Expert Systems With Applications 64 (2016) 141–168

Contents lists available at ScienceDirect

Expert Systems With Applications

journal homepage: www.elsevier.com/locate/eswa

A generalized abstract argumentation framework for inconsistency-tolerant ontology reasoning

Martín O. Moguillansky*, Guillermo R. Simari

Institute for Computer Science and Engineering (ICIC) National Council of Scientific and Technical Research (CONICET) AI R & D Lab (LIDIA), Department of Computer Science and Engineering (DCIC) Universidad Nacional del Sur (UNS), ARGENTINA

A R T I C L E I N F O

Article history:
Received 13 April 2016
Revised 18 July 2016
Accepted 19 July 2016
Available online 21 July 2016

Keywords:
Argumentation
Inconsistency-tolerant knowledge base reasoning
Argumentation rationality
Description logics

A B S T R A C T

We propose a new family of abstract argumentation frameworks which we refer to as generalized (identified through the acronym `GenAF`), due to its ability of adapting to different representation languages. `GenAF`s are formalized upon an unspecified representation language $\mathcal{L}$ but enriched with some inner structure which allows relating formulæ from the underlying knowledge base with primitive elements of arguments. The well-known Dung's standard semantics are adapted to construct the `GenAF`'s reasoning machinery. As an application, we reify the `GenAF`'s abstract language for arguments with the basic $\mathcal{ALC}$ flexibility of the presented formalism and a way of applying argumentation for reasoning over inconsistent ontologies. Finally, a detailed study is performed on the matter of argumentation rationality. The `GenAF` when concretized into a generalized argumentation system (`GenAS`) is studied under specific conditions which turn it into a standard logic-based argumentation system. This brings the opportunity to verify the scope of widely accepted postulates for logic-based argumentation to control the well behavior of the `GenAS` and to relate it to other argumentation systems.

1. Introduction

The *Generalized argumentation framework* (`GenAF`)[1] is another alternative for bridging the gap between argumentation frameworks (`AF`s) of extreme abstraction like Dung's (Dung, 1995) and logic-based argumentation systems which are constructed upon an underlying knowledge base (KB). However, the `GenAF` is an argumentation framework of high versatility obtained from the generalization of the representational logic $\mathcal{L}$ upon which it is constructed. Thus, for an abstract `GenAF`, $\mathcal{L}$ will remain unspecified. This decision comes from a main motivation: implement a computer system dedicated to reason about inconsistent knowledge bases (KBs), where both its logic $\mathcal{L}$ and the underlying KB could be selected at will by a simple redeployment. Thus, the deployment of the inconsistency-tolerant `GenAF`-reasoner for a specific logic $\mathcal{L}$ would require three different modules: 1) the abstract `GenAF` module, 2) an $\mathcal{L}$ instantiation module, and 3) a (consistency-based) $\mathcal{L}$-reasoner. With a little more detail, the first module specifies the interaction with the second module for instantiating $\mathcal{L}$ and, with the third module for constructing arguments and attacks. The second module specifies the concretization of certain `GenAF`'s fundamental elements for manipulating $\mathcal{L}$ formulæ. This is referred as *Generalized argumentation theory* (`GenAT`). Finally, the third module refers to any yet implemented reasoner for consistent KBs expressed through an specific logic $\mathcal{L}$. Afterwards, the deployed $\mathcal{L}$-`GenAF` reasoner could be consulted about the acceptability of a $\mathcal{L}$-query wrt. a given KB $\Sigma \subseteq \mathcal{L}$. This would be done through the interaction with a run time constructed *Generalized argumentation system* (`GenAS`) which would provide argumentation semantics for reasoning upon conflicting arguments constructed from Σ.

Keeping a practical approach in mind requires caring about how construction of arguments would take place in practice. While purely theoretical approaches do not even consider such matters, rule-based approaches like ABA (Bondarenko, Dung, Kowalski, & Toni, 1997; Bondarenko, Toni, & Kowalski, 1993), ASPIC+ (Prakken, 2010), or *DELP* (García & Simari, 2004) are defined only upon a specific representation language. In consequence, the challenge of this article is twofold: to bring versatility on the logic $\mathcal{L}$ (*external versatility*) and to bring versatility for structuring knowledge upon the logic $\mathcal{L}$ (*internal versatility*). External versatility is required for providing reasoning services for several logics. On the other hand, internal versatility is motivated by the pursue of efficient reasoning services. Reducing the computational cost of

* Corresponding author.
E-mail addresses: mom@cs.uns.edu.ar (M.O. Moguillansky), grs@cs.uns.edu.ar (G.R. Simari).

[1] Preliminary approaches to a `GenAF` were given in Moguillansky, Rotstein, and Falappa (2010a); Moguillansky, Rotstein, Falappa, and Simari (2009); (Rotstein, Moguillansky, García, & Simari, 2010).

http://dx.doi.org/10.1016/j.eswa.2016.07.027
0957-4174/

reasoning usually requires manipulating formulæ of the KB. This is usually done through the use of techniques of knowledge compilation (like Darwiche & Marquis, 2002, among others) which turns the knowledge base into a normal form. With the same objective, many description logics bring alternatives for compiling its formulæ. For instance, $\mathcal{ALC}$ ontologies can be unfolded (Schlobach, Huang, Cornet, & van Harmelen, 2007) or turned into the Bienvenu's $\mathcal{ALC}$ prime implicate normal form (Bienvenu, 2008), and also $\mathcal{EL}$ ontologies can be normalized according to Baader, Brandt, and Lutz (2005) and Brandt (2004) with polynomial results.

With this regards, some description logics (DLs) for ontology reasoning, present different restrictions on their description languages with the objective of improving the computational complexity of reasoning about the ontology they describe. Hence, the computational cost is reduced in detriment of expressivity. It is for that reason that negation of axioms in DLs is a well-known problem and in general is not possible, which makes the definition of a generalized argumentation machinery more particular; *e.g.*, the DL-Lite$_{\mathcal{A}}$ (Calvanese, De Giacomo, Lembo, Lenzerini, & Rosati, 2007) description language does not accept conjunctions. For a concept inclusion axiom like $A \sqsubseteq B$ [2], its negation $\neg(A \sqsubseteq B)$ –in accordance to Flouris, Huang, Pan, Plexousakis, and Wache (2006)– would have two possibilities: *consistency-negation*, which ends up with an assertion like $\exists(A \sqcap \neg B)(x)$ (*i.e.*, existence of some individual corresponding both to A and B), and *coherence-negation*, which leads to an axiom like $A \sqsubseteq \neg B$. Clearly, the former case falls out of the scope of the language; however, having two assertions like $A(a)$ and $\neg B(a)$, would be equivalent to the consistency-negation of the original axiom. This inconvenience –which is just one particularity of one of the various existing DLs– will affect how a generalized argumentation framework defines its notion of counterargument which usually relies upon some form of negation of formulæ for identifying conflicts. For this purpose, DL reasoners usually bring (un)satisfiability checking reasoning services. A GenAF can take advantage of such situation by fundamenting its machinery upon satisfiability checking, for instance, for configuring conflicts. This would allow the usage of such reasoning service through the interaction with a DL reasoner (third module) for the construction of the GenAS.

