" RAF (and its d-structures), where a nonrecursive RAF is a RAF in which no attack targets another attack.

III. Logical Description of a RAF

We propose a logical description of a RAF, that allows an explicit representation of arguments, attacks and their properties (accepted argument, attacked argument, valid attack, ...). We have been inspired by works in bioinformatics (see [17], [18]), where *metabolic networks* are used to describe the chemical reactions of cells; these reactions can be negative (inhibition of a protein) or positive (production of a new protein) and they can depend on other proteins or other reactions. A translation from metabolic networks to classical logic has been proposed in [18], which allows for the use of automated deduction methods for reasoning on these networks.

Given RAF a recursive argumentation framework, $\Sigma(RAF)$ will denote the set of first-order logic formulae describing RAF.

A. Vocabulary

The following unary predicate symbols are used: Acc, $NAcc$, Val, $Attack$, Arg and the following unary functions symbols : T, S, with the following meaning:

- $Acc(x)$ (resp. $NAcc(x)$) means "x is accepted" (resp. "x cannot be accepted"), when x denotes an argument
- $Val(\alpha)$ means "α is valid" when α denotes an attack
- $Attack(x)$ means "x is an attack"
- $Arg(x)$ means "x is an argument"
- $T(x)$ (resp. $S(x)$) denotes the target (resp. source) of x, when x denotes an attack

The binary equality predicate is also used. Note that the quantifiers $\exists$ and $\forall$ range over some domain D. To restrict them to subsets of D, bounded quantifiers will be used:

$\forall x \in E\ (P(x))$ means $\forall x\ (x \in E \rightarrow P(x))$ or equivalently $\forall x (E(x) \rightarrow P(x))$.

So we will use:

- $\forall x \in Attack\ (\Phi(x))$ (resp. $\exists x \in Attack\ (\Phi(x))$)
- and $\forall x \in Arg\ (\Phi(x))$ (resp. $\exists x \in Arg\ (\Phi(x))$).

Note that the meaning of $NAcc(x)$ is not "x is not accepted" but rather "x cannot be accepted" (for instance because x is the target of a valid attack whose source is accepted). Hence, $NAcc(x)$ is not logically equivalent to $\neg Acc(x)$. However, the logical theory will enable to deduce $\neg Acc(x)$ from $NAcc(x)$, as shown below.

B. Logical theory

Two kinds of formulae describe RAF:

- the formulae describing the general behaviour of an attack, possibly recursive, in an argumentation framework, *i.e.* how an attack interacts with arguments and other attacks related to it.
- and the formulae encoding the specificities of the current framework.

The meaning of an attack is described under the form of constraints on its source (an argument) and its target (an argument or an attack). Moreover, as attacks may be attacked by other attacks, some attacks may not be valid.

- If an attack from an argument to an attack is valid, then if its source is accepted, its target *is not* valid.
- If an attack between two arguments is valid and if its source is accepted, then its target *cannot be* accepted. In that case, the target *is not* accepted.

Using the vocabulary defined above, these constraints can be expressed by the following formulae:

(1) $\forall x \in Attack\ (\forall y \in Attack\ ((Val(y) \wedge (T(y) = x) \wedge Acc(S(y))) \rightarrow \neg Val(x)\ ))$

(2) $\forall x \in Arg\ (\forall y \in Attack\ ((Val(y) \wedge (T(y) = x) \wedge Acc(S(y))) \rightarrow NAcc(x)\ ))$

(3) $\forall x \in Arg\ (NAcc(x) \rightarrow \neg Acc(x))$

Two other formulae limit the domain to arguments and attacks.

(4) $\forall x\ (Attack(x) \rightarrow \neg Arg(x))$

(5) $\forall x\ (Arg(x) \vee Attack(x))$

Note that we assume that the argumentation framework is finite, with $\mathbf{A} = \{a_1, \ldots a_n\}$ and $\mathbf{R} = \{\alpha_1, \ldots, \alpha_m\}$. Then, the logical encoding of specificities of the RAF leads to the following set of formulas:

(6) $(S(\alpha) = a) \wedge (T(\alpha) = b)$ for all $\alpha \in \mathbf{R}$ with $s(\alpha) = a$ and $t(\alpha) = b$

(7) $\forall x\ (Arg(x) \leftrightarrow (x = a_1) \vee \ldots \vee (x = a_n))$

(8 $\forall x\ (Attack(x) \leftrightarrow (x = \alpha_1) \vee \ldots \vee (x = \alpha_m))$

(9) $a_i \neq a_j$ for all $a_i, a_j \in \mathbf{A}$ with $i \neq j$

(10) $\alpha_i \neq \alpha_j$ for all $\alpha_i, \alpha_j \in \mathbf{R}$ with $i \neq j$

In the following, we will write s_α (resp. t_α) in place of $S(\alpha)$ (resp. $T(\alpha)$) for simplicity.

