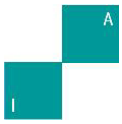
Inteligencia Artificial. Revista Iberoamericana de Inteligencia Artificial
ISSN: 1137-3601
revista@aepia.org
Asociación Española para la Inteligencia Artificial
España

Moguillansky, Martín O.; Rotstein, Nicolás D.; Falappa, Marcelo A.
Generalized Abstract Argumentation: A First-order Machinery towards Ontology Debugging
Inteligencia Artificial. Revista Iberoamericana de Inteligencia Artificial, vol. 14, núm. 46, 2010, pp. 17-33
Asociación Española para la Inteligencia Artificial
Valencia, España

Inteligencia Artificial 46(2010), 17-33
doi: 10.4114/ia.v14i46.1510



INTELIGENCIA ARTIFICIAL

http://erevista.aepia.org/

Generalized Abstract Argumentation: A First-order Machinery towards Ontology Debugging

Martín O. Moguillansky, Nicolás D. Rotstein, and Marcelo A. Falappa

Consejo Nacional de Investigaciones Científicas y Técnicas (CONICET)
Artificial Intelligence Research and Development Laboratory (LIDIA)
Department of Computer Science and Engineering (DCIC)
Universidad Nacional del Sur (UNS) – Bahía Blanca, ARGENTINA
e-mail: {mom,ndr,mfalappa}@cs.uns.edu.ar

Abstract Same as Dung's abstract argumentation framework (`AF`), the notion of generalized abstract argumentation framework (`GenAF`) aims at reasoning about inconsistency disregarding any logic for arguments; thus both the knowledge base (KB) and language to express beliefs remain unspecified. Nonetheless, reifying the abstract configuration of the `AF` argumentation machineries to specific logics may bring about some inconveniences.

A `GenAF` is assumed to eventually relate first-order logic (FOL) formulae to abstract arguments. The main purpose of the generalization is to provide a theory capable of reasoning (following argumentation technics) about inconsistent knowledge bases (KB) expressed in any FOL fragment. Consequently, the notion of argument is related to a single formula in the KB. This allows to share the same primitive elements from both, the framework (arguments) and, the KB (formulae). A framework with such features would not only allow to manage a wide range of knowledge representation languages, but also to cope straightforwardly with the dynamics of knowledge.

Once the formalism is presented, we propose a reification to the description logic $\mathcal{ALC}$ with the intention to handle ontology debugging. In this sense, since argumentation frameworks reason over graphs that relate arguments through attack, our methodology is proposed to bridge ontological inconsistency sources to attack relations in argumentation. Finally, an argumentation semantics adapted to `GenAF`'s, is proposed as a consistency restoration tool with the objective of debugging and repairing ontologies.

Keywords: Argumentation, first-order logic, inconsistency tolerance, ontology debugging, description logics.

1 Introduction

The formalism studied in this work is based on the widely accepted Dung's argumentation framework (`AF`) [6]. An `AF` is deemed as abstract since the language used to define arguments remains unspecified, thus, arguments in an `AF` are treated as "black boxes" of knowledge. In this work we go one step further into a not-so-abstract form of argumentation by proposing an argument language $\mathbb{A}$`rgs` in order to provide some structure to the notion of arguments while keeping them abstract. Intuitively, an *argument* may be seen as *an indivisible piece of knowledge inferring a claim from a set of premises*. Since claims and premises are distinguishable entities of any argument, we will allow both to be expressed through different sublanguages. The proposed *argument language* $\mathbb{A}$`rgs` is thus characterized through the interrelation between its inner components. Assuming arguments specified through $\mathbb{A}$`rgs` would bring about a highly versatile framework given that different knowledge representation languages could be handled through it. But consequently, some basic elements of the argumentation machinery should be accommodated,

ISSN: 1988-3064(on-line)

giving rise to a new kind of abstract argumentation frameworks identified as generalized (GenAF). The first approaches to a GenAF in [14, 17] were inspired by [24, 23].

The GenAF here proposed aims at reasoning about inconsistent KB's expressed through some FOL fragment. Consequently, $\mathbb{A}$rgs will eventually be reified according to the worked KB. Thus, the maximum expressive power of a GenAF is imposed by restricting the inner components of $\mathbb{A}$rgs to be bounded to some logic $\mathcal{L}^{\kappa}$, with $\kappa \in \mathbb{N}_0$[1]. Formulae in $\mathcal{L}^{\kappa}$ are those of FOL that can be built with the help of predicate symbols with arity $\leq \kappa$, including equality and constant symbols, but without function symbols. In particular, $\mathcal{L}^2$ has been shown to be decidable in [19]. An example of an $\mathcal{L}^2$-compliant logic is the $\mathcal{ALC}$ DL used to describe basic ontologies. The interested reader is referred to [5, 1].

A normal form for a $\mathcal{L}^{\kappa}$ KB is presented to reorganize the knowledge in the KB through sentences conforming some minimal pattern, which will be interpreted as single arguments in the GenAF. Therefore, a GenAF may be straightforwardly adapted to deal with dynamics of knowledge as done in [23]: removing an argument from the framework would mean deleting a statement from the KB. Argumentation frameworks were also related to FOL in [4], however, since there was no intention to cope with dynamics of arguments, no particular structure was provided to manage statements in the KB through single arguments. In this sense, our proposal is more similar to that in [27], although we relate the notions of deduction and conflict to FOL interpretations.

Specifying $\mathbb{A}$rgs could bring about some problems: the language for claims may consider conjunctive and/or disjunctive formulae. For the former case, the easiest option is to trigger a different claim for each conjunctive term. For the case of disjunctive formulae for claims, the problem seems to be more complicated. To that matter we introduce the notion of *coalition*, which is a structure capable of grouping several arguments with the intention to support an argument's premise, identify conflictive sources of knowledge, or even to infer new knowledge beyond the one specified through the arguments considered in it. In argumentation theory, an argument's premises are satisfied in order for that argument to reach its claim. This is usually referred as *support relation* [24], handled in this work through coalitions.

Usually, an abstract argument is treated as an indivisible entity that suffices to support a claim; here arguments are also indivisible but they play a smaller role: they are aggregated in structures which can be thought as if they were arguments in the usual sense [4]. The idea behind the aggregation of arguments within a structure is similar to that of sub-arguments [10]. Classic argumentation frameworks consider only ground arguments, that is, a claim is directly inferred if the set of premises are conformed. In our framework, we consider two different kinds of arguments: *ground* and *schematic*. In this sense, a set of premises might consider free variables, meaning that the claim, and therefore the inference, will depend on them. Thus, when an argument $\mathcal{B}$ counts with free variables in its claim or premises, it will be called schematic; whereas $\mathcal{B}$ is referred as ground, when its variables are instantiated. Instantiation of variables within a schematic argument may occur as a consequence of its premises being supported.

Afterwards, a basic acceptability semantics is proposed, inspired in the grounded semantics [3]. These semantics ensure the obtention of a consistent set of arguments, from which the accepted knowledge (warranted formulae) can be identified.

Once the GenAF is presented, the logic $\mathcal{L}^{\kappa}$ will be reified to the description logic $\mathcal{ALC}$ for ontologies, given that any concept description in such a logic can be translated into an $\mathcal{L}^2$ formula. Details on this matter may be referred to [5] and [1]. Besides, in [9] an extension of the $\mathcal{ALC}$ DL is presented and proved to be equivalent to $\mathcal{L}^2$. Considering an $\mathcal{ALC}$ ontology into an $\mathcal{ALC}$-based GenAF requires to relate some classical argumentation elements [6] like attack and support to identify different levels of inconsistency in ontologies [7]. Thereafter, the acceptability semantics presented before will be applied to the $\mathcal{ALC}$-based GenAF to determine a methodology for ontology debugging.

It is important to note that the argumentation machinery here proposed is semantically determined –by effect of interpretations as done in model-based theories. This would allow to propose further implementations relying on any preferred reasoner engine. For instance, an $\mathcal{ALC}$-based GenAF may be handled via tableaux technics which are usually used to implement ontology reasoners. Consequently, the proposed ontology debugging model could be developed by emulating the argumentation machinery, without implementing it. This is so given that $\mathcal{ALC}$ axioms are interpreted as single arguments in the GenAF, and in this manner, the proposed argumentation methodology may be simply considered a theoretical tool.

[1] Natural numbers are enclosed in the sets $\mathbb{N}_0 = \{0, 1, \dots\}$ and $\mathbb{N}_1 = \{1, 2, \dots\}$.

2 Foundations for a Generalized AF

For $\mathcal{L}^\kappa$, we use $p, p_1, p_2, \ldots$ and $q, q_1, q_2, \ldots$ to denote monadic predicate letters, $r, r_1, r_2, \ldots$ for dyadic predicate letters, x, y for free variable objects, and a, b, c, d for constants (individual names). Besides, the logic $\mathcal{L}_\mathcal{A} \subset \mathcal{L}^\kappa$ identifies the fragment of $\mathcal{L}^\kappa$ describing *assertional formulae* (ground atoms and their negations). Recall that ground atoms are atomic formulae which do not consider variable objects. The logic $\mathcal{L}^\kappa$ is interpreted as usual through interpretations $\mathcal{I} = \langle \Delta^\mathcal{I}, p^\mathcal{I}, p_1^\mathcal{I}, \ldots, q^\mathcal{I}, q_1^\mathcal{I}, \ldots, r^\mathcal{I}, r_1^\mathcal{I}, \ldots \rangle$, where $\Delta^\mathcal{I}$ is the interpretation domain, and $p^\mathcal{I}, p_1^\mathcal{I}, \ldots, q^\mathcal{I}, q_1^\mathcal{I}, \ldots, r^\mathcal{I}, r_1^\mathcal{I}, \ldots$ interpret $p, p_1, \ldots, q, q_1, \ldots, r, r_1, \ldots$, respectively. For an interpretation $\mathcal{I}$, some $a \in \Delta^\mathcal{I}$, and a formula $\varphi(x)$, we write $\mathcal{I} \models \varphi(a)$ if $\mathcal{I}, v \models \varphi(x)$, for the assignment v mapping x to a. For simplicity we omit universal quantifiers writing $\varphi(x)$ to refer to $(\forall x)(\varphi(x))$.

As mentioned before, we will rely on a (abstract) language $\mathbb{A}\texttt{rgs}$ (for arguments) composed by two (unspecified) inner sub-languages: $\mathcal{L}_\texttt{pr}$ (for premises) and $\mathcal{L}_\texttt{cl}$ (claims).

Definition 1 (Argument Language) *Given the logic $\mathcal{L}^\kappa$, an **argument language** $\mathbb{A}\texttt{rgs}$ is defined as $2^{\mathcal{L}_\texttt{pr}} \times \mathcal{L}_\texttt{cl}$, where $\mathcal{L}_\texttt{cl} \subseteq \mathcal{L}^\kappa$ and $\mathcal{L}_\texttt{pr} \subseteq \mathcal{L}^\kappa$ are recognized as the respective languages for claims and premises in $\mathbb{A}\texttt{rgs}$.*

Since a premise is supported through the claim of other argument/s, the expressivity of both languages $\mathcal{L}_\texttt{pr}$ and $\mathcal{L}_\texttt{cl}$ should be controlled in order to allow every describable premise to be supported by formulae from the language for claims. Therefore, to handle the language $\mathbb{A}\texttt{rgs}$ at an abstract level, we will characterize it by relating $\mathcal{L}_\texttt{pr}$ and $\mathcal{L}_\texttt{cl}$.

Definition 2 (Legal Argument Language) *An argument language $2^{\mathcal{L}_\texttt{pr}} \times \mathcal{L}_\texttt{cl}$ is **legal** iff for every $\rho \in \mathcal{L}_\texttt{pr}$ there is a set $\Phi \subseteq \mathcal{L}_\texttt{cl}$ such that $\Phi \models \rho$ **(support)**.*

In the sequel any argument language used will be assumed to be legal. Argumentation frameworks are a tool to reason about potentially inconsistent knowledge bases. Due to complexity matters, it would be interesting to interpret any $\mathcal{L}^\kappa$ KB directly as an argumentation framework with no need to transform the KB to a `GenAF`. Intuitively, an argument poses a reason to believe in a claim if it is the case that its premises are supported. This intuition is similar to the notion of material conditionals (implications "$\rightarrow$") in classical logic. Hence, statements from a KB could give rise to a single argument. To this end, we propose a normal form for $\mathcal{L}^\kappa$ KBs.

Definition 3 (pre-Argumental Normal Form (pANF)) *Given a knowledge base $\Sigma \subseteq \mathcal{L}^\kappa$, and an argument language $\mathbb{A}\texttt{rgs}$, Σ conforms to the **pre-argumental normal form** (pANF) iff every formula $\varphi \in \Sigma$ is an assertion in $\mathcal{L}_\mathcal{A}$, or it corresponds to the form $\rho_1 \wedge \ldots \wedge \rho_n \rightarrow \alpha$, where $\alpha \in \mathcal{L}_\texttt{cl}$ and $\rho_i \in \mathcal{L}_\texttt{pr}$ $(1 \leq i \leq n)$. Hence, each formula $\varphi \in \Sigma$ is said to be in pANF.*

Example 1 [2] *Suppose $\mathcal{L}_\texttt{cl}$ and $\mathcal{L}_\texttt{pr}$ are concretized as follows: $\mathcal{L}_\texttt{cl}$ allows disjunctions but prohibits conjunctions; whereas $\mathcal{L}_\texttt{pr}$ avoids both conjunctions and disjunctions. This would require for a formula like $(p_1(x) \wedge p_2(x)) \vee (p_3(x) \wedge p_4(x)) \rightarrow q_1(x) \wedge (q_2(x) \vee q_3(x))$ to be reformatted into the pANF formulae $p_1(x) \wedge p_2(x) \rightarrow q_1(x)$, $p_1(x) \wedge p_2(x) \rightarrow q_2(x) \vee q_3(x)$, $p_3(x) \wedge p_4(x) \rightarrow q_1(x)$ and $p_3(x) \wedge p_4(x) \rightarrow q_2(x) \vee q_3(x)$.*

Next we formalize the generalized notion of argument independently from a KB. The relation between premises and claims wrt. a KB could be referred to Remark 1.