Internal versatility will be achieved through an *argument language framework* (*AL-framework*), which structures arguments in accordance to the normalization method for compiling formulæ. An AL-framework defines a tuple containing $\mathcal{L}$ as a general unspecified fragment of FOL along with the sublanguages for representing claims (or right-hand sides of formulæ), premises (or left-hand sides of formulæ), and assertional knowledge, according to the DL terminology (or ground formulæ). The AL-framework is rationalized through *legality conditions* for ensuring an appropriate interrelation between its sublanguages. This allows the formalization of an *argument (representation) language* for providing some structure to the still abstract notion of argument. As a consequence, this generalization renders a less abstract form of argumentation which serves to develop the necessary theoretical elements to give form to the structures in which arguments may aggregate to support a query, or even to attack other structures. A nice property of the GenAF is that it shares the same primitive elements from the normalized KB: an *argumental atom* (or just, *atom*) stands for a single formula from the underlying KB. A resulting advantage is that a GenAF may be straightforwardly adapted to deal with dynamics of knowledge as done in Moguillansky, Wassermann, and Falappa (2012): removing an argument from the framework would mean deleting a single minimal piece of it, *i.e.*, an atom, which implies the deletion of a formula from the KB. Additionally, specifying the inner components of arguments allows to establish the general form of the minimal portion of a formula which could be removed from a KB.

The GenAT appears as the result of the concretization of the fundamental elements of the GenAF including the AL-framework, an underlying KB, a *pre-argumental normalization function* for turning the KB into a repository of argumental atoms, and others. This allows the construction of arguments and attacks for building the GenAS. Thus, a GenAS can be seen as a non-standard logic-based argumentation system due to its additional qualities and versatilities. However, a GenAS under certain conditions behaves as a standard logic-based system. This allows the possibility to rely upon well known advances in the area of rationality for logic-based argumentation systems. Argumentation postulates proposed in Amgoud (2014) allows this theoretic study for clarifying the different alternatives in the behavior of a GenAS. As a consequence of this analysis, an alternative notion of admissibility for logic-based argumentation is given as an essential for the formalization of logic-based argumentation semantics based upon Dung's standard semantics. The conditions under which a GenAS guarantees the five argumentation postulates proposed are established through corresponding theorems.

In summary, Section 2 briefly introduces the different variations of abstract argumentation and how we start from them to obtain the specification of a generalized argumentation framework. Section 3 presents the fundamental elements for an abstract specification of the framework. Those elements, once concretized, conform the GenAT which allows the construction of the repository of atoms for the GenAF. Section 4 details the construction of arguments upon atoms and identify attack relations upon conflicting arguments. In Section 5, the GenAS is constructed upon a GenAF, a set of argumental structures related to the GenAF, and an attack relation between pairs of argumental structures. Argumentation semantics applies over GenASs as a standard way of reasoning upon conflicting arguments. Section 6 studies the application of the GenAF to reason over inconsistent $\mathcal{ALC}$ ontologies through a concrete specialized GenAS. Section 7 proposes a reduction of a GenAS into a classical logic-based system for studying postulates and properties as proposed by Amgoud in Amgoud (2014). This allows to understand the behavior of GenASs under usual circumstances and to relate our proposal with others. Section 8 discusses related and future work. Finally, Section 9 presents a final discussion on the proposed theory. The proofs of the proposed properties are included in the Appendix.

2. Abstract frameworks and our way towards generalized frameworks

Abstract frameworks, like Dung's AFs (Dung, 1995), are defined by a pair $\langle \mathbf{A}, \mathbf{R} \rangle$, where $\mathbf{A}$ is a set of abstract arguments $\mathcal{A}$, and $\mathbf{R}$ is the set of pairs $(\mathcal{A}_1, \mathcal{A}_2)$ denoting relations $\mathcal{A}_1 \mathbf{R} \mathcal{A}_2$ to indicate $\mathcal{A}_1 \in \mathbf{A}$ *defeats* $\mathcal{A}_2 \in \mathbf{A}$. Dung refers to the defeat relation through the term *attack*. Nothing is said about what arguments are, nor about its inner structure. The defeat relation is used for describing conflicting arguments.

Bipolarity (Amgoud, Cayrol, Lagasquie-Schiex, & Livet, 2008) introduced the notion of support as a new sort of relation among arguments. This brought to light the idea that arguments are naturally interrelated by two diametrically opposed forces: a positive one or *support*, and a negative one or *defeat*. Informally, an argument $\mathcal{A}_1$ supports a second argument $\mathcal{A}_2$ if $\mathcal{A}_1$ provides some additional justification for a premise in $\mathcal{A}_2$. A bipolar framework is defined as a triple $\langle \mathbf{A}, \mathbf{R}_d, \mathbf{R}_s \rangle$, where $\mathbf{R}_d$ ($\mathbf{R}_s$) is the defeat (support) relation, such that $\mathcal{A}_1 \mathbf{R}_d \mathcal{A}_2$ ($\mathcal{A}_1 \mathbf{R}_s \mathcal{A}_2$) implies that argument $\mathcal{A}_1 \in \mathbf{A}$ defeats (supports) argument $\mathcal{A}_2 \in \mathbf{A}$.

[2] The DL-axiom $A \sqsubseteq B$ can be seen as a FOL formula $p(x) \rightarrow q(x)$, where p and q stand for concepts A and B, respectively.

Subargumentation (Martínez, García, & Simari, 2007) was introduced with the objective of identifying the minimal portion of an argument that is defeated by another argument through the relation **R**, and thereafter avoid the reintroduction of the defeated portion in the reasoning process. These frameworks usually incorporate a set $\trianglelefteq$ containing relations $\mathcal{A}_2 \trianglelefteq \mathcal{A}_1$, indicating $\mathcal{A}_2 \in \mathbf{A}$ is a subargument of $\mathcal{A}_1 \in \mathbf{A}$, where $\langle \mathbf{A}, \mathbf{R}, \trianglelefteq \rangle$ is the new frameworks' definition. In this manner, an abstract argument can be broken into pieces, namely its subarguments which are abstract arguments themselves. This argument-unfolding process may continue until we obtain a subargument that is atomic, that is, an argument whose only subargument is itself.