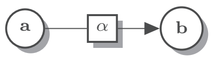
The logical theory $\Sigma(RAF)$ corresponding to RAF consists of the above 10 formulae. It is obviously consistent.

Example 2 (cont'd): Using the equality axioms, a simplified version of $\Sigma(RAF)$ can be obtained (in particular tautologies are omitted):[3]

$$\begin{aligned}\Sigma(RAF) = \{ & (Val(\beta) \wedge Acc(c)) \rightarrow \neg Val(\alpha) \text{ (from (1))},\\ & (Val(\delta) \wedge Acc(d)) \rightarrow \neg Val(\beta) \text{ (from (1))},\\ & (Val(\alpha) \wedge Acc(a)) \rightarrow NAcc(b) \text{ (from (2))},\\ & NAcc(b) \rightarrow \neg Acc(b) \text{ (from (3))},\\ & NAcc(a) \rightarrow \neg Acc(a) \text{ (from (3))},\\ & NAcc(c) \rightarrow \neg Acc(c) \text{ (from (3))},\\ & NAcc(d) \rightarrow \neg Acc(d) \text{ (from (3))}\}\end{aligned}$$

In the particular case of a nonrecursive RAF, formula (1) is a tautology. However, formula (2) cannot be simplified as it cannot be deduced that the attacks are valid. Indeed, the logical theory $\Sigma(RAF)$ only captures the description of RAF and is not concerned with the semantics of the framework (the logical description of the semantics is handled in the next section).

Example 3: Consider the nonrecursive RAF depicted by the following figure:



$\Sigma(RAF)$ enables to deduce the formula $(Val(\alpha) \wedge Acc(a)) \rightarrow NAcc(b)$. Note that if $Val(\alpha)$ is assumed, we obtain $Acc(a) \rightarrow NAcc(b)$ and from $NAcc(b) \rightarrow \neg Acc(b)$ it follows that $Acc(a) \rightarrow \neg Acc(b)$ and so $Acc(b) \rightarrow \neg Acc(a)$. However, it cannot be deduced that $Acc(b) \rightarrow NAcc(a)$. Indeed, the predicate $NAcc$ allows for the representation of the direction of an attack between two arguments and avoids the contraposition of the attack.

IV. Logical Formalization of Semantics

A. Logical Encoding of Semantics

In presence of higher-order attacks, the conflict-freeness, defence and reinstatement principles must take into account the fact that attacks might be not valid. Moreover, for each of these principles, two versions will be given, one for arguments and another one for attacks. Then, for each principle, we give

[3] The simplification will be applied for the other examples.

a logical expression, thus leading to add formulas to the base $\Sigma(RAF)$ and producing new bases.

1) Conflict-freeness: The conflict-freeness principle is formulated as follows (for arguments and for attacks):

- If there is a valid attack between two arguments, they cannot be jointly accepted.
- If there is an attack from an accepted argument to an attack, these attacks cannot be both valid.

Note that these properties are already expressed in $\Sigma(RAF)$ by the formulae (1), (2), (3).

2) Defence: The idea is to claim that an argument a is defended by a set of arguments S if S weakens each attack α to a, either by attacking the source of α, or by attacking α itself. Moreover the defence should be obtained with valid attacks. So, the defence principle is formulated as follows (for arguments and for attacks):

- An attacked argument may be accepted only if for each attack to it, either the source or the attack itself is in turn attacked by a valid attack from an accepted argument.
- An attack may be valid only if for each attack to it, either the source or the attack itself is in turn attacked by a valid attack from an accepted argument.

These properties are expressed by the following formulae:

(11) $\forall\alpha \in Attack$ (
$Acc(t_\alpha)$
$\rightarrow (\exists\beta \in Attack$
$(t_\beta \in \{s_\alpha, \alpha\}^4 \wedge Val(\beta) \wedge Acc(s_\beta)))$)

(12) $\forall\alpha \in Attack$ ($\forall\delta \in Attack$ (
$((\delta = t_\alpha) \wedge Val(\delta))$
$\rightarrow (\exists\beta \in Attack$
$(t_\beta \in \{s_\alpha, \alpha\} \wedge Val(\beta) \wedge Acc(s_\beta)))$))

These formulae are added to the base $\Sigma(RAF)$, thus producing the base $\Sigma_d(RAF)$.