Definition 4 (Argument) *An **argument** $\mathcal{B} \in \mathbb{A}\texttt{rgs}$ is a pair $\langle \Gamma, \alpha \rangle$, where $\Gamma \subseteq \mathcal{L}_\texttt{pr}$ is a finite set of finite premises, $\alpha \in \mathcal{L}_\texttt{cl}$, its finite claim, and $\Gamma \cup \{\alpha\} \not\models \bot$ **(consistency)**.*

Usually, *evidence* is considered a basic irrefutable piece of knowledge. This means that evidence does not need to be supported given that it is self-justified by definition. Thus, two options appear to specify evidence: as a separate entity in the framework, or as arguments with no premises to be satisfied. In this article we assume the latter posture, referring to them as *evidential arguments*.

[2] For simplicity, examples are enclosed within $\mathcal{L}^2$ to consider only predicates of arity ≤ 2.

Definition 5 (Evidence) *Given an argument $\mathcal{B} \in \mathbb{A}\mathtt{rgs}$, $\mathcal{B}$ is referred as **evidential argument** (or just evidence) iff $\mathcal{B} = \langle\{\}, \alpha\rangle$ with $\alpha \in \mathcal{L}_{\mathcal{A}}$ (assertional formulae).*

Given $\mathcal{B} \in \mathbb{A}\mathtt{rgs}$, its claim and set of premises are identified by the functions $\mathfrak{cl} : \mathbb{A}\mathtt{rgs} \longrightarrow \mathcal{L}_{\mathtt{cl}}$, and $\mathfrak{pr} : \mathbb{A}\mathtt{rgs} \longrightarrow 2^{\mathcal{L}_{\mathtt{pr}}}$, respectively. For instance, given $\mathcal{B} = \langle\{\rho_1, \rho_2\}, \alpha\rangle$, its premises are $\mathfrak{pr}(\mathcal{B}) = \{\rho_1, \rho_2\}$, and its claim, $\mathfrak{cl}(\mathcal{B}) = \alpha$. Arguments will be obtained from `pANF` formulae through an *argument translation function* $\mathfrak{arg} : \mathcal{L}^{\kappa} \longrightarrow \mathbb{A}\mathtt{rgs}$ such that $\mathfrak{arg}(\varphi) = \langle\{\rho_1, \ldots, \rho_n\}, \alpha\rangle$ *iff* $\varphi \in \mathcal{L}^{\kappa}$ is a `pANF` formula $\rho_1 \wedge \ldots \wedge \rho_n \rightarrow \alpha$ and $\mathfrak{arg}(\varphi)$ verifies the conditions in Definition 4. Otherwise, $\mathfrak{arg}(\varphi) = \langle\emptyset, \bot\rangle$. An evidential argument $\mathfrak{arg}(\varphi) = \langle\emptyset, \alpha\rangle$ appears if φ is $\rightarrow \alpha$.

Example 2 (Continued from Example 1) *For the formulae given in Example 1, by effect of the function "$\mathfrak{arg}$" the following arguments are triggered: $\langle\{p_1(x), p_2(x)\},\ q_1(x)\rangle$, $\langle\{p_1(x), p_2(x)\}, q_2(x) \vee q_3(x)\rangle$, $\langle\{p_3(x), p_4(x)\},\ q_1(x)\rangle$ and $\langle\{p_3(x), p_4(x)\}, q_2(x) \vee q_3(x)\rangle$.*

As mentioned before, it is important to recall that the notion of argument adopted in this work differs from its usual usage. This is made clear in the following remark.

Remark 1 *Given a `pANF` KB $\Sigma \subseteq \mathcal{L}^{\kappa}$, a formula $\varphi \in \Sigma$, and its associated argument $\mathfrak{arg}(\varphi) = \langle\Gamma, \alpha\rangle$; it follows $\Sigma \models (\bigwedge \Gamma) \rightarrow \alpha$, but $\Gamma \models \alpha$ does not necessarily hold.*

A more restrictive definition of argument could consider conditions like $\Gamma \not\models \alpha$, and/or $\Gamma \setminus \{\rho\} \not\models \rho$, with $\rho \in \Gamma$. However, its appropriate discussion exceeds the scope of this article. For the usual notion of argument see *argumental structures* in Definition 15.

The formalization of the `GenAF` will rely on *normality conditions*: user defined constraints in behalf of the appropriate construction of the argumentation framework.

Definition 6 (GenAF) *A **generalized abstract argumentation framework** (`GenAF`) is a pair $\langle\mathbf{A}, \mathbf{N}\rangle$, where $\mathbf{A} \subseteq \mathbb{A}\mathtt{rgs}$ is a finite set of arguments, and $\mathbf{N} \subseteq \mathbb{N}\mathtt{orm}$, a finite set of normality condition functions $\mathfrak{nc} : 2^{\mathbb{A}\mathtt{rgs}} \longrightarrow \{true, false\}$. The domain of functions $\mathfrak{nc}$ is identified through $\mathbb{N}\mathtt{orm}$, and $\mathbb{G}$ identifies the class of every `GenAF`. The set $\mathbf{E} \subseteq \mathbf{A}$ encloses every evidential argument from $\mathbf{A}$.*

A normality condition required through $\mathbf{N}$, could be to require evidence to be consistent, that is no pair of contradictory evidential arguments should be available in the framework. Other conditions could be to restrict arguments from being non-minimal justifications for the claim, or from including the claim itself as a premise.

(evidence coherency) there is no pair $\langle\{\}, \alpha\rangle \in \mathbf{A}$ and $\langle\{\}, \neg\alpha\rangle \in \mathbf{A}$.

(minimality) there is no pair $\langle\Gamma, \alpha\rangle \in \mathbf{A}$ and $\langle\Gamma', \alpha\rangle \in \mathbf{A}$ such that $\Gamma' \subset \Gamma$.

(relevance) there is no $\langle\Gamma, \alpha\rangle \in \mathbf{A}$ such that $\alpha \in \Gamma$.

Other normality conditions may appear depending on the concretization of the logic for arguments and the environment the framework is set to model. The complete study of these features falls out of the scope of this article. Given a `GenAF` $T = \langle\mathbf{A}, \mathbf{N}\rangle$, we will say that T is a *theory* *iff* for every normality condition $\mathfrak{nc} \in \mathbf{N}$ it follows $\mathfrak{nc}(\mathbf{A}) = true$. From now on we will work only with theories, thus unless the contrary is stated, every framework will be assumed to conform a theory. Moreover, the framework specification is done in such a way that its correctness does not rely on the normality conditions required. Thus, a `GenAF` $\langle\mathbf{A}, \mathbf{N}\rangle$, with $\mathbf{N} = \emptyset$, will be trivially a theory.

In order to univocally determine a single `GenAF` from a given KB and a set of normality conditions, it is necessary to assume a comparison criterion among formulae in the KB. Such criterion could be defined for instance, upon entrenchment of knowledge, *i.e.*, levels of importance are related to formulae in the KB. As will be seen later, this criterion will determine the *argument comparison criterion* from which the attack relation is usually specified in the classic argumentation literature. Next, we define a *theory function* to identify the `GenAF` associated to a KB.

Definition 7 (Theory Function) *Given a* pANF *knowledge base* $\Sigma \subseteq \mathcal{L}^{\kappa}$*, and a set* $\mathbf{N} \subseteq \mathbb{N}\mathtt{orm}$ *of normality condition functions* $\mathfrak{nc}$*, a* ***theory function*** $\mathfrak{genaf} : 2^{\mathcal{L}^{\kappa}} \times 2^{\mathbb{N}\mathtt{orm}} \longrightarrow \mathbb{G}$ *identifies the* GenAF $\mathfrak{genaf}(\Sigma, \mathbf{N}) = \langle \mathbf{A}, \mathbf{N} \rangle$*, where* $\mathbf{A} \subseteq \{\mathfrak{arg}(\varphi) | \varphi \in \Sigma$ *and* $\mathfrak{arg}(\varphi)$ *is an argument*$\} \cup \{\mathfrak{arg}(\varphi \rightarrow \varphi') | (\neg\varphi' \rightarrow \neg\varphi) \in \Sigma$ *and* $\mathfrak{arg}(\varphi \rightarrow \varphi')$ *is an argument*$\}$ *and* $\mathbf{A}$ *is the maximal set (wrt. set inclusion and the comparison criterion in* Σ*) such that for every* $\mathfrak{nc} \in \mathbf{N}$ *it holds* $\mathfrak{nc}(\mathbf{A}) = true$*.*

The GenAF obtained by the function "$\mathfrak{genaf}$" will consider a maximal subset of the KB Σ such that the resulting set of arguments (triggered by "$\mathfrak{arg}$") is compliant with the normality conditions. Note that also the counterpositive formula of each one considered is assumed to conform an argument in the resulting GenAF. This is natural since counterpositive formulae from the statements in a KB are implicitly considered to reason in classical logic. In a GenAF, this issue is done by considering both explicitly.

3 The GenAF Argumentation Machinery

The purpose of generalizing an abstract argumentation framework comes from the need of managing different argument languages specified through some FOL fragment. Given the specification of $\mathbb{A}\mathtt{rgs}$, different possibilities may arise, for instance, the language for claims may accept disjunction of formulae. Thus, it is possible to infer a formula in $\mathcal{L}_{\mathtt{cl}}$ from several arguments in the GenAF through their claims. Consider for example, two arguments $\langle \{p_1(x)\}, q_1(x) \vee q_2(x) \rangle$ and $\langle \{p_2(x)\}, \neg q_2(x) \rangle$, the claim $q_1(x)$ may be inferred. This kind of constructions are similar to arguments themselves, but are implicitly obtained from the GenAF at issue. To such matter, the notion of *claiming-coalition* is introduced as *a coalition required to infer a new claim.*

In general, a *coalition* might be interpreted as *a minimal and consistent set of arguments guaranteeing certain requirement.* We say that a coalition $\widehat{\mathcal{C}} \subseteq \mathbb{A}\mathtt{rgs}$ is consistent *iff* $\mathfrak{prset}(\widehat{\mathcal{C}}) \cup \mathfrak{clset}(\widehat{\mathcal{C}}) \not\models \bot$, while minimality ensures that $\widehat{\mathcal{C}}$ guarantees a requirement θ *iff* there is no proper subset of $\widehat{\mathcal{C}}$ guaranteeing θ. The functions $\mathfrak{clset} : 2^{\mathbb{A}\mathtt{rgs}} \longrightarrow 2^{\mathcal{L}_{\mathtt{cl}}}$ and $\mathfrak{prset} : 2^{\mathbb{A}\mathtt{rgs}} \longrightarrow 2^{\mathcal{L}_{\mathtt{pr}}}$ are defined as $\mathfrak{clset}(\widehat{\mathcal{C}}) = \{\mathfrak{cl}(\mathcal{B}) | \mathcal{B} \in \widehat{\mathcal{C}}\}$, and $\mathfrak{prset}(\widehat{\mathcal{C}}) = \bigcup_{\mathcal{B} \in \widehat{\mathcal{C}}} \mathfrak{pr}(\mathcal{B})$, to respectively identify the set of claims and premises from $\widehat{\mathcal{C}}$. In this article, three types of coalitions will be considered. Regarding claiming-coalitions, the requirement θ is a new inference in $\mathcal{L}_{\mathtt{cl}}$ from the arguments considered by the coalition.

Definition 8 (Claiming-Coalition) *Given a* GenAF $\langle \mathbf{A}, \mathbf{N} \rangle \in \mathbb{G}$*, and a formula* $\alpha \in \mathcal{L}_{\mathtt{cl}}$*, a set of arguments* $\widehat{\mathcal{C}} \subseteq \mathbf{A}$ *is a claiming-coalition, or just a* ***claimer****, of* α *iff* $\widehat{\mathcal{C}}$ *is the minimal coalition guaranteeing* $\mathfrak{clset}(\widehat{\mathcal{C}}) \models \alpha$ *and* $\widehat{\mathcal{C}}$ *is consistent.*

Note that a claiming-coalition containing a single argument $\mathcal{B}$ is a primitive coalition for the claim of $\mathcal{B}$. As said before, an argument needs to find its premises supported as a functional part of the reasoning process to reach its claim. In this framework, due to the characterization of $\mathbb{A}\mathtt{rgs}$, sometimes a formula from $\mathcal{L}_{\mathtt{pr}}$ could be satisfied only through several formulae from $\mathcal{L}_{\mathtt{cl}}$. This means that a single argument is not always enough to support a premise of another argument. Thus, we will extend the usual definition of supporter [24] by introducing the notion of *supporting-coalition.*

Definition 9 (Supporting-Coalition) *Given a* GenAF $\langle \mathbf{A}, \mathbf{N} \rangle \in \mathbb{G}$*, an argument* $\mathcal{B} \in \mathbf{A}$*, and a premise* $\rho \in \mathfrak{pr}(\mathcal{B})$*. A set of arguments* $\widehat{\mathcal{C}} \subseteq \mathbf{A}$ *is a supporting-coalition, or just a* ***supporter****, of* $\mathcal{B}$ *through* ρ *iff* $\widehat{\mathcal{C}}$ *is the minimal coalition guaranteeing* $\mathfrak{clset}(\widehat{\mathcal{C}}) \models \rho$ *and* $\widehat{\mathcal{C}} \cup \{\mathcal{B}\}$ *is consistent.*

Example 3 *Assume* $\mathbf{A} = \{\mathcal{B}_1, \mathcal{B}_2, \mathcal{B}_3, \mathcal{B}_4\}$*, where* $\mathcal{B}_1 = \langle \{p_1(x)\}, q_1(x) \rangle$*,* $\mathcal{B}_2 = \langle \{p_1(x)\}, q_2(x) \rangle$*,* $\mathcal{B}_3 = \langle \{p_2(x)\}, p_1(x) \vee q_1(x) \rangle$*, and* $\mathcal{B}_4 = \langle \{p_3(x)\}, \neg q_1(x) \rangle$*. The set* $\widehat{\mathcal{C}} = \{\mathcal{B}_3, \mathcal{B}_4\}$ *is a supporter of* $\mathcal{B}_2$*. Note that* $\widehat{\mathcal{C}}$ *cannot be a supporting-coalition of* $\mathcal{B}_1$ *since it violates (supporter) consistency.*

When not every necessary argument to conform the supporting-coalition is present in $\mathbf{A}$, the (unsupported) premise is referred as free.