When considering subargumentation along with a support relation (see Moguillansky et al., 2009; Rotstein et al., 2010), it is possible to think an argument either as a "black box of knowledge" or as a tree of inner subarguments whose links are a support relation between a child and its parent. Hence, the support relation turns to be exclusive: an argument $\mathcal{A}$ can only be considered when each of its premises is supported by other arguments. This allows to consider certain dependency among arguments in the sense that some arguments may depend on the existence of others to take place. These sort of frameworks are defined by a quartet $\langle \mathbf{A}, \mathbf{R}_d, \mathbf{R}_s, \trianglelefteq \rangle$, and their purpose is to keep certain level of abstraction while getting closer to a concrete argumentation system allowing the study of properties corresponding to a specific domain, *e.g.*, reasoning over inconsistency in knowledge bases.

Our proposal of generalized frameworks has the objective of introducing a new form of abstract AFs which is capable of being adapted to different representation languages for handling a wide range of inconsistent knowledge bases represented in these languages; for that reason, a generalized framework gets closer to a concrete argumentation system while still keeping a certain level of abstraction. To this end, we will consider an abstract representation language $\mathcal{L}$ which, when instantiated will determine every relation in the system; that is, a generalized argumentation framework, or GenAF, will be defined by a set **A** of argumental atoms and a logical system considering the language $\mathcal{L}$ and its entailment "$\vDash$" from which it will be possible to construct the support ($\mathbf{R}_s$), defeat ($\mathbf{R}_d$), and subargument ($\trianglelefteq$) relations altogether. Thus, a GenAF will be given through a pair $\langle \pi, \mathbf{A} \rangle$, where π stands for an *argument language framework* containing specifications for the obtention of the aforementioned relations and for constructing $\mathcal{L}$-arguments by relying upon atoms from **A**. One of the most remarkable advantages of GenAFs is their application to description logics for reasoning over inconsistent ontologies. For that reason, in Section 6 we have presented an example in that important area of real world applications.

3. Fundamentals for logic-Generalized AFs

This section is devoted to the introduction of those basic elements upon which our logic-generalized frameworks will be formalized. In Section 3.1 we introduce the elementary conceptual backgrounds on logics and the use of interpretation models through the well known set theoretic Tarskian semantics. Afterwards, in Section 3.2, we introduce the GenAF basic elements by abstracting away from any possible repository of knowledge. Finally, in Section 3.3, we introduce the construction of GenAFs from underlying knowledge bases. We discuss an appropriate construction of GenAFs through knowledge base normalization methods.

3.1. Backgrounds

The maximum expressive power of a GenAF is imposed by restricting its inner components to correspond to some logic $\mathcal{L}^\kappa$, with $\kappa \in \mathbb{N}_0$ (where $\mathbb{N}_0$ is the set of natural numbers including zero). Formulæ in $\mathcal{L}^\kappa$ are those of FOL that can be built with the help of predicate symbols of arity $\leq \kappa$, including equality and constant symbols, but without function symbols. In particular, $\mathcal{L}^2$ has been shown to be decidable in Mortimer (1975); an example of an $\mathcal{L}^2$-compliant logic is the $\mathcal{ALC}$ DL used to describe basic ontologies (Baader, 1999; Borgida, 1996).

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 (or higher order) predicate letters, *x, y* for free variable objects, and *a, b, c, d* for constants (individual names). We use Greek lowercase letters like ϑ to refer to formulæ like $p_1(x) \wedge p_2(x) \rightarrow q_1(x)$ and Greek capital letters like Γ to refer to sets containing formulæ like ϑ. We may also make explicit the usage of variables (or constants) by writing $\vartheta(x)$ (or $\vartheta(a)$) to refer to any formula which considers a free variable x (a constant a). To identify formulæ using predicates of arity > 1 we will write $\vartheta(\bar{x})$, where $\bar{x} = \langle x_1, \ldots, x_n \rangle$ is a vector assuming each x_i in $\bar{x}$ is a free variable; $\vartheta(\bar{a})$, where $\bar{a} = \langle a_1, \ldots, a_n \rangle$ is a vector assuming each a_i in $\bar{a}$ is a constant; and $\vartheta(\bar{x})(\bar{a})$, for a formula considering a combination of free variables x_i in $\bar{x} = \langle x_1, \ldots, x_n \rangle$ and constants a_j in $\bar{a} = \langle a_1, \ldots, a_m \rangle$. For instance, $p_1(x_1) \wedge p_2(a) \wedge r_1(x_1, a, x_2) \rightarrow p_3(a)$ can be abstracted away by writing $\vartheta(\bar{x})(\bar{a})$, where $\bar{x} = \langle x_1, x_2 \rangle$ and $\bar{a} = \langle a \rangle$, or also for making the abstraction a little more specific, by writing $\vartheta_1(\bar{x})(\bar{a}) \rightarrow \vartheta_2(\bar{a})$. For simplicity, we will omit universal quantifiers writing $(\exists \bar{y})\vartheta(\bar{x})(\bar{y})$ to refer to $(\forall x_1, \ldots, x_n)(\exists y_1, \ldots, y_m)(\vartheta(x_1, \ldots, x_n, y_1, \ldots, y_m))$.

A *substitution* $\nu = \bar{x}/\bar{a} = \{x_1/a_1, \ldots, x_n/a_n\}$ is referred for mapping free variables x_i in $\bar{x} = \langle x_1, \ldots, x_n \rangle$ to constants a_i in $\bar{a} = \langle a_1, \ldots, a_n \rangle$. Thus, $\vartheta(\bar{x})[\nu] = \vartheta(\bar{a})$ holds whenever vector $\bar{x}$ is substituted through $\nu = \bar{x}/\bar{a}$ by vector $\bar{a}$. For instance, $(p_1(x_1) \wedge p_2(x_2) \rightarrow r(x_1, y_2))[\nu] = p_1(a_1) \wedge p_2(a_2) \rightarrow r(a_1, a_2)$ given the substitution $\nu = \{x_1/a_1, x_2/a_2\}$. Substitutions may also be applied over sets Γ of formulæ. For instance, $\Gamma(\bar{x})[\nu] = \Gamma(\bar{a})$ holds through a substitution $\nu = \bar{x}/\bar{a}$. Moreover, whenever an explicit reference to variables and/or constants is not necessary, we may write $\Gamma[\nu]$ for identifying a set whose formulæ are those in Γ substituted through ν.