3) Reinstatement: Based on the previous notion of defence, the reinstatement principle is formulated as follows (for arguments and for attacks):

- An argument must be accepted provided that, for each attack to it, the source or the attack itself is in turn attacked by a valid attack from an accepted argument.
- An attack may be valid provided that for each attack to it, either the source or the attack itself is in turn attacked by a valid attack from an accepted argument.

These properties are expressed by the following formulae:

(13) $\forall c \in Arg$ (
($\forall\alpha \in Attack$ $(t_\alpha = c$
$\rightarrow (\exists\beta \in Attack$
$(t_\beta \in \{s_\alpha, \alpha\} \wedge Val(\beta) \wedge Acc(s_\beta)))))$
$\rightarrow Acc(c)$)

(14) $\forall\delta \in Attack$ (
($\forall\alpha \in Attack$ $(t_\alpha = \delta$
$\rightarrow (\exists\beta \in Attack$
$(t_\beta \in \{s_\alpha, \alpha\} \wedge Val(\beta) \wedge Acc(s_\beta)))))$
$\rightarrow Val(\delta)$)

[4]Strictly speaking, should be written as follows : $t_\beta = s_\alpha \vee t_\beta = \alpha$.

These formulae are added to the base $\Sigma(RAF)$, thus producing the base $\Sigma_r(RAF)$.

4) Stability: The stability requirement can be formulated as follows (one for arguments and one for attacks):

- If an argument is not accepted, it must be attacked by a valid attack from an accepted argument.
- If an attack is not valid, it must be attacked by a valid attack from an accepted argument.

These properties are expressed by the following formulae:

(15) $\forall c \in Arg$ (
$\neg Acc(c)$
$\rightarrow (\exists\beta \in Attack$
$((t_\beta = c) \wedge Val(\beta) \wedge Acc(s_\beta)))$)

(16) $\forall\alpha \in Attack$ (
$\neg Val(\alpha)$
$\rightarrow (\exists\beta \in Attack$
$((t_\beta = \alpha) \wedge Val(\beta) \wedge Acc(s_\beta))$)

These formulae are added to the base $\Sigma(RAF)$, thus producing the base $\Sigma_s(RAF)$.

Example 2 (cont'd): $\Sigma_d(RAF)$ is obtained from $\Sigma(RAF)$ by adding the formulas:
$Acc(b) \rightarrow (Val(\beta) \wedge Acc(c))$,
$\neg Val(\beta)$ and
$Val(\alpha) \rightarrow (Val(\delta) \wedge Acc(d))$.
$\Sigma_r(RAF)$ is obtained from $\Sigma(RAF)$ by adding the formulas:
$Acc(a)$,
$Acc(c)$,
$Acc(d)$,
$(Val(\beta) \wedge Acc(c)) \rightarrow Acc(b)$,
$Val(\delta)$ and
$(Val(\delta) \wedge Acc(d)) \rightarrow Val(\alpha)$.

B. Characterizing Semantics of a RAF

We propose characterizations of the structures under different semantics in terms of models of the bases $\Sigma(RAF)$, $\Sigma_d(RAF)$, $\Sigma_r(RAF)$, $\Sigma_s(RAF)$.

Let $RAF = \langle \mathbf{A}, \mathbf{R}, s, t\rangle$. Given $\mathcal{I}$ an interpretation of $\Sigma(RAF)$, we define:

- $S_\mathcal{I} = \{x \in \mathbf{A} | \mathcal{I}(Acc(x)) = \textit{true}\}$
- $\Gamma_\mathcal{I} = \{x \in \mathbf{R} | \mathcal{I}(Val(x)) = \textit{true}\}$

Moreover, let $\mathcal{I}$ be a model of $\Sigma(RAF)$:

- $\mathcal{I}$ is a $\subseteq$-maximal model of $\Sigma(RAF)$ iff there is no model $\mathcal{I}'$ of $\Sigma(RAF)$ with $(S_\mathcal{I} \cup \Gamma_\mathcal{I}) \subset (S_{\mathcal{I}'} \cup \Gamma_{\mathcal{I}'})$.
- $\mathcal{I}$ is a $\subseteq$-minimal model of $\Sigma(RAF)$ iff there is no model $\mathcal{I}'$ of $\Sigma(RAF)$ with $(S_{\mathcal{I}'} \cup \Gamma_{\mathcal{I}'}) \subset (S_\mathcal{I} \cup \Gamma_\mathcal{I})$.