Definition 10 (Free Premise) *Given a* GenAF $\langle \mathbf{A}, \mathbf{N} \rangle \in \mathbb{G}$ *and an argument* $\mathcal{B} \in \mathbf{A}$*, a premise* $\rho \in \mathfrak{pr}(\mathcal{B})$ *is* ***free*** *wrt.* $\mathbf{A}$ *iff there is no supporter* $\widehat{\mathcal{C}} \subseteq \mathbf{A}$ *of* $\mathcal{B}$ *through* ρ*.*

From Example 3, premises $p_2(x) \in \mathfrak{pr}(\mathcal{B}_3)$, $p_3(x) \in \mathfrak{pr}(\mathcal{B}_4)$, and $p_1(x) \in \mathfrak{pr}(\mathcal{B}_1)$ are free wrt. $\mathbf{A}$; whereas $p_1(x) \in \mathfrak{pr}(\mathcal{B}_2)$ is not.

When a schematic argument is fully supported from evidence ($\widehat{\mathcal{C}} \subseteq \mathbf{E}$), its claim is ultimately instantiated ending up as a ground formula. Therefore, an argument $\mathcal{B}$ may be included in a supporting coalition $\widehat{\mathcal{C}}$ of $\mathcal{B}$ itself due to the substitution of variables. This situation is made clearer later and may be referred to Example 5. The quest for a supporter $\widehat{\mathcal{C}}$ of some argument $\mathcal{B}$ through a premise ρ in it, describes a recursive supporting process given that each premise in $\widehat{\mathcal{C}}$ needs to be also supported. When this process does ultimately end in a supporter containing only evidential arguments, we will distinguish $\rho \in \mathfrak{pr}(\mathcal{B})$ not only as non-free but also as *closed*.

Definition 11 (Closed Premise) *Given a* GenAF $\langle \mathbf{A}, \mathbf{N} \rangle \in \mathbb{G}$*, and an argument* $\mathcal{B} \in \mathbf{A}$*, a premise* $\rho \in \mathfrak{pr}(\mathcal{B})$ *is* ***closed*** *wrt.* $\mathbf{A}$ *iff there exists a supporter* $\widehat{\mathcal{C}} \subseteq \mathbf{A}$ *of* $\mathcal{B}$ *through* ρ *such that either* $\mathfrak{prset}(\widehat{\mathcal{C}}) = \emptyset$*, or every premise in* $\mathfrak{prset}(\widehat{\mathcal{C}})$ *is closed.*

The idea behind closing premises is to identify those arguments that effectively state a reason from the GenAF to believe in their claims. Such arguments will be those for which the support of each of its premises does ultimately end in a set of evidential arguments –and therefore no more premises are required to be supported. Thus, every premise in an argument is closed *iff* the claim is *inferrable*. This is natural since inferrable claims can be effectively reached from evidence. Finally, when the claiming-coalition of an inferrable claim passes the acceptability analysis, the claim ends up *warranted*. Acceptability analysis and warranted claims will be detailed later, in Sect. 4.2.

Definition 12 (Inferrable Formula) *Given a* GenAF $\langle \mathbf{A}, \mathbf{N} \rangle \in \mathbb{G}$*, a formula* $\alpha \in \mathcal{L}_{\mathtt{cl}}$ *is* ***inferrable*** *from* $\mathbf{A}$ *iff there exists a claiming-coalition* $\widehat{\mathcal{C}} \subseteq \mathbf{A}$ *for* α *such that either* $\mathfrak{prset}(\widehat{\mathcal{C}}) = \emptyset$*, or every premise in* $\mathfrak{prset}(\widehat{\mathcal{C}})$ *is closed.*

The supporting process closing every premise in a claiming-coalition $\widehat{\mathcal{C}}$ to verify whether the claim is inferrable, clearly conforms a tree rooted in $\widehat{\mathcal{C}}$. We will refer to such tree as *supporting-tree*, and to each branch in it as *supporting-chain*.

Definition 13 (Supporting-Chain) *Given a* GenAF $\langle \mathbf{A}, \mathbf{N} \rangle \in \mathbb{G}$*, a formula* $\alpha \in \mathcal{L}_{\mathtt{cl}}$*, and a sequence* $\lambda \in (2^{\mathbf{A}})^n$ *such that* $\lambda = \widehat{\mathcal{C}}_1 \ldots \widehat{\mathcal{C}}_n$*, where* $n \in \mathbb{N}_1$*,* $\widehat{\mathcal{C}}_1$ *is a claiming-coalition for* α*, and for every* $i \in \mathbb{N}_1$ *it follows* $\widehat{\mathcal{C}}_i \subseteq \mathbf{A}$*, and* $\widehat{\mathcal{C}}_{i+1}$ *is a supporting-coalition through some* $\rho_i \in \mathfrak{prset}(\widehat{\mathcal{C}}_i)$*. The notations* $|\lambda| = n$ *and* $\lambda[i]$ *are used to respectively identify the* ***length*** *of* λ *and the* ***node*** $\widehat{\mathcal{C}}_i$ *in it. The last supporting-coalition in* λ *(referred as* ***leaf****) is identified through the function* $\mathfrak{leaf}(\lambda) = \lambda[|\lambda|]$*. The function* $\overleftarrow{\lambda} : (2^{\mathbf{A}})^n \times \mathbb{N}_0 \longrightarrow \mathcal{L}_{\mathtt{cl}} \cup \mathcal{L}_{\mathtt{pr}} \cup \{\bot\}$ *identifies the* ***link*** $\overleftarrow{\lambda}[0] = \alpha$*; or* $\overleftarrow{\lambda}[i] = \rho_i$ *(*$0 < i < |\lambda|$*), where* $\rho_i \in \mathfrak{prset}(\lambda[i])$ *is supported by* $\lambda[i+1]$*; or* $\overleftarrow{\lambda}[i] = \bot$ *(*$i \geq |\lambda|$*). The set* $\lambda^* = \bigcup_i \lambda[i]$ *(with* $0 < i \leq |\lambda|$*) identifies the set of arguments included in* λ*. Finally,* λ *is a* ***supporting-chain for*** α *wrt.* $\mathbf{A}$ *iff it guarantees:*

(minimality) $\widehat{\mathcal{C}} \subseteq \lambda^*$ *is a supporter (claimer if* $i = 0$*) of* $\overleftarrow{\lambda}[i]$ *iff* $\widehat{\mathcal{C}} = \lambda[i+1]$ *(*$0 \leq i < |\lambda|$*).*

(exhaustivity) *every* $\rho \in \mathfrak{prset}(\mathfrak{leaf}(\lambda))$ *is free wrt.* λ^**.*

(acyclicity) $\overleftarrow{\lambda}[i] = \overleftarrow{\lambda}[j]$ *iff* $i = j$*, with* $\{i, j\} \subseteq \{0, \ldots, |\lambda| - 1\}$*.*

(consistency) $\mathfrak{prset}(\lambda^*) \cup \mathfrak{clset}(\lambda^*) \not\models \bot$*.*

From the definition above, a supporting-chain is a finite sequence of interrelated supporting-coalitions $\widehat{\mathcal{C}}_i$ through a link $\rho_i \in \mathfrak{prset}(\widehat{\mathcal{C}}_i)$ supported by $\widehat{\mathcal{C}}_{i+1}$. It is finite indeed, given that the set $\mathbf{A}$ is also finite, and that no link could be repeated in the chain (acyclicity). The minimality condition (wrt. set inclusion over λ^*) stands to consider as less arguments from $\mathbf{A}$ as it is possible in order to obtain the same chain, whereas the exhaustivity condition (wrt. the length $|\lambda|$) ensures that the chain is as long as it is possible wrt. λ^* (without cycles), that is, λ has all the possible links that can appear from the arguments considered to build it. Note that from minimality no pair of arguments for a same claim could be simultaneously considered in any supporting-chain. Finally, consistency is required given that the intention of the supporting-chain is to provide a tool to close a premise from the claiming-coalition. Next, supporting-trees are formalized upon the definition of supporting-chains.

Definition 14 (Supporting-Tree) *Given a* GenAF $\langle \mathbf{A}, \mathbf{N} \rangle \in \mathbb{G}$*, a formula* $\alpha \in \mathcal{L}_{\mathtt{cl}}$*, and a tree* $\mathcal{T}$ *of coalitions* $\widehat{\mathcal{C}} \subseteq \mathbf{A}$ *such that each node* $\widehat{\mathcal{C}}$ *is either:*

- ***the root*** *iff* $\widehat{\mathcal{C}}$ *is a claiming-coalition for* α*; or*
- ***an inner node*** *iff* $\widehat{\mathcal{C}}$ *is a supporting-coalition through* $\rho \in \mathfrak{prset}(\widehat{\mathcal{C}}')$*, where* $\widehat{\mathcal{C}}' \subseteq \mathbf{A}$ *is either an inner node or the root.*

The membership relation will be overloaded by writing $\lambda \in \mathcal{T}$ *and* $\widehat{\mathcal{C}} \in \mathcal{T}$ *to respectively identify the branch* λ *and the node* $\widehat{\mathcal{C}}$ *from* $\mathcal{T}$*. The set* $\mathcal{T}^* = \bigcup_{\widehat{\mathcal{C}} \in \mathcal{T}} \widehat{\mathcal{C}}$ *identifies the set of arguments included in* $\mathcal{T}$*. Hence,* $\mathcal{T}$ *is a* ***supporting-tree*** *iff it guarantees:*

(completeness) *every* $\lambda \in \mathcal{T}$ *is a supporting-chain of* α *wrt.* $\mathbf{A}$*.*

(minimality) *for every* $\lambda \in \mathcal{T}$*,* $\widehat{\mathcal{C}} \subseteq \mathcal{T}^*$ *is a supporting-coalition (claimer if* $i = 0$*) through* $\overleftarrow{\lambda}[i]$ *iff* $\widehat{\mathcal{C}} = \lambda[i+1]$ *(*$0 \leq i < |\lambda|$*).*

(exhaustivity) *for every* $\rho \in \mathfrak{prset}(\mathcal{T}^*)$*, if there is no* $\lambda \in \mathcal{T}$ *such that* $\overleftarrow{\lambda}[i] = \rho$ *(*$0 < i < |\lambda|$*) then* ρ *is free wrt.* $\mathcal{T}^*$*.*

(consistency) $\mathfrak{prset}(\mathcal{T}^*) \cup \mathfrak{clset}(\mathcal{T}^*) \not\models \bot$*.*

Finally, the notation $\mathfrak{Trees}_{\mathbf{A}}(\alpha)$ *identifies the set of all supporting-trees for* α *from* $\mathbf{A}$*.*

The completeness condition is required in order to restrict the supporting-tree to consider only supporting-chains as their branches. Similar to supporting-chain, minimality is required to avoid considering extra arguments to build the tree, while exhaustivity stands to ensure that every possible supporting-coalition $\widehat{\mathcal{C}} \subseteq \mathcal{T}^*$ through a premise in $\mathfrak{prset}(\mathcal{T}^*)$ is an inner node in the tree. Finally, consistency ensures that the whole supporting process of the premises in the claiming-coalition will end being non-contradictory, even among branches. It is important to note that a supporting-tree for $\alpha \in \mathcal{L}_{\mathtt{cl}}$ determines the set of arguments used in the (possibly inconclusive)[3] supporting process of some claiming-coalition of α. Such set will be referred as *structure*.