We interpret $\mathcal{L}^\kappa$ in terms of the standard set theoretic Tarskian semantics, 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. We say that the interpretation $\mathcal{I}$ is a model of a formula $\vartheta \subseteq \mathcal{L}^\kappa$, by writing $\mathcal{I} \models \vartheta$, if ϑ is true according to $\mathcal{I}$, or equivalently when $\vartheta^\mathcal{I} \neq \emptyset$ holds. By writing $\vartheta_1 \vDash \vartheta_2$ we mean that ϑ_2 holds in all the cases that ϑ_1 holds, or equivalently, $\vartheta_1^\mathcal{I} \subseteq \vartheta_2^\mathcal{I}$ holds. The same intuition is extended for sets of formulæ, for instance, by writing $\Sigma \models \vartheta$, for any $\Sigma \subseteq \mathcal{L}^\kappa$, we mean that $\Sigma^\mathcal{I} \subseteq \vartheta^\mathcal{I}$, or equivalently that ϑ holds in the context of Σ, or what it the same, that ϑ is true for every model $\mathcal{I}$ of Σ. When an interpretation $\mathcal{I}$ is a model of the knowledge base $\Sigma \subseteq \mathcal{L}^\kappa$, we write $\mathcal{I} \models \Sigma$, implying $\mathcal{I} \models \vartheta$, for every formula $\vartheta \in \Sigma$. A knowledge base Σ *logically implies* (or *entails*) a formula ϑ, written $\Sigma \vDash \vartheta$, if for every model $\mathcal{I}$ of *KB*, $\mathcal{I} \models \vartheta$. As usual, a KB is said *satisfiable* or *consistent*, if it admits at least one model. For an interpretation $\mathcal{I}$, some $\{a, b\} \subseteq \Delta^\mathcal{I}$, and a formula $\theta(x, y)$, we write $\mathcal{I} \models \theta(a, b)$ if $\mathcal{I}, \nu \models \theta(x, y)$, for the substitution $\nu = \{x/a, y/b\}$ mapping x to a and y to b. We also write $\vartheta'(x, y) \models_{(\mathcal{I},\nu)} \vartheta(x, y)$ whenever there is an interpretation $\mathcal{I}$ and a substitution ν satisfying $\vartheta'(x, y) \models_\mathcal{I} \vartheta(a, b)$. Moreover, given $\Gamma \subseteq \mathcal{L}^\kappa$, we write $\Gamma \models_{(\mathcal{I},\nu)} \vartheta(x, y)$ whenever we need to make explicit the interpretation $\mathcal{I}$ and substitution ν through which $\Gamma \models_\mathcal{I} \vartheta(a, b)$ is satisfied.

Sometimes, it may be the case that a knowledge base Σ contains a formula ϑ of the form $\rho_1 \wedge \ldots \wedge \rho_n \rightarrow \alpha$ which is satisfiable given that there is a model $\mathcal{I}$ of *KB* uch that $\rho_1^\mathcal{I} \cap \ldots \cap \rho_n^\mathcal{I} \subseteq \alpha^\mathcal{I}$ holds, but then it may happen that $\rho_1^\mathcal{I} \cap \ldots \cap \rho_n^\mathcal{I} = \emptyset$ holds for every such model $\mathcal{I}$. This is the case of an unsatisfiable left hand-

Table 1
Mathematical domain sets used throughout the article.

Symbol	Domain name/Usage	Definition
$\mathbb{N}_0$	Natural numbers (zero included)	Page 5
$\mathcal{L}$	FOL fragment	3.1
$\mathcal{L}^\kappa$	General Logic $\mathcal{L}$ with arity $\leq \kappa$	3.1
$\mathcal{L}_{\mathtt{cl}}$	Language for claims	3.1
$\mathcal{L}_{\mathtt{pr}}$	Language for premises	3.1
$\mathcal{L}_{\mathtt{a}}$	Assertional language	3.1
$\mathbb{L}$	Argument language (AL) frameworks	3.1
$\mathbb{A}_\pi$	Atoms from an AL-framework $\pi \in \mathbb{L}$	3.2
$\mathbb{F}$	Generalized argumentation frameworks	3.7
$\mathbb{T}$	Generalized argumentation theories	3.14
$\mathbb{G}$	Generalized argumentation systems	5.1
$\mathcal{ALC}$	Standard description logic	6.1
$\mathcal{L}_{\mathrm{T}}$	π-pANF $\mathcal{ALC}$ axiom	6.1
$\mathbb{B}$	Logic-based arguments	7.1
$\mathbb{S}$	Logic-based argumentation systems	7.2

side of a formula wrt. Σ. Athough satisfiability/consistency does still hold in Σ thi condition renders a "kind of inconsistency" at non-assertional level. That is, the inconsistency appears only when the necessary assertions (ground formulæ) are incorporated to the knowledge base. Such a situation provokes the support of the left-hand side $\rho_1, \ldots, \rho_n$ and afterwards, the right-hand side α is also inferred, triggering the inconsistency. This is a usual discussion in ontology engineering, referred as *incoherence* (Flouris et al., 2006), which we will attend opportunely.

We will also use some additional notational conventions: Blackboard bold (*aka* Mathematical Double-struck) capital letters like $\mathbb{L}, \mathbb{A}, \mathbb{F}, \mathbb{T}$, and others, are referred to identify mathematical domain sets; Gothic (*aka* Mathematical Fraktur) lowercase letters (or words) like $\mathfrak{pr}$, $\mathfrak{cl}$, and $\mathfrak{genaf}$, for identifying functions; the Mathematical Calligraphic capital letters[3] for identifying elements of the GenAF, for instance $\mathcal{A}$ identifies argumental atoms, $\mathcal{C}$ for coalitions of argumental atoms, $\mathcal{S}$ for argumental structures composed by argumental atoms, and $\mathcal{L}$ (generally with subscripts) for identifying (representation) languages, were $\mathcal{L}_{\mathtt{pr}}$ refers to the language for premises, $\mathcal{L}_{\mathtt{cl}}$ for claims, and $\mathcal{L}_{\mathtt{a}}$ for assertions; some special Greek lowercase letters identify concrete instances like, π for the argument language framework, δ for a specific GenAF, τ for a GenAT, or σ for a GenAS; bold capital letters will identify sets containing elements expressed through Mathematical Calligraphic capital letters, for instance, $\mathbf{A}$ is a set of atoms $\mathcal{A}$, $\mathbf{S}_\delta$ is a set of argumental structures $\mathcal{S}$, $\mathbf{R}$ and $\mathbf{C}$ identify attack relations between pairs $(\mathcal{S}_1, \mathcal{S}_2)$ of conflicting structures $\mathcal{S}_1$ and $\mathcal{S}_2$. Thus, for instance, we may write: *given a* GenAF $\delta \in \mathbb{F}$ *st.* $\delta = \langle \pi, \mathbf{A} \rangle$, *where* $\pi \in \mathbb{L}$ *is the legal AL-framework constructed from the domain* $\mathbb{L}$, *and* $\mathbf{A} \subseteq \mathbb{A}_\pi$ *is the set of atoms* $\mathcal{A}$ *from the domain* $\mathbb{A}_\pi$ *st. for each atom* $\mathcal{A} \in \mathbf{A}$ *it follows* ...Tables 1-5 show in detail the notations used throughout the article.