We have the following characterizations:

Proposition 1: Let $RAF = \langle \mathbf{A}, \mathbf{R}, s, t\rangle$. Let $U = (S, \Gamma)$ a structure on RAF.

1) U is conflict-free iff there exists $\mathcal{I}$ model of $\Sigma(RAF)$ with $S_\mathcal{I} = S$ and $\Gamma_\mathcal{I} = \Gamma$.

2) U is admissible iff there exists $\mathcal{I}$ model of $\Sigma_d(RAF)$ with $S = S_\mathcal{I}$ and $\Gamma_\mathcal{I} = \Gamma$.
3) U is complete iff there exists $\mathcal{I}$ model of $\Sigma_d(RAF) \cup \Sigma_r(RAF)$ with $S = S_\mathcal{I}$ and $\Gamma_\mathcal{I} = \Gamma$.
4) U is a stable structure iff there exists $\mathcal{I}$ model of $\Sigma_s(RAF)$ with $S_\mathcal{I} = S$ and $\Gamma_\mathcal{I} = \Gamma$.
5) U is a preferred structure iff there exists $\mathcal{I}$ $\subseteq$-maximal model of $\Sigma_d(RAF)$ with $S_\mathcal{I} = S$ and $\Gamma_\mathcal{I} = \Gamma$.
6) U is the grounded structure iff $S = S_\mathcal{I}$ and $\Gamma_\mathcal{I} = \Gamma$ where $\mathcal{I}$ is a $\subseteq$-minimal model of $\Sigma_r(RAF)$.[5]

Example 2 (cont'd): There is only one complete structure: $(\{a, c, d\}, \{\alpha, \delta\})$. Indeed, every model $\mathcal{I}$ of $\Sigma_d(RAF) \cup \Sigma_r(RAF)$ is such that $S_\mathcal{I} = \{a, c, d\}$ and $\Gamma_\mathcal{I} = \{\alpha, \delta\}$, in other words, every model of $\Sigma_d(RAF) \cup \Sigma_r(RAF)$ satisfies $Acc(a)$, $Acc(c)$, $Acc(d)$, $Val(\delta)$, $Val(\alpha)$ and falsifies $Acc(b)$, $Val(\beta)$. An example of admissible (but not complete) structure is $(\{a, d\}, \{\delta\})$. Indeed, there is a model $\mathcal{I}$ of $\Sigma_d(RAF)$ with $S_\mathcal{I} = \{a, d\}$ and $\Gamma_\mathcal{I} = \{\delta\}$.

D-structures can also be characterized. Let us recall that d-structures are particular structures in which acceptable attacks are forced to be valid. So, we consider the base $\Sigma(RAF)$ augmented with the formula that expresses the reinstatement principle for attacks, that is formula (14).

As said before, complete structures are d-structures. So we just have to complete Prop.1 with the characterizations of conflict-free and admissible d-structures.

Proposition 2: Let $RAF = \langle \mathbf{A}, \mathbf{R}, s, t\rangle$. Let $U = (S, \Gamma)$ a structure on RAF.

1) U is a conflict-free d-structure iff there exists $\mathcal{I}$ model of $\Sigma(RAF) \cup \{(14)\}$ with $S_\mathcal{I} = S$ and $\Gamma_\mathcal{I} = \Gamma$.
2) U is an admissible d-structure iff there exists $\mathcal{I}$ model of $\Sigma_d(RAF) \cup \{(14)\}$ with $S = S_\mathcal{I}$ and $\Gamma_\mathcal{I} = \Gamma$.

Example 2 (cont'd): From (14), the following formulae are obtained: $Val(\delta)$ and $(Val(\delta) \wedge Acc(d)) \rightarrow Val(\alpha)$.
An example of admissible (but not complete) d-structure is $(\{a, d\}, \{\alpha, \delta\})$. Indeed, there is a model $\mathcal{I}$ of $\Sigma_d(RAF) \cup \{(14)\}$ with $S_\mathcal{I} = \{a, d\}$ and $\Gamma_\mathcal{I} = \{\alpha, \delta\}$. Note that $(\{a, d\}, \{\delta\})$ is an admissible structure but not an admissible d-structure.