Definition 15 (Structure) *Given a* GenAF $\langle \mathbf{A}, \mathbf{N} \rangle \in \mathbb{G}$*, and a formula* $\alpha \in \mathcal{L}_{\mathtt{cl}}$*, a set* $\mathbb{S} \subseteq \mathbf{A}$ *identifies a* ***structure*** *for* α *iff there is a supporting-tree* $\mathcal{T} \in \mathfrak{Trees}_{\mathbf{A}}(\alpha)$ *for* α *such that* $\mathbb{S} = \mathcal{T}^*$*. The claim and premises of* $\mathbb{S}$ *can be respectively determined through the functions* $\mathfrak{cl} : 2^{\mathbb{A}\mathtt{rgs}} \longrightarrow \mathcal{L}_{\mathtt{cl}}$ *and* $\mathfrak{pr} : 2^{\mathbb{A}\mathtt{rgs}} \longrightarrow 2^{\mathcal{L}_{\mathtt{pr}}}$*, such that* $\mathfrak{cl}(\mathbb{S}) = \alpha$ *and* $\mathfrak{pr}(\mathbb{S}) = \{\rho \in \mathfrak{prset}(\mathbb{S}) \mid \rho \text{ is a free premise wrt. } \mathbb{S}\}$*. Finally, the structure* $\mathbb{S}$ *is* ***argumental*** *iff* $\mathfrak{pr}(\mathbb{S}) = \emptyset$*, otherwise* $\mathbb{S}$ *is* ***schematic****.*

Note that functions "$\mathfrak{pr}$" and "$\mathfrak{cl}$" are overloaded and can be applied both to arguments and structures. This is not going to be problematic since either usage will be rather explicit. Besides, a structure $\mathbb{S}$ formed by a single argument is referred as *primitive* *iff* $|\mathbb{S}| = 1$. Thus, if $\mathbb{S} = \{\mathcal{B}\}$ then $\mathfrak{pr}(\mathcal{B}) = \mathfrak{pr}(\mathbb{S})$ and $\mathfrak{cl}(\mathcal{B}) = \mathfrak{cl}(\mathbb{S})$. However, not every single argument has an associated primitive structure. For instance, unless relevance would be required as a framework's normality condition, no structure could contain an argument $\langle \{p(x)\}, p(x) \rangle$ given that it would violate (supporting-chain) acyclicity. Finally, when no distinction is needed, we refer to primitive, schematic, or argumental structures, simply as structures.

Example 4 *Given two arguments* $\mathcal{B}_1 = \langle \{p(x)\}, q(x) \rangle$ *and* $\mathcal{B}_2 = \langle \{q(x)\}, p(x) \rangle$*. The set* $\{\mathcal{B}_1, \mathcal{B}_2\}$ *cannot be a structure for* $q(x)$ *since* $\{\mathcal{B}_1\}\{\mathcal{B}_2\}\{\mathcal{B}_1\}\ldots$ *is a supporting-chain violating acyclicity. Similarly,* $\{\mathcal{B}_1, \mathcal{B}_2\}$ *could neither be a structure for* $p(x)$*.*

Given two structures $\mathbb{S} \subseteq \mathbb{A}\mathtt{rgs}$ for $\alpha \in \mathcal{L}_{\mathtt{cl}}$, and $\mathbb{S}' \subseteq \mathbb{A}\mathtt{rgs}$ for $\alpha' \in \mathcal{L}_{\mathtt{cl}}$, $\mathbb{S}'$ is a *sub-structure* of $\mathbb{S}$ (noted as $\mathbb{S}' \trianglelefteq \mathbb{S}$) *iff* $\mathbb{S}' \subseteq \mathbb{S}$. Besides, $\mathbb{S}' \triangleleft \mathbb{S}$ *iff* $\mathbb{S}' \subset \mathbb{S}$.

Proposition 1 [4] *Given a* GenAF $\langle \mathbf{A}, \mathbf{N} \rangle \in \mathbb{G}$*, a formula* $\alpha \in \mathcal{L}_{\mathtt{cl}}$*, and two structures* $\mathbb{S} \subseteq \mathbf{A}$ *for* α *and* $\mathbb{S}' \subseteq \mathbf{A}$ *for* α*,*

[3] Inconclusive supporting processes lead to schematic structures with non-free premises wrt. $\mathbf{A}$.
[4] In this work, proofs were omitted due to space reasons.

- *if* $\mathbb{S}' \lhd \mathbb{S}$ *then* $\mathfrak{pr}(\mathbb{S}) \neq \mathfrak{pr}(\mathbb{S}')$.
- *if* $\mathbb{S}$ *is argumental then* $\mathfrak{leaf}(\lambda) \subseteq \mathbf{E}$, *for every* $\lambda \in \mathcal{T}$ *where* $\mathcal{T} \in \mathfrak{Trees}_{\mathbb{S}}(\alpha)$.

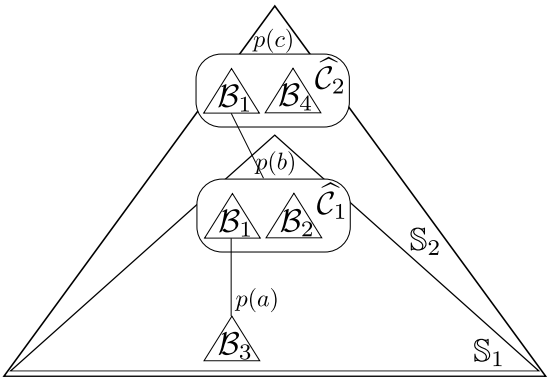
Lemma 1 *Given a* GenAF $\langle \mathbf{A}, \mathbf{N} \rangle \in \mathbb{G}$, *and a formula* $\alpha \in \mathcal{L}_{\mathtt{cl}}$, *a structure* $\mathbb{S} \subseteq \mathbf{A}$ *for* α *is argumental iff* α *is inferrable.*

If a formula $\varphi(x) \in \mathcal{L}_{\mathtt{cl}}$ (where x is a free variable) is inferrable then there exists an argumental structure $\mathbb{S}$ for $\varphi(x)$. Note now that since every argumental structure contains an empty set of premises, its supporting-tree $\mathcal{T}$ has only evidential arguments in their leaves. Thus, since the claim of evidential arguments are expressed in the language $\mathcal{L}_{\mathcal{A}}$–it considers no free variables– the inner supporting process of $\mathbb{S}$ performed through $\mathcal{T}$ ends up applying a *variable substitution*, for instance mapping x to a, such that $\mathfrak{cl}(\mathbb{S}) = \varphi(a)$. Finally, if a structure states a property about some element of the world through a claim considering only free variables then it is schematic.

Lemma 2 *Given a* GenAF $\langle \mathbf{A}, \mathbf{N} \rangle \in \mathbb{G}$, *and a formula* $\varphi(x) \in \mathcal{L}_{\mathtt{cl}}$, *a structure* $\mathbb{S} \subseteq \mathbf{A}$ *for* $\varphi(x)$ *is argumental iff* $\mathfrak{cl}(\mathbb{S}) = \varphi(a)$ *and* $\varphi(a), v \models \varphi(x)$, *where* v *maps* x *to* a.

In the following example, the formation of complex structures is illustrated. Note that an argument might be used several times within a structure if it is supported by different arguments. This is so due to different variable instantiantions.

Example 5 *Assume a* GenAF $\langle \mathbf{A}, \mathbf{N} \rangle$ *such that* $\{\mathcal{B}_1, \mathcal{B}_2, \mathcal{B}_3, \mathcal{B}_4\} \subseteq \mathbf{A}$ *where* $\mathcal{B}_1 = \langle \{p(x)\}, (\forall y)(\neg r(x,y) \vee p(y)) \rangle$, $\mathcal{B}_2 = \langle \{\}, r(a,b) \rangle$, $\mathcal{B}_3 = \langle \{\}, p(a) \rangle$, *and* $\mathcal{B}_4 = \langle \{\}, r(b,c) \rangle$. *Set* $\widehat{\mathcal{C}}_1 = \{\mathcal{B}_1, \mathcal{B}_2\}$ *is a supporter of* $\mathcal{B}_1$ *through* $p(b)$, *and set* $\widehat{\mathcal{C}}_2 = \{\mathcal{B}_1, \mathcal{B}_4\}$ *is a claimer of* $p(c)$. *Note that as a result of substituting variables* x *and* y *(from* $\mathcal{B}_1 \in \widehat{\mathcal{C}}_1$*) to* a *and* b *(from* $\mathcal{B}_2 \in \widehat{\mathcal{C}}_1$*) respectively, we have* $\mathfrak{prset}(\widehat{\mathcal{C}}_1) = \{p(a)\}$, *which in turn is supported through the primitive coalition* $\{\mathcal{B}_3\}$; *substituting similarly* x *and* y *(from* $\mathcal{B}_1 \in \widehat{\mathcal{C}}_2$*) to* b *and* c *(from* $\mathcal{B}_4 \in \widehat{\mathcal{C}}_2$*), set of premises* $\mathfrak{prset}(\widehat{\mathcal{C}}_2) = \{p(b)\}$ *ends up supported by* $\widehat{\mathcal{C}}_1$. *Hence, the schematic structures* $\mathbb{S}_1 = \{\mathcal{B}_1, \mathcal{B}_2, \mathcal{B}_3\}$ *for* $p(b)$, *and* $\mathbb{S}_2 = \{\mathcal{B}_1, \mathcal{B}_2, \mathcal{B}_3, \mathcal{B}_4\}$ *for* $p(c)$, *appear such that* $\mathbb{S}_1 \lhd \mathbb{S}_2$. *Note that the supporting-tree in* $\mathfrak{Trees}_{\mathbb{S}_2}(p(c))$ *has the unique supporting-chain* $\{\mathcal{B}_1, \mathcal{B}_4\}\{\mathcal{B}_1, \mathcal{B}_2\}\{\mathcal{B}_3\}$. *The figure depicted on the right illustrates both schematic structures and their inner components detailed above.*



4 Founding the GenAF Reasoner

In this section we present the foundamentals for recognizing conflicts from the GenAF's set of arguments and the semantics to specify the acceptability of arguments involved in the generalized framework.

4.1 Conflict Recognition

Two argumental structures are in conflict whenever their claims cannot be assumed together. Schematic structures may also be conflictive if it is the case that a claim of one of them could support a premise of the other, but a supporting-coalition does not exist given consistency would be violated. A second option of conflict between schematic structures appears when the premises of one of them infer the premises of the other, and either claim is in conflict with some premise from the other, or both claims cannot be assumed together. The intuition for this may be seen as a framework lacking of evidence to close every premise in each structure, but a hypothetical addition of the lacking evidence of one of them would be enough to include in the new framework two different argumental structures containing each original schematic structure. In such a case, the conflict conforms to the first case given.

This discussion may be made extensive to coalitional sets of structures. Analogous to coalitions of arguments, a *coalition of structures* might be interpreted as *a minimal and consistent set of structures guaranteeing certain requirement*. To go one step further into the formalization of a coalition $\widehat{\mathbb{C}} \subseteq 2^{\mathbb{A}\mathtt{rgs}}$ of structures $\mathbb{S} \subseteq \mathbf{A}$, we will rely on the set $\mathbb{C}^* = \bigcup_{\mathbb{S}\in\widehat{\mathbb{C}}} \mathbb{S}$ of arguments from $\widehat{\mathbb{C}}$. Therefore, we say that a coalition $\widehat{\mathbb{C}}$ of structures $\mathbb{S}$, is consistent *iff* $\mathfrak{prset}(\mathbb{C}^*) \cup \mathfrak{clset}(\mathbb{C}^*) \not\models \bot$, while minimality ensures $\widehat{\mathbb{C}}$ guarantees a requirement θ *iff* there is no proper subset of $\widehat{\mathbb{C}}$ guaranteeing θ, and there is no $\widehat{\mathbb{C}'} \subseteq 2^{\mathbb{A}\mathtt{rgs}}$ guaranteeing θ such that $\mathbb{C}'^* \subset \mathbb{C}^*$. Note that minimality not only looks for the smallest set of structures, but also for the smallest structures.

Coalition of structures are sets grouping structures to guarantee certain requirement θ: *conflict*. For the formalization of the notion of conflict, we will rely on the functions $\mathfrak{clset} : 2^{2^{\mathbf{A}}} \longrightarrow 2^{\mathcal{L}_{\mathtt{cl}}}$ and $\mathfrak{prset} : 2^{2^{\mathbf{A}}} \longrightarrow 2^{\mathcal{L}_{\mathtt{pr}}}$, which are respectively defined as $\mathfrak{clset}(\widehat{\mathbb{C}}) = \{\mathfrak{cl}(\mathbb{S}) | \mathbb{S} \in \widehat{\mathbb{C}}\}$, and $\mathfrak{prset}(\widehat{\mathbb{C}}) = \bigcup_{\mathbb{S}\in\widehat{\mathbb{C}}} \mathfrak{pr}(\mathbb{S})$. Note that functions "$\mathfrak{clset}$" and "$\mathfrak{prset}$" are overloaded and can be applied both to sets of arguments (for instance coalitions $\widehat{\mathcal{C}}$) and to coalitions $\widehat{\mathbb{C}}$ of structures. For this latter case, the functions' outcomes are the claims and premises of the structures included by the coalition $\widehat{\mathbb{C}}$. Next, we specify the notion of conflict between pairs of coalition of structures.

Definition 16 (Conflicting Coalitions) *Given a* GenAF $\langle \mathbf{A}, \mathbf{N}\rangle \in \mathbb{G}$*, two coalitions* $\widehat{\mathbb{C}} \subseteq 2^{\mathbf{A}}$ *and* $\widehat{\mathbb{C}'} \subseteq 2^{\mathbf{A}}$ *of structures are in* ***conflict*** *iff it follows:*

- *Both coalitions are related either through dependency or support:*

  (dependency) $\mathfrak{prset}(\widehat{\mathbb{C}}) \models \mathfrak{prset}(\widehat{\mathbb{C}'})$.

  (support) $\mathfrak{clset}(\widehat{\mathbb{C}}) \models \mathfrak{prset}(\widehat{\mathbb{C}'})$.