3.2. Elementary GenAF elements: argument language, argumental atoms and generalized framework

In abstract argumentation *à la* Dung, an argument is considered an indivisible piece of knowledge. When some structure is incorporated to the notion of argument, such indivisible condition is given up. In the case of GenAFs, an argument will be seen as a piece of knowledge whose claim is reachable from premises that are supported via *argumental atoms*. This means that GenAF arguments can be informally seen as minimal sets of argumental atoms supporting a claim.

Although this is a not so abstract vision of arguments as Dung's, it keeps certain level of abstraction given that it allows to prescind from specific logic constructions in order to study the GenAF framework and the interrelation of its inner parts. By establishing an analogy to argumentation systems where arguments are minimal sets of formulæ inferring a claim, we can think of atoms as a way for representing formulæ, where a formula's consequent stands for the atom's claim and the antecedent for a conjunction of premises. That is, for an atom, the conjunction of its premises implies the claim. However the correct logical meaning of an atom will require to keep unbreakable the interrelation between its claim and set of premises. Hence, in GenAFs, the indivisible condition of Dung's abstract arguments is shifted to atoms, which in turn will serve as building blocks for GenAF's arguments.

The purpose of dismembering atoms in claims and premises, as their fundamental entities, is to give a formal instrumentation for manipulating knowledge in which premises may have syntactical differences wrt. claims. From an etymological viewpoint, a premise is one of a set of requirements for drawing a conclusion, or claim. Premises can not be necessarily thought (or represented) in the same manner that a claim is done. A variety of situations can be mentioned in which the decision of dismembering premises and claims from formulæ results benefiting. The justification has to do, in general, with the need for conforming certain properties towards efficiency advances in specialized reasoning services. For instance, a horn clause is a disjunction of literals with at most one positive (unnegated) literal, like $p \vee \neg p_1 \vee \neg p_2 \vee \ldots \vee \neg p_n$. In logic programming this is essential for writing clauses in the form of material implications as $p \leftarrow p_1 \wedge p_2 \wedge \ldots \wedge p_n$, which is the basis for the SLD-resolution inference rule used to implement logic programming in Prolog. In fact, a goal-reduction procedure is done from this kind of syntax. For instance, the Horn clause written behaves as the procedure: *to show p, firstly show* p_1 *and show* p_2 *and ... and show* p_n.

In description logics (DLs) this kind of reduction is quite common. In some cases, the description languages have specific restrictions for constructing DL rules (known as general concept inclusion axioms, or just GCIs), by specifying different syntaxes for antecedents and consequents. For instance, DL-Lite$_{\mathcal{A}}$ (Calvanese et al., 2007) specifies A for atomic concepts, B for basic concepts with the form of either A or $\exists r$ (where r stands for a relation), and C for general concepts as either B or $\neg B$; and specifies DL-Lite$_{\mathcal{A}}$ axioms through the form $B \sqsubseteq C$, admitting the use of negative basic concepts only on the right hand side of the axiom. In other cases, explicit syntax differences between antecedent and consequent are made to obtain normal forms. There is a wide range of DLs for which specific normal forms are proposed in order to provide new algorithms for proof procedure. The objective in such cases is to achieve optimizations for reasoning services, lowering its computational complexity without loosing expressivity. For instance, the DL $\mathcal{EL}$ (Baader et al., 2005), has no restrictions for using a concept format on one side or the other of a GCI $C \sqsubseteq D$, where C and D are general (non-atomic) concepts $C ::= A | C \sqcap C | \exists r.C$. For example, its is possible to write $\mathtt{Pericardium} \sqsubseteq \mathtt{Tissue} \sqcap \exists \mathtt{containedIn.Heart}$ to specify that the pericardium is a tissue contained in the heart. In FOL this can be interpreted through a formula like $p_1(x) \rightarrow p_2(x) \wedge (\exists y)(r(x, y) \wedge p_3(y))$, where p_1 stands for Pericardium, p_2 for Tissue, p_3 for Heart, and the relation r, for containedIn. In order to decide subsumption[4] $C \sqsubseteq D$, a decision procedure would be to apply the $\mathcal{ALC}$-tableaux algorithm for deciding consistency of $\mathcal{ALC}$-concepts. Such procedure would take exponentially many steps in the worst case. However, a normal form could be used to refor-

[3] Do not confuse with series of capital letters like $\mathcal{ALC}$ and $\mathcal{EL}$, whose typography is usually used to identify DLs' families.

[4] Subsumption is a DLs' reasoning service which consist of analysing if there is an interpretation model $\mathcal{I}$ verifying $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$.

Table 2
Individuals and constants used throughout the article.

Symbol	Domain	Name/Usage	Definition
κ	$\mathbb{N}_0$	Predicate's arity	Page 5
π	$\mathbb{L}$	Argument language framework	3.1
$\mathcal{A}$	$\mathbb{A}_\pi$	Argumental atom	3.3
δ	$\mathbb{F}$	Generalized argumentation framework (GenAF)	3.7
ϵ	$\epsilon = \langle\emptyset, \perp\rangle$	Escape atom	3.12
τ	$\mathbb{T}$	Generalized argumentation theory (GenAT)	3.15
$\mathcal{C}$	$\wp(\mathcal{L}_{\mathtt{pr}}) \times \wp(\mathbb{A}_\pi) \times \mathcal{L}_{\mathtt{cl}}$	Claiming-coalition	4.2
$\mathcal{S}$	$\wp(\mathcal{L}_{\mathtt{pr}}) \times \wp(\mathbb{A}_\pi) \times \mathcal{L}_{\mathtt{cl}}$	Argumental structure	4.5
σ	$\mathbb{G}$	Generalized argumentation system (GenAS)	5.1
$\sigma^{\mathbf{C}}$	$\mathbb{G}$	Consistency based GenAS	5.11
$\mathfrak{s}$		Argumentation semantics	5.11
$\mathcal{B}$	$\mathbb{B}$	Logic-based argument	7.1

Table 3
Sets used throughout the article.