C. Case of AF

As said before, an AF can be viewed as a particular RAF. So we can consider the associated logical theory, which we denote by $\Sigma(AF)$ for simplicity. Moreover, in the particular case of an AF, the semantics recalled in SectionII-A assume that each attack is valid. As a consequence, the logical theory $\Sigma(AF)$ can be replaced by a logically equivalent theory built as follows: For each $(a, b) \in \mathbf{R}$, the attack from a to b is described by the formulae $Acc(a) \rightarrow NAcc(b)$ and $NAcc(b) \rightarrow \neg Acc(b)$.

[5]It also holds that U is the grounded structure iff $S = S_\mathcal{I}$ and $\Gamma_\mathcal{I} = \Gamma$ where $\mathcal{I}$ is a $\subseteq$-minimal model of $\Sigma_d(RAF) \cup \Sigma_r(RAF)$. Considering $\Sigma_d(RAF) \cup \Sigma_r(RAF)$ instead of $\Sigma_r(RAF)$ might be useful from a computational point of view, when searching for minimal models.

Then, the standard defence, reinstatement and stability principles are encoded with simplified versions of formulae (11), (13) and (15) (as attacks are never attacked, formulae (12), (14) and (16) would be tautologies). Let $AF = \langle \mathbf{A}, \mathbf{R}\rangle$. For $x \in \mathbf{A}$, let $\mathbf{R}^-(x)$ denote the set of its attackers. For each principle, a set of formulas is provided, one for each argument:

- *Defence:* For each $x \in \mathbf{A}$,
$$Acc(x) \rightarrow (\wedge_{y \in \mathbf{R}^-(x)}(\vee_{z \in \mathbf{R}^-(y)} Acc(z)))$$
- *Reinstatement:* For each $x \in \mathbf{A}$,
$$(\wedge_{y \in \mathbf{R}^-(x)}(\vee_{z \in \mathbf{R}^-(y)} Acc(z))) \rightarrow Acc(x)$$
- *Stability:* For each $x \in \mathbf{A}$,
$$\neg Acc(x) \rightarrow (\vee_{y \in \mathbf{R}^-(x)} Acc(y))$$

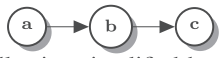
We denote by $\Sigma_d(AF)$ (resp. $\Sigma_r(AF)$, $\Sigma_s(AF)$) the logical theories obtained by adding all the formulas encoding defence (resp. reinstatement, stability) to $\Sigma(AF)$. Given $\mathcal{I}$ be an interpretation of $\Sigma(AF)$, we still denote by $S_\mathcal{I}$ the set $\{x \in \mathbf{A} | \mathcal{I}(Acc(x)) = \textit{true}\}$. If $\mathcal{I}$ is a model of $\Sigma(AF)$, $\mathcal{I}$ is said to be a $\subseteq$-maximal (resp. minimal) model of $\Sigma(AF)$ iff there is no model $\mathcal{I}'$ of $\Sigma(AF)$ such that $S_\mathcal{I} \subset S_{\mathcal{I}'}$ (resp. $S_{\mathcal{I}'} \subset S_\mathcal{I}$).

Then, we get the following characterizations:

Proposition 3: Let $AF = \langle \mathbf{A}, \mathbf{R}\rangle$. Let $S \subseteq \mathbf{A}$.

1) S is conflict-free in $\langle \mathbf{A}, \mathbf{R}\rangle$ iff there exists $\mathcal{I}$ model of $\Sigma(AF)$ with $S_\mathcal{I} = S$.
2) S is admissible in $\langle \mathbf{A}, \mathbf{R}\rangle$ iff there exists $\mathcal{I}$ model of $\Sigma_d(AF)$ with $S = S_\mathcal{I}$.
3) S is a complete extension of $\langle \mathbf{A}, \mathbf{R}\rangle$ iff there exists $\mathcal{I}$ model of $\Sigma_d(AF) \cup \Sigma_r(AF)$ with $S = S_\mathcal{I}$.
4) S is a stable extension of $\langle \mathbf{A}, \mathbf{R}\rangle$ iff there exists $\mathcal{I}$ model of $\Sigma_s(AF)$ with $S_\mathcal{I} = S$.
5) U is a preferred extension of $\langle \mathbf{A}, \mathbf{R}\rangle$ iff there exists $\mathcal{I}$ $\subseteq$-maximal model of $\Sigma_d(AF)$ with $S_\mathcal{I} = S$.
6) S is the grounded extension iff $S = S_\mathcal{I}$ where $\mathcal{I}$ is a $\subseteq$-minimal model of $\Sigma_r(AF)$.