- *The conflict appears either from claim-clash or premise-clash:*

  (claim-clash) $\mathfrak{clset}(\widehat{\mathbb{C}}) \cup \mathfrak{clset}(\widehat{\mathbb{C}'}) \models \bot$.

  (premise-clash) $\mathfrak{clset}(\widehat{\mathbb{C}}) \cup \mathfrak{prset}(\widehat{\mathbb{C}'}) \models \bot$*, or* $\mathfrak{clset}(\widehat{\mathbb{C}'}) \cup \mathfrak{prset}(\widehat{\mathbb{C}}) \models \bot$.

It is important to note that for any conflicting pair, each involved coalition of structures guarantees minimality and consistency. Later on we will see how acceptability of arguments benefits from these requirements. Next we exemplify the four different types of conflict that may be recognized from a GenAF following Definition 16.

Example 6 *Let* $\{\mathcal{B}_1, \mathcal{B}_2, \mathcal{B}_3, \mathcal{B}_4, \mathcal{B}_5, \mathcal{B}_6, \mathcal{B}_7\} \subseteq \mathbf{A}$ *where* $\mathcal{B}_1 = \langle\{p_1(x)\}, p_2(x)\rangle$, $\mathcal{B}_2 = \langle\{p_2(x)\}, p_3(x)\rangle$, $\mathcal{B}_3 = \langle\{p_1(x)\}, \neg p_3(x)\rangle$, $\mathcal{B}_4 = \langle\{\neg p_3(x)\}, p_1(x)\rangle$, $\mathcal{B}_5 = \langle\{p_1(x), \neg p_2(x)\}, p_3(x)\rangle$, $\mathcal{B}_6 = \langle\{p_4(x)\}, \neg p_3(x) \vee \neg p_1(x)\rangle$, $\mathcal{B}_7 = \langle\{p_5(x)\}, p_1(x)\rangle$.

(dependency & claim-clash) $\widehat{\mathbb{C}}_1 = \{\{\mathcal{B}_1, \mathcal{B}_2\}\}$ *and* $\widehat{\mathbb{C}}_2 = \{\{\mathcal{B}_3\}\}$.

(dependency & premise-clash) $\widehat{\mathbb{C}}_3 = \{\{\mathcal{B}_1\}\}$ *and* $\widehat{\mathbb{C}}_4 = \{\{\mathcal{B}_5\}\}$.

(support & claim-clash) $\widehat{\mathbb{C}}_1 = \{\{\mathcal{B}_1, \mathcal{B}_2\}\}$ *and* $\widehat{\mathbb{C}}_5 = \{\{\mathcal{B}_6\}, \{\mathcal{B}_7\}\}$.

(support & premise-clash) $\widehat{\mathbb{C}}_1 = \{\{\mathcal{B}_1, \mathcal{B}_2\}\}$ *and* $\widehat{\mathbb{C}}_6 = \{\{\mathcal{B}_4\}\}$.

In order to decide which coalition of structures succeeds from a conflicting pair, an *argument comparison criterion* "$\succcurlyeq$" is assumed to be determined from the comparison criterion among formulae in the KB (see Sect. 2). Afterwards, two conflicting coalitions of structures $\widehat{\mathbb{C}}_1$ and $\widehat{\mathbb{C}}_2$ are assumed to be ordered by a function "$\mathfrak{pref}$" relying on "$\succcurlyeq$", where $\mathfrak{pref}(\widehat{\mathbb{C}}_1, \widehat{\mathbb{C}}_2) = (\widehat{\mathbb{C}}_1, \widehat{\mathbb{C}}_2)$ implies the attack relation $\widehat{\mathbb{C}}_1 \mathbf{R}_{\mathbf{A}} \widehat{\mathbb{C}}_2$, *i.e.*, $\widehat{\mathbb{C}}_1$ is a *defeater of* (or it defeats) $\widehat{\mathbb{C}}_2$. In such a case, $\widehat{\mathbb{C}}_2$ is said to be *defeated*. Moreover, if there is no defeater of $\widehat{\mathbb{C}}_1$ then it is said to be *undefeated*. Note that when no pair of arguments is related by "$\succcurlyeq$", both $\widehat{\mathbb{C}}_1 \mathbf{R}_{\mathbf{A}} \widehat{\mathbb{C}}_2$ and $\widehat{\mathbb{C}}_2 \mathbf{R}_{\mathbf{A}} \widehat{\mathbb{C}}_1$ appear from any conflicting pair $\widehat{\mathbb{C}}_1$ and $\widehat{\mathbb{C}}_2$. Finally, the set $\mathbf{R}_{\mathbf{A}} = \{(\widehat{\mathbb{C}}_1, \widehat{\mathbb{C}}_2) \mid \widehat{\mathbb{C}}_1$ and $\widehat{\mathbb{C}}_2$ are in conflict and $\mathfrak{pref}(\widehat{\mathbb{C}}_1, \widehat{\mathbb{C}}_2) = (\widehat{\mathbb{C}}_1, \widehat{\mathbb{C}}_2)\}$ identifies the *attack relations* from a GenAF $\langle \mathbf{A}, \mathbf{N}\rangle \in \mathbb{G}$.

Theorem 1 *Given a* GenAF $\langle \mathbf{A}, \mathbf{N}\rangle \in \mathbb{G}$*,* $\mathcal{L}_{\mathtt{cl}} = \mathcal{L}_{\mathtt{pr}} = \mathcal{L}_{\mathcal{A}}$ *iff* $\langle A, \hookrightarrow\rangle$ *is a Dung's AF, such that* $A = \{\mathbb{S} \subseteq \mathbf{A} | \ \mathbb{S}$ *is an argumental structure* $\}$ *and* $\hookrightarrow = \{(\mathbb{S}_1, \mathbb{S}_2) \subseteq \mathbf{A} \times \mathbf{A} | (\{\mathbb{S}_1\}, \{\mathbb{S}_2\}) \in \mathbf{R}_{\mathbf{A}}\}$.

4.2 Acceptability Analysis

Assuming a set of normality conditions $\mathbf{N}$, an inconsistent KB Σ leads to conflicting arguments within the associated $\mathfrak{genaf}(\Sigma, \mathbf{N}) = \langle \mathbf{A}, \mathbf{N} \rangle$. Thus, each minimal source of inconsistency within Σ is reflected as an attack in the resulting GenAF. Since the objective of a GenAF is to reason about a KB under uncertainty, there is a need for a mechanism that allows us to obtain those arguments that prevail over the rest. That is, those arguments that can be consistently assumed together, following some policy. For instance, structures with no defeaters should always prevail, since there is nothing strong enough to be posed against them. The tool we need to resolve inconsistency is the notion of *acceptability of arguments*, which is defined on top of an *argumentation semantics* [3]. There are several well-known argumentation semantics, such as the grounded, the stable, and the preferred semantics [6]. These semantics ensure the obtention of consistent sets of arguments, namely *extensions*. That is, the set of accepted arguments calculated following any of these semantics is such that no pair of conflicting arguments appears in that same extension. Finally, an extension determines a maximal consistent subset of the KB Σ.

It is important to notice that some problems like multiple extensions may arise from semantics like both the *stable* and the *preferred*. This would require to make a choice among them. On the other hand, the outcome of the *grounded semantics* is always a single extension, which could be empty. Finally, since dealing with multiple extensions is a problem that falls outside the scope of this article, we will choose the grounded semantics, which can be implemented with a simple algorithm. Consequently, we define a mapping $\mathfrak{sem} : \mathbb{G} \longrightarrow 2^{\mathbb{A}\mathfrak{rgs}}$, that intuitively behaves as follows. The set $X \subseteq \mathbf{A}$ is the minimal set verifying $X \subseteq \bigcup_{(\widehat{\mathbb{C}'}, \widehat{\mathbb{C}}) \in \mathbf{R_A}} \mathbb{C}^*$ for every undefeated $\widehat{\mathbb{C}'}$ defeating $\widehat{\mathbb{C}}$, and for each $\widehat{\mathbb{C}}$ it follows $\mathbb{C}^* \cap X \neq \emptyset$. As a result, other coalition of structures defeated by $\widehat{\mathbb{C}}$ could appear undefeated. Thus, this process is iteratively applied over the set of arguments $\mathbf{A}' = \mathbf{A} \setminus (X \cup X')$, where X' contains the arguments which are counterpositive to those in X, until no conflicting pair is identified. Finally, the extension $\mathbf{A}'$ of the GenAF is determined.

As stated before, the outcome of a grounded semantics could be an empty extension. Such an issue arises when there is a loop in the structures attack graph, that is $(\widehat{\mathbb{C}'}, \widehat{\mathbb{C}}) \in \mathbf{R_A}$ and $(\widehat{\mathbb{C}}, \widehat{\mathbb{C}'}) \in \mathbf{R_A}$. To overcome this, some argument from either $\widehat{\mathbb{C}}$ or $\widehat{\mathbb{C}'}$ could be included in X, and therefore the loop would be broken, and the process determined by applying "$\mathfrak{sem}$" can be reconsidered. Finally, a non-empty conflict-free extension is obtained.

Proposition 2 *Given a* GenAF $\langle \mathbf{A}, \mathbf{N} \rangle \subseteq \mathbb{G}$*, the* GenAF $\langle \mathbf{A}', \mathbf{N} \rangle$*, where* $\mathbf{A}' = \mathfrak{sem}(\langle \mathbf{A}, \mathbf{N} \rangle)$*, is conflict free, i.e.,* $\mathbf{R}_{\mathbf{A}'} = \emptyset$*.*

Given a (potentially inconsistent) pANF knowledge base $\Sigma \subseteq \mathcal{L}^{\kappa}$, and a set of normality conditions $\mathbf{N} \subseteq \mathbb{N}\mathtt{orm}$, it is possible to redefine the notion of entailment "$\models$" from Σ by reasoning about it over its associated GenAF $\mathfrak{genaf}(\Sigma, \mathbf{N})$, such that $\Sigma \models_G \alpha$ *iff* there exists an argumental structure $\mathbb{S}$ for α such that $\mathbb{S} \subseteq \mathfrak{sem}(\mathfrak{genaf}(\Sigma, \mathbf{N}))$. In such a case, the inferrable claim α is said to be *warranted* and therefore, $\Sigma \models_G \alpha$ holds. Note that if Σ is consistent and $\alpha \in \mathcal{L}_{\mathtt{cl}}$, "$\models_G$" equals the classical entailment "$\models$".

Theorem 2 *Given a consistent* pANF *knowledge base* $\Sigma \subseteq \mathcal{L}^{\kappa}$*, a set of normality conditions* $\mathbf{N} \subseteq \mathbb{N}\mathtt{orm}$*, and a formula* $\alpha \in \mathcal{L}_{\mathtt{cl}}$*,* $\Sigma \models \alpha$ *iff* $\Sigma \models_G \alpha$*.*

5 Ontology Debugging Through the GenAF

In what follows we propose a reification of $\mathcal{L}^{\kappa}$ to the description logic $\mathcal{ALC}$ in order to obtain the generalized GenAF associated to $\mathcal{ALC}$ ontologies. Afterwards, since some basic elements from argumentation like attack and support may allow to manage inconsistencies in ontologies, we propose an acceptability semantics for arguments in order to obtain a related maximal consistent ontology.

5.1 $\mathcal{ALC}$ Overview

Before presenting the reification subtleties, a very brief overview of the $\mathcal{ALC}$ DL will be given, for more detailed information refer to [2]. An interpretation $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$ consists of a nonempty domain $\Delta^{\mathcal{I}}$, and

an interpretation function $\cdot^{\mathcal{I}}$ that maps every concept to a subset of $\Delta^{\mathcal{I}}$, every role to a subset of $\Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$, and every individual to an element of $\Delta^{\mathcal{I}}$. Symbols $A, A_1, A_2, \ldots$ and $B, B_1, B_2, \ldots$ are used to denote atomic DL concepts, $C, C_1, C_2, \ldots$ and $D, D_1, D_2, \ldots$, to denote general DL concepts, and $R, R_1, R_2, \ldots$, to denote atomic DL roles. The description language $\mathcal{ALC}$ is formed by concept definitions according to the syntax $C, D ::= A|\bot|\top|\neg C|C \sqcap D|C \sqcup D|\forall R.C|\exists R.C$ where the interpretation function $\cdot^{\mathcal{I}}$ is extended to the universal concept as $\top^{\mathcal{I}} = \Delta^{\mathcal{I}}$; the bottom concept as $\bot^{\mathcal{I}} = \emptyset$; the full negation or complement as $(\neg C)^{\mathcal{I}} = \Delta^{\mathcal{I}} \backslash C^{\mathcal{I}}$; the intersection as $(C \sqcap D)^{\mathcal{I}} = C^{\mathcal{I}} \cap D^{\mathcal{I}}$; the union as $(C \sqcup D)^{\mathcal{I}} = C^{\mathcal{I}} \cup D^{\mathcal{I}}$; the universal quantification as $(\forall R.C)^{\mathcal{I}} = \{a \in \Delta^{\mathcal{I}} | \forall b.(a, b) \in R^{\mathcal{I}} \rightarrow b \in C^{\mathcal{I}}\}$; and the full existential quantification as $(\exists R.C)^{\mathcal{I}} = \{a \in \Delta^{\mathcal{I}} | \exists b.(a, b) \in R^{\mathcal{I}} \wedge b \in C^{\mathcal{I}}\}$.