Symbol	Domain	Set name	Definition
Σ	$\mathcal{L}^\kappa$	Knowledge base	Page 6
$\mathbf{A}$	$\mathbb{A}_\pi$	Argumental atoms	3.7
Ω		Atom constraints	3.11
$\mathbf{S}_\delta$	$\wp(\mathcal{L}_{\mathtt{pr}}) \times \wp(\mathbb{A}_\pi) \times \mathcal{L}_{\mathtt{cl}}$	δ-structures	4.7
$\mathbf{E}$	$\wp(\wp(\mathbf{S}_\delta))$	Extension	5.10
$\mathcal{O}$	$\mathcal{L}^\kappa \times \mathcal{L}_{\mathtt{a}}$	Ontology	6.1
$\mathcal{O}$	$\mathcal{L}_{\mathtt{T}} \times \mathcal{L}_{\mathtt{a}}$	π-pANF $\mathcal{ALC}$ ontology	6.1
T	$\mathcal{L}_{\mathtt{T}}$	TBox	6.1
A	$\mathcal{L}_{\mathtt{a}}$	ABox	6.1
$\mathcal{E}$	$\wp(\wp(\mathbb{B}))$	Logic-based $\mathfrak{s}$-extension	7.3
$\mathbf{B}$	$\wp(\mathbb{B})$	(GenAF-Translated) logic-based arguments	7.7

Table 4
Relations used throughout the article.

Symbol	Domain	Relation name	Definition
$\trianglelefteq$	$\mathbf{S}_\delta \times \mathbf{S}_\delta$	Substructure	4.6
$\triangleleft$	$\mathbf{S}_\delta \times \mathbf{S}_\delta$	Strict substructure	4.6
$\succeq$	$\mathbf{A} \times \mathbf{A}$	Atom comparison	Page 21
$\succeq$	$\mathbf{S}_\delta \times \mathbf{S}_\delta$	Structure comparison	Page 21
$\mathbf{R}_\delta$	$\mathbf{S}_\delta \times \mathbf{S}_\delta$	Attack	4.9
$\mathbf{C}_\delta$	$\mathbf{S}_\delta \times \mathbf{S}_\delta$	Consistency attack	4.10
$\mathbf{C}_\delta^b$	$\mathbf{S}_\delta \times \mathbf{S}_\delta$	Base consistency attack	4.10
$\approx\!\!\mid_{[\tau,\mathfrak{s}]}$	$\wp(\mathcal{L}^\kappa) \times \mathcal{L}^\kappa$	Warranted	5.12
$\approx\!\!\mid^{\mathbf{C}}_{[\tau,\mathfrak{s}]}$	$\wp(\mathcal{L}^\kappa) \times \mathcal{L}^\kappa$	Consistently warranted	5.12
$\mathbf{R}$	$\mathbb{B} \times \mathbb{B}$	(Maximal) Logic-based attack	7.2
$\mathbf{T}$	$\mathbb{B} \times \mathbb{B}$	(GenAF-Translated) Logic-based attack	7.7

mat $\mathcal{EL}$-axioms restricting them to basic concept inclusions like $A_1 \sqcap \ldots \sqcap A_n \sqsubseteq B$, $A \sqsubseteq \exists r.B$ and $\exists r.A \sqsubseteq B$, where A and B are atomic concepts and $n \geq 1$. Such normal form can be computed in polynomial time and does not increase the size of the TBox more than polynomially (Brandt, 2004). For the example given above, the normalization would render two concept inclusions $\mathtt{Pericardium} \sqsubseteq \mathtt{Tissue}$ and $\mathtt{Pericardium} \sqsubseteq \exists \mathtt{containedIn.Heart}$. Moreover, there is an extensive amount of bibliography on the consideration of irrestrictive GCIs for reasoning in DLs, including the quite suggestive title of Baader and Peñaloza's work "GCIs Make Reasoning [...] Undecidable" (Baader & Peñaloza, 2011).

An alternative for claims that should be modelled in argumental atoms is the knowledge that represents factual information, *i.e.*, ground formulæ (no variables objects). Inspired by its usage in Description Logics, we refer to this kind of formulæ as *assertions*. The assertional knowledge describes what is true in the modelled domain and usually defines situation-specific knowledge. Since assertions represent factual information that is considered unconditionally, it is natural to think that they will appear as claims in argumental atoms that will be free of premises.

In addition, splitting the general argument language in two independent languages for premises and claims allows to specify as small as possible argumental atoms which will be benefiting for recognizing the minimal portions of arguments that become defeated in a graph; and, this is important for both a) avoiding cyclic paths of reasoning (argumentation lines when using dialectical trees) without unnecessary loss of knowledge (García & Simari, 2004) and b) handling argumentation dynamics by applying change operations like revisions or contractions for which it is important to perform minimal deletions of argumental knowledge (Moguillansky et al., 2012).

At the moment we will concentrate on specifying the sublanguages that will be involved in the construction of an *argument language*. To such end, we propose an *argument language framework* which will identify the sublanguages for claims ($\mathcal{L}_{\mathtt{cl}}$), premises ($\mathcal{L}_{\mathtt{pr}}$), and assertions ($\mathcal{L}_{\mathtt{a}}$), within a given logic $\mathcal{L}^\kappa$.

Definition 3.1 (Argument language framework)**.** A tuple $\langle \mathcal{L}^\kappa, \mathcal{L}_{\mathtt{cl}}, \mathcal{L}_{\mathtt{pr}}, \mathcal{L}_{\mathtt{a}} \rangle$ is identified as an **argument language framework** (or **AL-framework**, for short), where $\mathcal{L}^\kappa$ is the language of an underlying general logic, $\mathcal{L}_{\mathtt{cl}} \subseteq \mathcal{L}^\kappa$ is a language for claims, $\mathcal{L}_{\mathtt{pr}} \subseteq \mathcal{L}^\kappa$ is a language for premises, and $\mathcal{L}_{\mathtt{a}} \subseteq \mathcal{L}_{\mathtt{cl}}$ is an assertional language (ground formulæ). The set $\mathbb{L}$ will identify the domain of all possible AL-frameworks.

An *argument (representation) language* will be defined by establishing the interrelation between the sublanguages $\mathcal{L}_{\mathtt{pr}}$ and $\mathcal{L}_{\mathtt{cl}}$ of $\mathcal{L}^\kappa$. The idea here is to formalize the intuition given before for the configuration of atoms: a set of premises for drawing a claim. An argument language built from an AL-framework will later identify the domain of GenAFs argumental atoms.

Definition 3.2 (Argument language)**.** Given an AL-framework $\pi \in \mathbb{L}$ (where $\pi = \langle \mathcal{L}^\kappa, \mathcal{L}_{\mathtt{cl}}, \mathcal{L}_{\mathtt{pr}}, \mathcal{L}_{\mathtt{a}} \rangle$), the **argument language** $\mathbb{A}_\pi$ will be referred for identifying the set $\wp(\mathcal{L}_{\mathtt{pr}}) \times \mathcal{L}_{\mathtt{cl}}$ obtained from π.