Example 4: Consider the AF represented by:



It can be encoded by the following simplified bases:

$\Sigma(AF) = \{Acc(a) \rightarrow NAcc(b),$
$NAcc(b) \rightarrow \neg Acc(b),$
$Acc(b) \rightarrow NAcc(c),$
$NAcc(c) \rightarrow \neg Acc(c),$
$NAcc(a) \rightarrow \neg Acc(a)\}$
and $\Sigma_d(AF) = \Sigma(AF) \cup$
$\{\neg Acc(b),$
$Acc(c) \rightarrow Acc(a)\}$.

Every $\subseteq$-maximal model of $\Sigma_d(AF)$ satisfies $Acc(a)$, $Acc(c)$ and falsifies $Acc(b)$. That corresponds to the unique preferred extension $\{a, c\}$.

V. Related Works

From the seminal work of Dung [2], several works have proposed to connect abstract argumentation with logic programming (see [12] for recent work and more references). The issue is to find an appropriate encoding of an AF into a logic program P, so that applying logic programming semantics to P enables to capture argumentation semantics of the original AF. Dung [2] has proposed an encoding allowing the capture of (only) grounded and stable semantics. In [12], the encoding allows for the characterization of the standard argumentation semantics (grounded, stable, preferred and complete semantics) through 3-valued models of a logic program.

In the particular case of an AF, the logical representation of [15] using classical propositional logic augmented with strong negation is very close to our proposal. However, taking into account higher-order attacks requires a modification of the original framework with the addition of nodes and joint attacks.

The issue of logical encoding of abstract argumentation has recently been addressed for different other purposes independently of the notion of logic programming. In [13], [19] acceptance conditions and standard semantics are encoded by first-order logical formulae (given a semantics σ and a set S of arguments, a formula is provided which is satisfiable iff S is a σ-extension). However, the argumentation framework itself is not represented. A similar issue is addressed in [20] with a modal logic, considering that the accessibility relation is the inverse of the attack relation; the same kind of work is presented in [21] using signed theories and QBF formulae; [22] presents algorithms using particular logical notions (minimal correction sets, backbone) in order to compute some semantics (semi-stable and eager); [23] translates complete labellings into logical formulae in order to compute preferred extensions with SAT solvers; [24] proposes a metalevel analysis of the computation problems related to given semantics in order to automatically generate solvers adapted to these problems.

In the more general abstract dialectical framework [25], each argument is associated with a propositional formula which represents the acceptance conditions of the argument. This logical translation enables to capture easily the stable semantics. However, recursive interactions are not taken into account.

[14] proposes a complete framework for handling the dynamics on an AF. A first-order logical language is presented, enabling to describe the structure of an AF, to express incomplete knowledge on an AF and to encode change operations on an AF.

Moreover in the context of the First International Competition on Computational Models of Argumentation (ICCMA), different solvers have been proposed and tested (*e.g.* [26], or [27]). However, in all these works, neither the attack relation itself is logically encoded, nor the recursive aspects are taken into account.

VI. Conclusion

In this work, we have proposed a logical encoding of argumentation frameworks with higher-order attacks. Our proposal enables to separate the logical expression of the meaning of an attack (simple or higher-order) and the logical expression of acceptability semantics. These semantics (introduced in [10]) specify the conditions under which the arguments (resp. the attacks) are considered as accepted, directly on the extended framework, without translating the original framework into an AF.

Then, we are able to characterize the output of a given argumentation framework (under the form of structures) in logical terms (namely as particular models of a logical theory). That opens a way for computational issues by using logical tools. As a preliminary work in that direction, a software has been developed [28] that enables to represent a RAF, to express the associated logical theories $\Sigma(RAF), \Sigma_d(RAF), \ldots$, and to compute the structures under different semantics.

Another feature of our work is its conservative generalization of AF, when d-structures are considered.

Future works will include the study of a logical encoding of frameworks with higher-order attacks and higher-order supports. The difficulty is that there exist different interpretations of the notion of support in argumentation frameworks. In a first step, we plan to consider a recent framework that allows handling both higher-order attacks and higher-order evidential supports, with structure-based semantics [29].

References

[1] I. Rahwan and G. Simari, *Argumentation in Artificial Intelligence*. Springer, 2009.