An ontology is a pair $\mathcal{O} = \langle \mathcal{T}, \mathcal{A} \rangle$, where $\mathcal{T}$ represents the TBox, containing the terminologies (or axioms) of the application domain, and $\mathcal{A}$, the ABox, which contains assertions about named individuals in terms of these terminologies. Regarding the TBox $\mathcal{T}$, axioms are sketched as $C \sqsubseteq D$ and $C \equiv D$, therefore, an interpretation $\mathcal{I}$ satisfies them whenever $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$ and $C^{\mathcal{I}} = D^{\mathcal{I}}$ respectively. An interpretation $\mathcal{I}$ is a model for the TBox $\mathcal{T}$ if $\mathcal{I}$ satisfies all the axioms in $\mathcal{T}$. Thus, the TBox $\mathcal{T}$ is said to be satisfiable if it admits a model. Besides, in the ABox $\mathcal{A}$, $\mathcal{I}$ satisfies $C(a)$ if $a \in C^{\mathcal{I}}$, and $R(a, b)$ if $(a, b) \in R^{\mathcal{I}}$. An interpretation $\mathcal{I}$ is said to be a model of the ABox $\mathcal{A}$ if every assertion of $\mathcal{A}$ is satisfied by $\mathcal{I}$. Hence, the ABox $\mathcal{A}$ is said to be satisfiable if it admits a model. Finally, regarding the entire ontology, an interpretation $\mathcal{I}$ is said to be a model of $\mathcal{O}$ if every statement in $\mathcal{O}$ is satisfied by $\mathcal{I}$, and $\mathcal{O}$ is said to be satisfiable if it admits a model.

The different classes of inconsistencies in an ontology are defined through the usual meaning of *inconsistency* in classical logic, along with the notion of *incoherency* presented in [7]. That is, an ontology $\mathcal{O}$ is *inconsistent iff* it admits no model; on the other hand, the ontology $\mathcal{O}$ is *incoherent iff* there exists an unsatisfiable concept C in $\mathcal{O}$. In addition, a concept C is *unsatisfiable iff* for each interpretation $\mathcal{I} \in \mathcal{M}(\mathcal{O})$, it follows that $C^{\mathcal{I}} = \emptyset$ holds. Though incoherency is considered a kind of inconsistency in the TBox, it does not replace the usual notion of inconsistency, given that an incoherent ontology may admit models.

An ontology contains implicit knowledge that is made explicit through inferences. The notion of semantic entailment is given by $\mathcal{O} \models \alpha$, meaning that every model of the ontology $\mathcal{O}$ is also a model of the statement α. Formally, **(semantic entailment)** $\mathcal{O} \models \alpha$ *iff* $\mathcal{M}(\mathcal{O}) \subseteq \mathcal{M}(\{\alpha\})$. Just for simplicity, we shall abuse notation writing $\mathcal{O} = \mathcal{T} \cup \mathcal{A}$ (*eg.*, $\mathcal{O} = \{C \sqsubseteq D, A(a)\}$) to identify an ontology $\mathcal{O} = \langle \mathcal{T}, \mathcal{A} \rangle$ (*eg.*, $\mathcal{O} = \langle \{C \sqsubseteq D\}, \{A(a)\} \rangle$).

5.2 Specifying the $\mathcal{ALC}$ GenAF

The following grammars are proposed in order to specify the argument language used to represent $\mathcal{ALC}$-based ontologies into a `GenAF`, which will be referred as $\mathcal{ALC}$-`GenAF`. This basis is currently leading the investigations towards handling other more expressive DLs through a `GenAF`.

$$\phi ::= \top|A|\neg A|\forall R.\mathcal{L}_{disj}|\exists R.\mathcal{L}_{conj}$$

$$\mathcal{L}_{conj} ::= \phi|\mathcal{L}_{conj} \sqcap \mathcal{L}_{conj} \qquad \mathcal{L}_{\texttt{pr}} ::= \phi(\mathcal{L}_{var})$$
$$\mathcal{L}_{disj} ::= \phi|\mathcal{L}_{disj} \sqcup \mathcal{L}_{disj} \qquad \mathcal{L}_{\texttt{cl}} ::= \mathcal{L}_{disj}(\mathcal{L}_{var})|R(\mathcal{L}_{var}, \mathcal{L}_{var})$$
$$\mathcal{L}_{var} ::= a|b|x|y$$

In order to obtain a `GenAF` from an $\mathcal{ALC}$ ontology $\mathcal{O}$, it is needed to translate each axiom in $\mathcal{O}$ to *negation normal form*, so that negation appears only in front of atomic concepts. Afterwards, each axiom should turn to *disjunctive normal form* for the left-hand-side (*lhs*) part of the description, and to *conjunctive normal form* for its right-hand-side (*rhs*), conforming axioms like $lhs \sqsubseteq rhs$, where $lhs ::= \bot|\mathcal{L}_{conj} \sqcup \ldots \sqcup \mathcal{L}_{conj}$ and $rhs ::= \bot|\mathcal{L}_{disj} \sqcap \ldots \sqcap \mathcal{L}_{disj}$. An ontology containing this kind of axioms can be easily reformatted into a `pANF` ontology (see Definition 3), or straightforwardly, multiple arguments could be triggered from an axiom $lhs \sqsubseteq rhs$. That is, each *lhs* disjunction (in $\mathcal{L}_{conj}$) is interpreted as a set of premises $\mathcal{L}_{\texttt{pr}}$–one for each conjunction– and each *rhs* conjunction (in $\mathcal{L}_{disj}$), as a claim in $\mathcal{L}_{\texttt{cl}}$. This methodology is illustrated by the following example.

Example 7 *Given the $\mathcal{ALC}$ axiom $(A_1 \sqcap A_2) \sqcup (\forall R_1.A_3 \sqcap \exists R_2.\forall R_3.\neg A_4) \sqsubseteq (A_1 \sqcup A_2) \sqcap A_5$, eight arguments appear in the related* `GenAF`*:*

- $\langle\{A_1(x), A_2(x)\}, (A_1 \sqcup A_2)(x)\rangle$,
- $\langle\{(\forall R_1.A_3)(x),\ (\exists R_2.\forall R_3.\neg A_4)(x)\}, (A_1 \sqcup A_2)(x)\rangle$,
- $\langle\{A_1(x), A_2(x)\},\ A_5(x)\rangle$, *and*
- $\langle\{(\forall R_1.A_3)(x),\ (\exists R_2.\forall R_3.\neg A_4)(x)\},\ A_5(x)\rangle$,

and regarding the counterpositive formula,

- $\langle\{\neg A_1(x), \neg A_2(x)\}, (\neg A_1 \sqcup \neg A_2)(x)\rangle$,
- $\langle\{\neg A_1(x), \neg A_2(x)\}, (\exists R_1.\neg A_3 \sqcup \forall R_2.\exists R_3.A_4)(x)\rangle$,
- $\langle\{\neg A_5(x)\}, (\neg A_1 \sqcup \neg A_2)(x)\rangle$, *and*
- $\langle\{\neg A_5(x)\}, (\exists R_1.\neg A_3 \sqcup \forall R_2.\exists R_3.A_4)(x)\rangle$.

Some particularities appear for $\mathcal{ALC}$ DLs that require to be attended by following additional conventions: concept equivalences as $C_1 \equiv C_2$, are assumed as pairs $C_1 \sqsubseteq C_2$ and $C_2 \sqsubseteq C_1$; and inclusions $\bot \sqsubseteq C$ and $C \sqsubseteq \bot$, are assumed as $\neg C \sqsubseteq \top$ and $\top \sqsubseteq \neg C$, respectively, given that arguments cannot accept $\bot$ in any of their components (*c.f. consistency* in Definition 4). Finally, any assertion $A(a)$ (resp., $R(a, b)$) triggers an evidence $\langle\{\}, A(a)\rangle$ (resp., $\langle\{\}, R(a, b)\rangle$).

We define as $\mathcal{ALC}^{\mathbb{A}}$ to the logic for ontologies $\mathcal{O} \subseteq \mathcal{L}_{\mathcal{T}} \times \mathcal{L}_{\mathcal{A}}$, using $\mathcal{L}_{\mathcal{T}} ::= \mathcal{L}_{conj} \sqsubseteq \mathcal{L}_{disj}$ for axioms and $\mathcal{L}_{\mathcal{A}} ::= A(\mathcal{L}_{var})|\neg A(\mathcal{L}_{var})|R(\mathcal{L}_{var}, \mathcal{L}_{var})$ for assertions. It is clear that any $\mathcal{ALC}^{\mathbb{A}}$ ontology conforms the $\mathcal{ALC}$ DL, and it is always in $\mathtt{pANF}$.

Proposition 3 *Any $\mathcal{ALC}^{\mathbb{A}}$ ontology $\mathcal{O}$ is an $\mathcal{ALC}$ ontology conforming to* $\mathtt{pANF}$.

We will assume a function $\mathfrak{panf} : 2^{\mathcal{ALC}} \longrightarrow 2^{\mathcal{ALC}^{\mathbb{A}}}$, referred as *panf-DL function*, to translate any $\mathcal{ALC}$ ontology $\mathcal{O}$ into an equivalent $\mathcal{ALC}^{\mathbb{A}}$ ontology $\mathfrak{panf}(\mathcal{O})$. A desirable property of an $\mathcal{ALC}^{\mathbb{A}}$ ontology is that each statement in it generates a single argument (actually, a pair of arguments, if it is the case that its counterpositive falls within the language) in its related $\mathtt{GenAF}$. This statement holds except for unsatisfiable concept inclusions as $A \sqsubseteq \neg A$, which are filtered by *consistency* in Definition 4 –triggering no related argument in the $\mathtt{GenAF}$.

Proposition 4 *For any $\mathcal{ALC}$ ontology $\mathcal{O}$, the $\mathcal{ALC}^{\mathbb{A}}$ ontology $\mathfrak{panf}(\mathcal{O})$ is equivalent to $\mathcal{O}$.*

Given an $\mathcal{ALC}$ ontology $\mathcal{O}$ and a set of normality conditions $\mathbf{N} \subseteq \mathbb{N}\mathtt{orm}$, the associated $\mathtt{GenAF}$ $\langle\mathbf{A}, \mathbf{N}\rangle$ is obtained by means of the theory function "$\mathfrak{genaf}$" (see Definition 7) and the panf-DL function "$\mathfrak{panf}$" such that:

$$\langle\mathbf{A}, \mathbf{N}\rangle = \mathfrak{genaf}(\mathfrak{panf}(\mathcal{O}), \mathbf{N})$$

Note that cyclic terminologies like $A \equiv B$ will not be part of any structure due to acyclicity in Definition 13. An analogous case in FOL was illustrated in Example 4. The same situation occurs with axioms like $A \sqsubseteq A$.

5.3 Argumentation Semantics as an Ontology Debugging Tool

In an ontology, inconsistency implies that there are contradictory concept definitions, or assertions that will lead to conflicting arguments within the related $\mathtt{GenAF}$. Thus, once the translation is performed, each inconsistency in the original ontology will be reflected as an attack in the resulting $\mathtt{GenAF}$. Consequently, we can benefit from the argumentation semantics presented in Section 4.2 to repair $\mathcal{ALC}$ ontologies restoring their consistency. Hence, when we translate an ontology to a $\mathtt{GenAF}$, all what is left to do to resolve inconsistencies is to calculate the set of accepted arguments following the semantics specified by the function "$\mathfrak{sem}$", which will be afterwards translated back to a consistent/coherent ontology. For such matter, we assume a function $\mathfrak{ont} : 2^{\mathbb{A}\mathtt{rgs}} \longrightarrow 2^{\mathcal{ALC}}$ mapping a set of $\mathcal{ALC}$ arguments $\mathbf{A} \subseteq \mathbb{A}\mathtt{rgs}$ from a $\mathtt{GenAF}$ $\langle\mathbf{A}, \mathbf{N}\rangle \subseteq \mathbb{G}$ to an $\mathcal{ALC}$ ontology $\mathfrak{ont}(\mathbf{A})$. The specification of the function "$\mathfrak{ont}$" will be left out from this article, nonetheless, observe that its intuition follows backwards the ones given to obtain a $\mathtt{GenAF}$ by the theory function "$\mathfrak{genaf}$". Proposition 5 relates consistency/coherency of the $\mathfrak{ont}$-outcome to the attacks in the $\mathtt{GenAF}$.

Proposition 5 *Given an $\mathcal{ALC}$ GenAF $\langle \mathbf{A}, \mathbf{N} \rangle \subseteq \mathbb{G}$, $\mathbf{R_A} = \emptyset$ iff $\mathfrak{ont}(\mathbf{A})$ is a consistent-coherent $\mathcal{ALC}$ ontology.*

The relation stated in Proposition 5 along with that in Proposition 2 motivates the following result.

Lemma 3 *Given an $\mathcal{ALC}$ GenAF $\langle \mathbf{A}, \mathbf{N} \rangle \subseteq \mathbb{G}$, $\mathfrak{ont}(\mathfrak{sem}(\langle \mathbf{A}, \mathbf{N} \rangle))$ is a consistent-coherent $\mathcal{ALC}$ ontology.*

Next we define the *debugging function* as the mapping of an $\mathcal{ALC}$ ontology to a consistent-coherent $\mathcal{ALC}$ ontology.

Definition 17 (Ontology Debugging) *Given an $\mathcal{ALC}$ ontology $\mathcal{O}$ and a set of normality conditions $\mathbf{N} \subseteq \mathbb{N}\mathtt{orm}$, the **debugging function** $\mathfrak{debug} : 2^{\mathcal{ALC}} \longrightarrow 2^{\mathcal{ALC}}$ is defined as follows:*

$$\mathfrak{debug}(\mathcal{O}) = \mathfrak{ont}(\mathfrak{sem}(\mathfrak{genaf}(\mathfrak{panf}(\mathcal{O}), \mathbf{N}))).$$

Theorem 3 states the main contribution of the $\mathcal{ALC}$ GenAF regarding ontology debugging by means of Lemma 3 and Definition 17. Afterwards, Corollary 1 relates that result through "$\mathfrak{panf}$".