The following example shows in an intuitive manner, how simple formulæ expressed in two different logics can be represented

Table 5
Functions used throughout the article.

Function Id	Mathematical definition	Function name	Definition
$\mathfrak{panf}_\pi$	$\wp(\mathcal{L}^\kappa) \longrightarrow \wp(\mathcal{L}^\kappa)$	Pre-argumental normalization	3.9
$\mathfrak{cl}$	$\mathbb{A}_\pi \longrightarrow \mathcal{L}_{\mathtt{cl}}$	Claim	Page 14
$\mathfrak{pr}$	$\mathbb{A}_\pi \longrightarrow \wp(\mathcal{L}_{\mathtt{pr}})$	Premises	Page 14
ω_π	$\mathcal{L}^\kappa \longrightarrow \{\mathfrak{true}, \mathfrak{false}\}$	Atom constraint	3.10
$\mathfrak{atom}_{[\pi,\Omega]}$	$\mathcal{L}^\kappa \longrightarrow \mathbb{A}_\pi \cup \{\epsilon\}$	Atom translation	3.12
$\mathfrak{genaf}$	$\mathbb{T} \longrightarrow \mathbb{F}$	Theory function	3.15
$\overline{}$	$\mathcal{L}^\kappa \longrightarrow \mathcal{L}^\kappa$	Contrapositive formula	Page 16
$\mathfrak{clset}$	$\wp(\mathbb{A}_\pi) \longrightarrow \wp(\mathcal{L}_{\mathtt{cl}})$	Set of claims	4.1
$\mathfrak{prset}$	$\wp(\mathbb{A}_\pi) \longrightarrow \wp(\mathcal{L}_{\mathtt{pr}})$	Set of premises	4.1
$\mathfrak{pr}$	$\wp(\mathcal{L}_{\mathtt{pr}}) \times \wp(\mathbb{A}_\pi) \times \mathcal{L}_{\mathtt{cl}} \longrightarrow \wp(\mathcal{L}_{\mathtt{pr}})$	Coalition's premises	4.2
$\mathfrak{bd}$	$\wp(\mathcal{L}_{\mathtt{pr}}) \times \wp(\mathbb{A}_\pi) \times \mathcal{L}_{\mathtt{cl}} \longrightarrow \wp(\mathbb{A}_\pi)$	Coalition's body	4.2
$\mathfrak{cl}$	$\wp(\mathcal{L}_{\mathtt{pr}}) \times \wp(\mathbb{A}_\pi) \times \mathcal{L}_{\mathtt{cl}} \longrightarrow \mathcal{L}_{\mathtt{cl}}$	Coalition's claim	4.2
$\mathfrak{pr}$	$\mathbf{S}_\delta \longrightarrow \wp(\mathcal{L}_{\mathtt{pr}})$	Structure's premises	4.5
$\mathfrak{bd}$	$\mathbf{S}_\delta \longrightarrow \wp(\mathbb{A}_\pi)$	Structure's body	4.5
$\mathfrak{cl}$	$\mathbf{S}_\delta \longrightarrow \mathcal{L}_{\mathtt{cl}}$	Structure's claim	4.5
$\mathfrak{genas}$	$\mathbb{T} \longrightarrow \mathbb{G}$	`GenAS` constructor	5.2
$\mathfrak{base}$	$\wp(\mathbf{S}_\delta) \longrightarrow \wp(\mathcal{L}^\kappa)$	Base function	5.3
$\mathcal{F}$	$\wp(\mathbf{S}_\delta) \longrightarrow \wp(\mathbf{S}_\delta)$	Characteristic function	5.10
$\mathfrak{ext}_{\mathfrak{s}}$	$\mathbb{G} \longrightarrow \wp(\wp(\mathbf{S}_\delta))$	Set of $\mathfrak{s}$-extensions	5.11
$\mathfrak{ext}^{\mathfrak{C}}_{\mathfrak{s}}$	$\mathbb{G} \longrightarrow \wp(\wp(\mathbf{S}_\delta))$	Set of consistency $\mathfrak{s}$-extensions	5.11
$\mathfrak{warrant}_{[\tau,\mathfrak{s}]}$	$\wp(\mathcal{L}^\kappa) \longrightarrow \wp(\wp(\mathcal{L}^\kappa))$	Warrant function	d
$\mathfrak{warrant}^{\mathfrak{C}}_{[\tau,\mathfrak{s}]}$	$\wp(\mathcal{L}^\kappa) \longrightarrow \wp(\wp(\mathcal{L}^\kappa))$	Warrant consistent function	5.15
$\mathfrak{plausible}_{[\tau,\mathfrak{s}]}$	$\wp(\mathcal{L}^\kappa) \longrightarrow \wp(\mathcal{L}^\kappa)$	Plausible function	5.16
$\mathfrak{plausible}^{\mathfrak{C}}_{[\tau,\mathfrak{s}]}$	$\wp(\mathcal{L}^\kappa) \longrightarrow \wp(\mathcal{L}^\kappa)$	Plausible consistent function	5.16
AS	$\wp(\mathcal{L}^\kappa) \longrightarrow \mathbb{S}$	Logic-based argumentation system	7.2
Args	$\wp(\mathcal{L}^\kappa) \longrightarrow \wp(\mathbb{B})$	Maximal set of logic-based arguments	7.2
$Ext_{\mathfrak{s}}$	$\mathbb{S} \longrightarrow \wp(\wp(\mathbb{B}))$	Set of logic-based $\mathfrak{s}$-extensions	7.3
$\mathfrak{lbarg}$	$\mathbf{S}_\delta \longrightarrow \mathbb{B}$	Logic-based argument function	7.4
$\mathfrak{lbas}$	$\mathbb{G} \longrightarrow \mathbb{S}$	Logic-based AS function	7.7

through argumental atoms. Note the correlation between the format of formulæ and the configuration of atoms: a conjunctive antecedent implies atom's premises. Observe also that argumental atoms (and later on, argumental structures) are graphically represented as triangles.

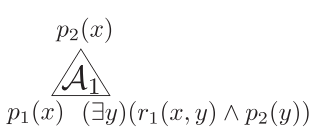
Example 1. "An alien who is son of an Argentinian is an Argentinian."

For its representation, we use unary predicates p_1 for "alien" and p_2 for "Argentinian", and a binary predicate r_1 for the relation "is son of".