[2] P. M. 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] H. Barringer, D. Gabbay, and J. Woods, "Temporal dynamics of support and attack networks : From argumentation to zoology," in *Mechanizing Mathematical Reasoning. LNAI 2605*. Springer Verlag, 2005, pp. 59–98.

[4] S. Modgil, "Reasoning about preferences in argumentation frameworks," *Artificial Intelligence*, vol. 173, pp. 901–934, 2009.

[5] P. Baroni, F. Cerutti, M. Giacomin, and G. Guida, "AFRA: Argumentation framework with recursive attacks," *International Journal of Approximate Reasoning*, vol. 52, pp. 19–37, 2011.

[6] D. M. Gabbay, "Fibring argumentation frames," *Studia Logica*, vol. 93, pp. 231–295, 2009.

[7] S. Villata, G. Boella, D. M. Gabbay, and L. van der Torre, "Modelling defeasible and prioritized support in bipolar argumentation," *Annals of Mathematics and Artificial Intelligence*, vol. 66, no. 1-4, pp. 163–197, 2012. [Online]. Available: http://dx.doi.org/10.1007/s10472-012-9317-7

[8] A. Cohen, S. Gottifredi, A. J. García, and G. R. Simari, "An approach to abstract argumentation with recursive attack and support," *Journal of Applied Logic*, vol. 13, no. 4, pp. 509–533, 2015.

[9] C. Cayrol, A. Cohen, and M.-C. Lagasquie-Schiex, "Towards a new framework for recursive interactions in abstract bipolar argumentation," in *Proceedings of International Conference on Computational Models of Argument (COMMA)*, P. Baroni, Ed. IOS Press, 2016, pp. 191–198.

[10] C. Cayrol, J. Fandinno, L. Fariñas del Cerro, and M.-C. Lagasquie-Schiex, "Valid attacks in argumentation frameworks with recursive attacks," in *Proc. of Thirteenth International Symposium on Commonsense Reasoning*, vol. 2052. CEUR Workshop Proceedings, 2017.

[11] A. Cohen, S. Gottifredi, A. J. García, and G. R. Simari, "On the acceptability semantics of argumentation frameworks with recursive attack and support," in *Proceedings of International Conference on Computational Models of Argument (COMMA)*, P. Baroni, Ed. IOS Press, 2016, pp. 231–242.

[12] M. Caminada, S. Sá, J. Alcântara, and W. Dvořák, "On the equivalence between logic programming semantics and argumentation semantics," *International Journal of Approximate Reasoning*, vol. 58, pp. 87 – 111, 2015, special Issue of the Twelfth European Conference on Symbolic and Quantitative Approaches to Reasoning with Uncertainty (ECSQARU 2013). [Online]. Available: http://dx.doi.org/10.1016/j.ijar.2014.12.004

[13] P. Besnard, S. Doutre, and A. Herzig, "Encoding argument graphs in logic," in *Proc of International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems (IPMU), No. 443 of Communications in Computer and Information Science*, A. Laurent, O. Strauss, B. Bouchon-Meunier, and R. Yager, Eds. Springer, 2014, pp. 345–354.

[14] F. Dupin de Saint-Cyr, P. Bisquert, C. Cayrol, and M.-C. Lagasquie-Schiex, "Argumentation update in YALLA (Yet Another Logic Language for Argumentation)," *International Journal of Approximate Reasoning*, vol. 75, pp. 57 – 92, 2016. [Online]. Available: http://dx.doi.org/10.1016/j.ijar.2016.04.003

[15] D. M. Gabbay and M. Gabbay, "The attack as strong negation, part I," *Logic Journal of the IGPL*, vol. 23, no. 6, pp. 881–941, 2015.

[16] C. Cayrol and M.-C. Lagasquie-Schiex, "Acceptability semantics in recursive argumentation frameworks: Logical encoding and computation," IRIT, Université Paul Sabatier, Toulouse, Rapport de recherche IRIT/RR–2018–02–FR, janvier 2018. [Online]. Available: http://www.irit.fr/publis/ADRIA/PapersMCL/Rapport-IRIT-2018-02.pdf

[17] R. Demolombe, L. Fariñas del Cerro, and N. Obeid, "Molecular Interaction Automated Maps (regular paper)," in *International Workshop on Learning and Non Monotonic Reasoning (LNMR), A Corunna, Spain, 15/09/2013*, vol. abs/1311.4639. http://arxiv.org/corr: Computing Research Repository, septembre 2013, pp. 31–42. [Online]. Available: http://oatao.univ-toulouse.fr/13253/

[18] J. M. Alliot, R. Demolombe, L. Fariñas del Cerro, M. Diéguez, and N. Obeid, "Reasoning on molecular interaction maps," in *7^{th} European Symposium on Computational Intelligence and Mathematics (ESCIM'15)*, Cádiz, Spain, 2015, pp. 263–269, available at: http://escim2015.uca.es/wp-content/uploads/2015/02/ESCIM2015-FINAL.pdf.