Theorem 3 *Given an $\mathcal{ALC}$ ontology $\mathcal{O}$, the $\mathcal{ALC}$ ontology $\mathfrak{debug}(\mathcal{O})$ is consistent-coherent.*

Corollary 1 *For any $\mathcal{ALC}$ ontology $\mathcal{O}$, it follows that $\mathfrak{debug}(\mathcal{O}) \subseteq \mathfrak{panf}(\mathcal{O})$ holds.*

The following examples are provided to illustrate the methodology proposed for ontology debugging.

Example 8 *Let $\mathcal{O} = \{A_1 \sqsubseteq B_1 \sqcap B_2,\ A_2 \sqsubseteq A_1 \sqcap \neg B_2,\ A_1(a),\ B_1(a),\ \neg B_2(a),\ A_2(a)\}$ be an $\mathcal{ALC}$ ontology, we want to debug $\mathcal{O}$ to obtain a related consistent-coherent ontology $\mathcal{O}^R$. Assuming set of normality conditions $\mathbf{N} \subseteq \mathbb{N}\mathtt{orm}$, and applying $\mathfrak{genaf}(\mathfrak{panf}(\mathcal{O}), \mathbf{N})$, a GenAF $\langle \mathbf{A}, \mathbf{N} \rangle$ appears:*

Statement	*Args.*
$A_1 \sqsubseteq B_1 \sqcap B_2$	$\{\mathcal{B}_1, \mathcal{B}_2\}$
$A_2 \sqsubseteq A_1 \sqcap \neg B_2$	$\{\mathcal{B}_3, \mathcal{B}_4\}$
$A_1(a)$	$\{\mathcal{B}_5\}$
$B_1(a)$	$\{\mathcal{B}_6\}$
$\neg B_2(a)$	$\{\mathcal{B}_7\}$
$A_2(a)$	$\{\mathcal{B}_8\}$

$$\mathbf{A} = \left\{ \begin{array}{l} \mathcal{B}_1 = \langle \{A_1(x)\}, B_1(x) \rangle \\ \mathcal{B}_2 = \langle \{A_1(x)\}, B_2(x) \rangle \\ \mathcal{B}_3 = \langle \{A_2(x)\}, A_1(x) \rangle \\ \mathcal{B}_4 = \langle \{A_2(x)\}, \neg B_2(x) \rangle \\ \mathcal{B}_5 = \langle \{\}, A_1(a) \rangle \\ \mathcal{B}_6 = \langle \{\}, B_1(a) \rangle \\ \mathcal{B}_7 = \langle \{\}, \neg B_2(a) \rangle \\ \mathcal{B}_8 = \langle \{\}, A_2(a) \rangle \end{array} \right\}$$

Consider the structures $\mathbb{S}_1 = \{\mathcal{B}_3, \mathcal{B}_2\}$, $\mathbb{S}_2 = \{\mathcal{B}_8\} \cup \mathbb{S}_1$, $\mathbb{S}_3 = \{\mathcal{B}_8, \mathcal{B}_4\}$, and $\mathbb{S}_4 = \{\mathcal{B}_5, \mathcal{B}_2\}$; and the primitive structures $\mathbb{S}_5 = \{\mathcal{B}_4\}$ and $\mathbb{S}_6 = \{\mathcal{B}_7\}$. Assuming $\mathcal{B}_2 \succcurlyeq \mathcal{B}_4$ and $\mathcal{B}_2 \succcurlyeq \mathcal{B}_7$, the attack relation set is $\mathbf{R_A} = \{(\{\mathbb{S}_1\}, \{\mathbb{S}_5\}), (\{\mathbb{S}_2\}, \{\mathbb{S}_6\}), (\{\mathbb{S}_4\}, \{\mathbb{S}_6\}), (\{\mathbb{S}_4\}, \{\mathbb{S}_3\})\}$ (see the graph depicted below in Figure 1). Note that $(\{\mathbb{S}_2\}, \{\mathbb{S}_3\})$ is not in $\mathbf{R_A}$ given that $\mathbb{S}_1 \trianglelefteq \mathbb{S}_2$, $\mathbb{S}_5 \trianglelefteq \mathbb{S}_3$, and $(\{\mathbb{S}_1\}, \{\mathbb{S}_5\}) \in \mathbf{R_A}$ (c.f. Definition 16).

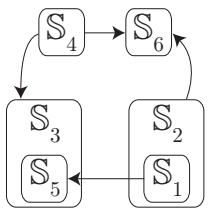


Figure 1: Graph resulting from $\mathbf{R_A}$.

The acceptability analysis determines $\mathbb{S}_3$, $\mathbb{S}_5$, and $\mathbb{S}_6$ to be removed, and since $\mathbb{S}_5 \trianglelefteq \mathbb{S}_3$, removing $\mathcal{B}_4$ and $\mathcal{B}_7$ is enough. Afterwards, through the function "$\mathfrak{sem}$" the set of arguments $\{\mathcal{B}_1, \mathcal{B}_2, \mathcal{B}_3, \mathcal{B}_5, \mathcal{B}_6, \mathcal{B}_8\}$ determines a conflict-free extension. Hence, through the debugging function $\mathfrak{debug}(\mathcal{O})$ a repaired ontology $\mathcal{O}^R$ is contructed such that:

$$\mathcal{O}^R = \{A_1 \sqsubseteq B_1, A_1 \sqsubseteq B_2,\ A_2 \sqsubseteq A_1,\ A_1(a),\ B_1(a),\ A_2(a)\}$$

Example 9 *Considering the specifications given in Example 8, if we assume $\mathcal{B}_7 \succcurlyeq \mathcal{B}_2 \succcurlyeq \mathcal{B}_4$, conflicts involving the structure $\mathbb{S}_6$ are inverted leading to $(\{\mathbb{S}_6\}, \{\mathbb{S}_2\})$ and $(\{\mathbb{S}_6\}, \{\mathbb{S}_4\})$. In such a case, only $\mathcal{B}_2$ would be left out of the corresponding extension, thus, the repaired ontology $\mathfrak{debug}(\mathcal{O}) = \mathcal{O}^R$ would end up being:*

$$\mathcal{O}^R = \{A_1 \sqsubseteq B_1,\ A_2 \sqsubseteq A_1,\ A_2 \sqsubseteq \neg B_2,\ A_1(a),\ B_1(a),\ \neg B_2(a),\ A_2(a)\}$$

Notice that by following the table relating statements and arguments in Example 8, a reconstruction procedure can be specified in order to obtain a repaired ontology containing the axiom $A_1 \sqsubseteq B_1 \sqcap B_2$ in place of the pair $A_1 \sqsubseteq B_1$ and $A_1 \sqsubseteq B_2$, for the case of Example 8; and analogously, the axiom $A_2 \sqsubseteq A_1 \sqcap \neg B_2$ in place of $A_2 \sqsubseteq A_1$ and $A_2 \sqsubseteq \neg B_2$, for Example 9. This motivates the analisys of a procedure which would take the resulting repaired ontology $\mathfrak{debug}(\mathcal{O})$ to generate an equivalent ontology $\mathcal{O}'$ guaranteing not only $\mathcal{O}' \subseteq \mathfrak{panf}(\mathcal{O})$ (as being stated by Corollary 1) but also $\mathcal{O}' \subseteq \mathcal{O}$. Observe nonetheless that this property cannot be verified when part of an axiom is removed: $A_2 \sqsubseteq \neg B_2$ in Example 8 –where the original axiom was $A_2 \sqsubseteq A_1 \sqcap \neg B_2$– and $A_1 \sqsubseteq B_2$ in Example 9 –where the original axiom was $A_1 \sqsubseteq B_1 \sqcap B_2$. In each of these cases, in order to guarantee $\mathcal{O}' \subseteq \mathcal{O}$, the alternative is to remove the complete original axiom; however, such decision would play in detriment of the intention to repair ontologies following some minimal change criterion. We will not go further into this subject.

In the following example we illustrate the debugging of an $\mathcal{ALC}$ ontology along with the interesting case of reintroduction of arguments within a supporting-chain due to different variables instantiantions.

Example 10 *Given the $\mathcal{ALC}$ ontology $\mathcal{O} = \{R(a,b),\ R(b,c),\ R(c,d),\ A(a),\ \neg A(c),\ \neg A(d),\ A \sqsubseteq \forall R.A\}$, the $\mathcal{ALC}$ GenAF $\langle \mathbf{A}, \mathbf{N} \rangle$ is determined as $\mathbf{A} = \{\mathcal{B}_1,\ \mathcal{B}_2,\ \mathcal{B}_3,\ \mathcal{B}_4,\ \mathcal{B}_5,\ \mathcal{B}_6,\ \mathcal{B}_7\}$, where $\mathcal{B}_1 = \langle \{\}, R(a,b) \rangle$, $\mathcal{B}_2 = \langle \{\}, R(b,c) \rangle$, $\mathcal{B}_3 = \langle \{\}, R(c,d) \rangle$, $\mathcal{B}_4 = \langle \{\}, A(a) \rangle$, $\mathcal{B}_5 = \langle \{\}, \neg A(c) \rangle$, $\mathcal{B}_6 = \langle \{\}, \neg A(d) \rangle$, and $\mathcal{B}_7 = \langle \{A(x)\}, (\forall R.A)(x) \rangle$.*

The argumental structure $\mathbb{S}_1 = \{\mathcal{B}_4, \mathcal{B}_7\}$ appears. Later on, the set $\widehat{\mathcal{C}}_1 = \{\mathcal{B}_7, \mathcal{B}_1\}$ is a supporting-coalition of $\mathcal{B}_7$ through $A(b)$, $\widehat{\mathcal{C}}_2 = \{\mathcal{B}_7, \mathcal{B}_2\}$ is a supporting-coalition of $\mathcal{B}_7$ through $A(c)$, and $\widehat{\mathcal{C}}_3 = \{\mathcal{B}_7, \mathcal{B}_3\}$ is a supporting-coalition of $\mathcal{B}_7$ through $A(d)$. Note that the set $\{\mathcal{B}_7, \mathcal{B}_1\}$ may conform two different structures: one for $A(b)$ if its supporting-chain is constituted by a single coalition $\{\mathcal{B}_7, \mathcal{B}_1\}$, and another one for $(\forall R.A)(b)$ if its supporting-chain is $\{\mathcal{B}_7\}\{\mathcal{B}_7, \mathcal{B}_1\}$.

We will work with the schematic structures $\mathbb{S}_2 = \{\mathcal{B}_7, \mathcal{B}_1\}$, $\mathbb{S}_3 = \{\mathcal{B}_7, \mathcal{B}_2\}$, and $\mathbb{S}_4 = \{\mathcal{B}_7, \mathcal{B}_3\}$, with $\mathfrak{cl}(\mathbb{S}_2) = (\forall R.A)(b)$, $\mathfrak{cl}(\mathbb{S}_3) = (\forall R.A)(c)$, and $\mathfrak{cl}(\mathbb{S}_4) = (\forall R.A)(d)$; and premises $\mathfrak{pr}(\mathbb{S}_2) = \{A(a)\}$, $\mathfrak{pr}(\mathbb{S}_3) = \{A(b)\}$, and $\mathfrak{pr}(\mathbb{S}_4) = \{A(c)\}$. Thus, we identify the related argumental structures $\mathbb{S}_5 = \{\mathcal{B}_4, \mathcal{B}_7, \mathcal{B}_1\}$, $\mathbb{S}_6 = \{\mathcal{B}_4, \mathcal{B}_7, \mathcal{B}_1, \mathcal{B}_2\}$, and $\mathbb{S}_7 = \{\mathcal{B}_4, \mathcal{B}_7, \mathcal{B}_1, \mathcal{B}_2, \mathcal{B}_3\}$, where $\mathbb{S}_2 \trianglelefteq \mathbb{S}_5$, $\mathbb{S}_3 \trianglelefteq \mathbb{S}_6$, and $\mathbb{S}_4 \trianglelefteq \mathbb{S}_7$, as well as $\mathbb{S}_5 \trianglelefteq \mathbb{S}_6$, and $\mathbb{S}_6 \trianglelefteq \mathbb{S}_7$. Note also that, $\mathbb{S}_1$ is sub-structure of $\mathbb{S}_5$, $\mathbb{S}_6$, and $\mathbb{S}_7$. Besides, from $\mathbb{S}_6$, the supporting-chain for $(\forall R.A)(c)$ is $\{\mathcal{B}_7\}\{\mathcal{B}_7, \mathcal{B}_2\}\{\mathcal{B}_7, \mathcal{B}_1\}\{\mathcal{B}_4\}$.

Consider now the coalitions of structures $\widehat{\mathbb{C}}_1 = \{\{\mathcal{B}_2\}, \{\mathcal{B}_5\}\}$, and $\widehat{\mathbb{C}}_2 = \{\{\mathcal{B}_3\}, \{\mathcal{B}_6\}\}$. Assuming $\mathcal{B}_7 \succcurlyeq \mathcal{B}_i$, $i \in \{1, \ldots, 6\}$, the following attack relations appear: $\{\mathbb{S}_5\}\mathbf{R}_{\mathbf{A}}\widehat{\mathbb{C}}_1$ and $\{\mathbb{S}_6\}\mathbf{R}_{\mathbf{A}}\widehat{\mathbb{C}}_2$ (refer to Fig. 2). Later on, considering also the coalitions of structures $\widehat{\mathbb{C}}_3 = \{\mathbb{S}_5, \{\mathcal{B}_2\}\}$, and $\widehat{\mathbb{C}}_4 = \{\mathbb{S}_6, \{\mathcal{B}_3\}\}$, the attack relations $\widehat{\mathbb{C}}_3\mathbf{R}_{\mathbf{A}}\{\{\mathcal{B}_5\}\}$ and $\widehat{\mathbb{C}}_4\mathbf{R}_{\mathbf{A}}\{\{\mathcal{B}_6\}\}$ appear (see Figure 2).

Afterwards, the acceptability analysis determines coalitions $\widehat{\mathbb{C}}_1$, $\widehat{\mathbb{C}}_2$, $\{\{\mathcal{B}_5\}\}$, and $\{\{\mathcal{B}_6\}\}$ to remove. Later on, the removal of $\mathcal{B}_5$ and $\mathcal{B}_6$ eliminates every attack, then the function "$\mathfrak{sem}$" determines the set of arguments $\{\mathcal{B}_1, \mathcal{B}_2, \mathcal{B}_3, \mathcal{B}_4, \mathcal{B}_7\}$ as the conflict-free extension. Finally, the operation $\mathfrak{debug}(\mathcal{O})$ constructs the debugged ontology $\mathcal{O}^R = \{A \sqsubseteq \forall R.A,\ R(a,b),\ R(b,c),\ R(c,d),\ A(a)\}$.

6 Related and Future Work

Debugging of terminologies is usually focused on the recognition of sources of concept-unsatisfiability. The union of conflictive coalitions of structures presented in this work, may be related to constructions like *minimal inconsistent preserving sub-terminologies* (MIPS) [26], which have been previously used to cope with ontology debugging [25] and change [20]. MIPS may be also related to works in ontology integration [11], and debugging like [12], where *maximally concept-satisfiable subsets* (MCSS) were proposed for that matter.

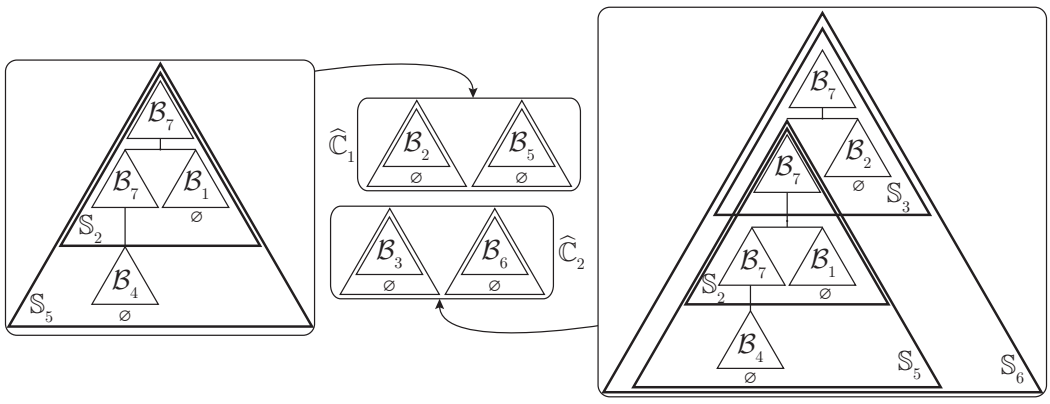


Figure 2: Some attacks from Example 10. Multiple occurrences of an argument within a structure refer to its different instances determined by every possible variable substitution.

It is interesting to extend this proposal beyond the scope of ontology debugging within ontology change. For instance, ontology evolution could benefit from this approach: $\perp$-Kernel Sets [8] (minimal sets inferring $\perp$) may also be related to the union of conflictive coalitions of structures. Moreover, in works like [22, 20, 13], incision functions are used to cut the appropriate piece of knowledge from every kernel such that every source of inconsistency would disappear. In that sense, the function "`sem`" removes the appropriate argument from each attack in order to eliminate every possible argument conflict from the `GenAF`, just like incision functions do.

Further implementations of the model here presented could be done (1) as a module to be incorporated to the DL reasoner, or (2) as a DL-argumentation reasoner. For the second option, a DL reasoner based on argumentation could be an interesting alternative to those like RACER, FaCT, and FaCT++. Besides the negotiation based approach presented in [21], to our knowledge the only approaches of reasoning about DLs based on argumentation are the one presented in [18] –based on dialectical argumentation– and the preliminary version [14] of the model presented here. An pure argumentative approach of ontology reasoning would cope "on the fly" with the decision of what to keep or discard from different sources of information without applying any changes to them. Moreover, an ontology may keep inconsistencies leaving its resolution up to the argumentation reasoning process, that is, the ontology reasoning machinery would manage to dynamically handle the inconsistency. This exposes an interesting proposal to incorporate to the semantic web the most characteristic feature of argumentation reasoners: to keep inconsistency while managing to reason on top of it. More on this latter subject may be referred to [18].

Since the approach here presented was constructed as an extension of the widely accepted `AF` [6], it could also benefit from other `AF`'s extensions like the *dynamic abstract argumentation framework* (`DAF`) preliminarily introduced in [23], and formalized afterwards in [24]. The `DAF` has the objective of dealing with dynamics of abstract argumentation frameworks, and therefore it seems interesting to extend the `GenAF` to a generalized-dynamic argumentation framework in order to deal with the problem of ontology evolution. Besides, a dialectical model of change over `DAF`'s was recently proposed in [23, 16], under the name of *argument theory change* (ATC), and reified to defeasible logic programming in [15]. Properties and methodologies from the ATC model of change could also be adapted to formalize a generalized-dynamic argumentation framework.

As mentioned before, the grounded semantics [6] could return empty extensions. For instance, refer to Example 8 assuming an empty comparison criterion "$\succcurlyeq$". Thus, the usage of different semantics [3] could be studied to overcome this issue. Future work also involves a deep investigation on the applicability of the generalized `GenAF` wrt. more expressive DLs.

7 Conclusion

A novel argumentation framework was presented as a generalization of the classical Dung's `AF` named `GenAF`. A `GenAF` aims at providing a straightforward reification tool to reason about inconsistent knowledge bases specified through FOL fragments.

The proposal of the `GenAF` keeps the abstraction from the logic used to represent knowledge inside arguments while specifying a logic $\mathbb{A}$`rgs` to give some structure to arguments. This allows to generalize the abstract frameworks in order to leave them prepared to deal with any knowledge representation language within FOL. In this sense, the notion of coalitions was introduced to characterize sets of arguments supporting a premise. A coalition of structures was also formalized in order to recognize some set of structures that in conjunction introduce a conflict regarding another argument in the `GenAF`. Such notions allow to generalize the argumentation notions of attack and support.

Besides (ground) arguments from usual `GenAF`'s, schematic arguments are also presented. This is a necessary extension to provide an analogy to basic FOL elements like polyadic predicates, which may consider several parameters. Recall that structures in usual `GenAF`'s are equivalent to classical arguments in an abstract framework like that of Dung. In this proposal, given that schematic structures are introduced, the notion of structures is slightly modified by allowing them to exist despite they keep free premises. Therefore, only argumental structures are comparable to classical arguments, since both require to satisfy the set of premises in order to reach the claim.

Different classes of attack were also proposed, and in particular, the attack of schematic structures, which recognize a conflict in advance. That is, conflicts are identified wrt. the smallest possible structures, irrespective of any bigger argumental structure. This matter is of utmost importance in the debugging of incoherence terminologies.

The definition of `GenAF`'s allows to cope with specific logics for arguments. A reification of the logic $\mathcal{L}^{\kappa}$–used to specify in an abstract manner the logic $\mathbb{A}$`rgs` for arguments in the `GenAF`– to the $\mathcal{ALC}$ DL renders an interesting methodology to handle ontology change. As a preliminary result, we proposed a novel theoretical approach to cope with ontology debugging through argumentation, although it seems to be useful to other subareas of ontology change like ontology evolution and integration.

Acknowledgements

We would like to thank Guillermo R. Simari for discussions and comments on preliminary drafts of this article. This paper is partially financed by CONICET (PIP 5050), Universidad Nacional del Sur (PGI 24/ZN11) and Agencia Nacional de Promoción Científica y Tecnológica (PICT 2002 Nro 13096).

References

[1] Franz Baader. Logic-Based Knowledge Representation. In *Artif. Intell. Today*, pages 13–41. 1999.

[2] Franz Baader, Diego Calvanese, Deborah L. McGuinness, Daniele Nardi, and Peter F. Patel-Schneider, editors. *The Description Logic Handbook: Theory, Implementation, and Applications*. Cambridge University Press, 2003.

[3] Pietro Baroni and Massimiliano Giacomin. On Principle-Based Evaluation of Extension-Based Argumentation Semantics. *Artificial Intelligence*, 171(10-15):675–700, 2007.

[4] Philippe Besnard and Anthony Hunter. Practical First-Order Argumentation. In *AAAI*, pages 590–595, 2005.

[5] Alex Borgida. On the relative expressiveness of description logics and predicate logics. *Artif. Intell.*, 82(1-2):353–367, 1996.

[6] Phan Minh Dung. On the Acceptability of Arguments and its Fundamental Role in Nonmonotonic Reasoning and Logic Programming and n-person Games. *Artif. Intell.*, 77:321–357, 1995.

[7] Giorgos Flouris, Zhisheng Huang, Jeff Z. Pan, Dimitris Plexousakis, and Holger Wache. Inconsistencies, Negations and Changes in Ontologies. In *AAAI*, pages 1295–1300, 2006.

[8] Sven Ove Hansson. Kernel Contraction. *Journal of Symbolic Logic*, 59:845–859, 1994.

[9] Carsten Lutz, Ulrike Sattler, and Frank Wolter. Description Logics and the Two-Variable Fragment. In *Description Logics*, 2001.

[10] Diego C. Martínez, Alejandro Javier García, and Guillermo Ricardo Simari. Modelling Well-Structured Argumentation Lines. In *IJCAI*, pages 465–470, 2007.

[11] Thomas Meyer, Kevin Lee, and Richard Booth. Knowledge Integration for Description Logics. In *AAAI*, pages 645–650, 2005.

[12] Thomas Meyer, Kevin Lee, Richard Booth, and Jeff Z. Pan. Finding Maximally Satisfiable Terminologies for the Description Logic ALC. In *AAAI*, 2006.

[13] Martín Moguillansky, Marcelo Falappa, and Guillermo Simari. Model-Based Contractions for Description Logics. In *NMR*, pages 34–42, 2008.

[14] Martín Moguillansky, Nicolás Rotstein, and Marcelo Falappa. A Theoretical Model to Handle Ontology Debugging & Change Through Argumentation. In *The ISWC-08 International Workshop on Ontology Dynamics (IWOD)*, pages 29–42, 2008.

[15] Martín Moguillansky, Nicolás Rotstein, Marcelo Falappa, Alejandro García, and Guillermo Simari. Argument Theory Change Applied to Defeasible Logic Programming. In *AAAI*, pages 132–137, 2008.

[16] Martín Moguillansky, Nicolás Rotstein, Marcelo Falappa, Alejandro García, and Guillermo Simari. Argument Theory Change Through Defeater Activation. In *CMNA Workshop*, pages 24–33, 2009.

[17] Martín Moguillansky, Nicolás Rotstein, Marcelo Falappa, and Guillermo Simari. Generalized Abstract Argumentation: Handling Arguments in FOL Fragments. In *ECSQARU*, pages 144–155, 2009.

[18] Martín Moguillansky and Renata Wassermann. Inconsistent-Tolerant DL-Lite Reasoning: An Argumentative Approach. In *The IJCAI-09 Workshop on Automated Reasoning about Context and Ontology Evolution (ARCOE)*, pages 7–9, 2009.

[19] Michael Mortimer. On Languages with Two Variables. *Zeitschr. f. math. Logik und Grundlagen d. Math.*, 21:135–140, 1975.

[20] Guilin Qi, Peter Haase, Zhisheng Huang, and Jeff Z. Pan. A Kernel Revision Operator for Terminologies. In *DL*, 2008.

[21] Guilin Qi, Weiru Liu, and David A. Bell. Combining multiple prioritized knowledge bases by negotiation. *Fuzzy Sets and Systems*, 158(23):2535–2551, 2007.

[22] Márcio Moretto Ribeiro and Renata Wassermann. Base Revision in Description Logics - preliminary results. In *The ISWC-07 International Workshop on Ontology Dynamics (IWOD)*, 2007.

[23] Nicolás Rotstein, Martín Moguillansky, Marcelo Falappa, Alejandro García, and Guillermo Simari. Argument Theory Change: Revision Upon Warrant. In *COMMA*, pages 336–347, 2008.

[24] Nicolás Rotstein, Martín Moguillansky, Alejandro García, and Guillermo Simari. An Abstract Argumentation Framework for Handling Dynamics. In *NMR*, pages 131–139, 2008.

[25] Stefan Schlobach. Debugging and Semantic Clarification by Pinpointing. In *ESWC*, pages 226–240, 2005.

[26] Stefan Schlobach and Ronald Cornet. Non-Standard Reasoning Services for the Debugging of Description Logic Terminologies. In *IJCAI*, pages 355–362, 2003.

[27] Gerard Vreeswijk. Abstract Argumentation Systems. *Artif. Intell.*, 90(1-2):225–279, 1997.