[19] P. Besnard, S. Doutre, V. H. Ho, and D. Longin, "SESAME - A System for Specifying Semantics in Abstract Argumentation (regular paper)," in *International Workshop on Systems and Algorithms for Formal Argumentation (SAFA), Potsdam, Germany, 13/09/2016*, M. Thimm, F. Cerutti, H. Strass, and M. Vallati, Eds., vol. 1672. http://CEUR-WS.org: CEUR Workshop Proceedings, septembre 2016, pp. 40–51. [Online]. Available: http://ceur-ws.org/Vol-1672/paper_4.pdf

[20] D. Grossi, "On the logic of argumentation theory," in *Proceedings of the 9th International Conference on Autonomous Agents and Multiagent Systems: Volume 1 - Volume 1*, ser. AAMAS '10. Richland, SC: International Foundation for Autonomous Agents and Multiagent Systems, 2010, pp. 409–416. [Online]. Available: http://dl.acm.org/citation.cfm?id=1838206.1838264

[21] O. Arieli and M. W. Caminada, "A QBF-based formalization of abstract argumentation semantics," *Journal of Applied Logic*, vol. 11, no. 2, pp. 229 – 252, 2013. [Online]. Available: http://dx.doi.org/10.1016/j.jal.2013.03.009

[22] J. P. Wallner, G. Weissenbacher, and S. Woltran, "Advanced SAT techniques for abstract argumentation," in *Computational Logic in Multi-Agent Systems: 14th International Workshop, CLIMA XIV, Corunna, Spain, September 16-18, 2013. Proceedings*, J. Leite, T. C. Son, P. Torroni, L. van der Torre, and S. Woltran, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, pp. 138–154. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-40624-9_9

[23] F. Cerutti, P. E. Dunne, M. Giacomin, and M. Vallati, "Computing preferred extensions in abstract argumentation: A SAT-based approach," in *Theory and Applications of Formal Argumentation: Second International Workshop, TAFA 2013, Beijing, China, August 3-5, 2013, Revised Selected papers*, E. Black, S. Modgil, and N. Oren, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2014, pp. 176–193. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-54373-9_12

[24] B. Bogaerts, T. Janhunen, and S. Tasharrofi, "Declarative solver development: Case studies," in *Proc. of the Fifteenth International Conference on Principles of Knowledge Representation and Reasoning (KR)*, 2016, pp. 74–83.

[25] G. Brewka and S. Woltran, "Abstract dialectical frameworks," in *Proc. of the 20^{th} International Conference on the Principles of Knowledge Representation and Reasoning*, Toronto, Canada, 2010, pp. 102–111.

[26] C. Beierle, F. Brons, and N. Potyka, "A software system using a SAT solver for reasoning under complete, stable, preferred, and grounded argumentation semantics," in *KI 2015: Advances in Artificial Intelligence - 38th Annual German Conference on AI, Dresden, Germany, September 21-25, 2015, Proceedings*, 2015, pp. 241–248. [Online]. Available: http://dx.doi.org/10.1007/978-3-319-24489-1_19

[27] J. M. Lagniez, E. Lonca, and J. G. Mailly, "Coquiaas: A constraint-based quick abstract argumentation solver," in *Tools with Artificial Intelligence (ICTAI), 2015 IEEE 27th International Conference on*, Nov 2015, pp. 928–935. [Online]. Available: http://dx.doi.org/10.1109/ICTAI.2015.134

[28] C. Cayrol and M.-C. Lagasquie-Schiex, "The Grafix website." [Online]. Available: http://www.irit.fr/grafix

[29] C. Cayrol, J. Fandinno, L. Fariñas del Cerro, and M.-C. Lagasquie-Schiex, "Argumentation frameworks with recursive attacks and evidence-based support," in *Proc. of Tenth International Symposium on Foundations of Information and Knowledge Systems (FoIKS)*, F. Ferrarotti and S. Woltran, Eds., vol. LNCS 10833. Springer-Verlag, 2018, pp. 150–169.