In FOL this would be:

$p_1(x) \wedge (\exists y)(r_1(x, y) \wedge p_2(y)) \rightarrow p_2(x)$

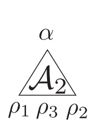
The argumental atom $\mathcal{A}_1$ that would arise is depicted on the right. Note that premises $p_1(x)$ and $(\exists y)(r_1(x, y) \wedge p_2(y))$ arise as the conjunctive terms conforming the antecedent of the formula while $p_2(x)$ stands for the atom's claim.



In logic programming (Horn logic), this would be:

`argentinian(X) :- alien(X), isSonOf(X,Y), argentinian(Y).`

For the argumental atom depicted on the right, due to space matters, premises were written as ρ_1, ρ_2, and ρ_3, and the claim as α.



$\alpha = \mathtt{argentinian(X)}$
$\rho_1 = \mathtt{alien(X)}$
$\rho_2 = \mathtt{isSonOf(X,Y)}$
$\rho_3 = \mathtt{argentinian(Y)}$

It is important to mention that the configuration of atoms will depend on the specification of the argument language –and thus, on the AL-framework. Notice that, in Example 1, for representing the same knowledge expressed in two different logics, the configuration of atoms may vary: atom $\mathcal{A}_1$ has only two premises while atom $\mathcal{A}_2$ has three. The reason is that the specification of the language $\mathcal{L}^\kappa$, and in particular the one for premises $\mathcal{L}_{\mathtt{pr}}$, determines the construction of terms (clauses) on the antecedent of a formula which ends up determining the premises of atoms. It is clear that there exists a syntactical dependency for the construction of argumental atoms on the argument language $\mathbb{A}_\pi$ throughout the AL-framework π.

Definition 3.3 (Argumental atom)**.** Given an AL-framework $\pi \in \mathbb{L}$, an **argumental atom** (or **atom**, for short) $\mathcal{A} \in \mathbb{A}_\pi$ is a pair $\langle \Gamma, \alpha \rangle$, where $\Gamma \subseteq \mathcal{L}_{\mathtt{pr}}$ is a finite (possibly empty) set of premises, $\alpha \in \mathcal{L}_{\mathtt{cl}}$ its claim, and it holds $\Gamma \cup \{\alpha\} \not\models \bot$ (consistency).

The consistency condition is natural to avoid constructing fallacious atoms. Note that we have formalized the notion of argumental atom by abstracting away from an epistemic source from which it may arise; however, it is necessary to recall that `GenAF`'s atoms will be ultimately constructed from an underlying KB. For instance, $\mathcal{A}_1 = \langle \{p\}, q \rangle$ is an atom according to Definition 3.3 that could appear from a formula like $(p \rightarrow q) \in \Sigma$, where Σ is the underlying KB. Note that unlike arguments, where the claim is inferred from an argument's body –or support set– atoms are in general not self conclusive[5], *i.e.*, for $\mathcal{A}_1$, $\{p\} \not\models q$ holds, and thus, it will be necessary to infer p in order to reach the claim q. Hence, we can say that an atom maintains the essential meaning of the formula it stands for.

Now, let us consider the case of atoms that are built with an empty set of premises. Usually, *evidence* (Walton, 2002), once admitted as such, is considered a basic irrefutable piece of knowledge. This means that evidence does not count with premises that need to be supported given that it is self-justified by definition.

[5] Self conclusive atoms appear from tautological implications like $p \vee q \rightarrow p$. In such a case, the atom would be $\langle \{p \vee q\}, p \rangle$, which can be seen as self conclusive given that the claim can be inferred straightforwardly from its premises, *i.e.*, $\{p \vee q\} \models p$.

We can mention two different options for specifying evidence: as a separate entity in the framework, as done in Rotstein et al. (2010); or as *evidential atoms*: atoms with no premises to be satisfied. In this article we assume the latter posture.

Definition 3.4 (Evidence)**.** Given an AL-framework $\pi \in \mathbb{L}$, an atom $\mathcal{A} \in \mathbb{A}_\pi$ is an **evidential atom** (or **evidence**) *iff* $\mathcal{A} = \langle \emptyset, \alpha \rangle$ and $\alpha \in \mathcal{L}_{\mathtt{a}}$ (assertional formula).

Evidential atoms have an empty set of premises, however, atoms with no premises can also be non-evidential if they have non-assertional formulæ as claims. For instance, an atom like $\langle \{\}, p(x) \rangle$. We refer to such non-evidential atoms as *primitive*.

Definition 3.5 (Primitive atom)**.** Given an AL-framework $\pi \in \mathbb{L}$, an atom $\mathcal{A} \in \mathbb{A}_\pi$ is a **primitive atom** *iff* $\mathcal{A} = \langle \emptyset, \alpha \rangle$ and $\alpha \notin \mathcal{L}_{\mathtt{a}}$ (non-assertional).

Premises can be thought as statements, or even assumptions, proposed for justifying a conclusion or claim. Therefore, it seems natural to expect more information, knowledge, or ultimately evidence, supporting each premise given. This action is performed through the interrelation with other argumental atoms which provide their claims for supporting an atom's premises. This is usually referred to as *support relation*. Whenever all premises are satisfied, the atom can finally draw its claim. The following example illustrates such relation in an intuitive manner.

Example 2. (Continues from Example 1). The argumental atoms illustrated before would be formally represented through the following structures:

$\mathcal{A}_1 = \langle \{p_1(x), (\exists y)(r_1(x, y) \wedge p_2(y))\}, p_2(x) \rangle$

$\mathcal{A}_2 = \langle \{\mathtt{alien(X)}, \mathtt{isSonOf(X,Y)}, \mathtt{argentinian(Y)}\}, \mathtt{argentinian(X)} \rangle$

On the other hand, the same knowledge can be also expressed in the DL $\mathcal{EL}$ as follows:

$\mathtt{Alien} \sqcap \exists\mathtt{isSonOf.Argentinian} \sqsubseteq \mathtt{Argentinian}$

Note that similar to $\mathcal{A}_1$, in this case, the $\mathcal{EL}$ atom resulting from the formula above would contain two premises:

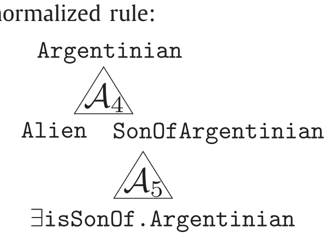
$\mathcal{A}_3 = \langle \{\mathtt{Alien}, \exists\mathtt{isSonOf.Argentinian}\}, \mathtt{Argentinian} \rangle$

However, once again, the configuration of atoms will depend on the specification of the argument language. As aforementioned, in order to take full advantage of the efficiency benefits for reasoning brought by the design of $\mathcal{EL}$, it is necessary to normalise the formula given above. The normalisation rules (Baader, 2003) bring about the following $\mathcal{EL}$-axioms:

$\mathtt{Alien} \sqcap \mathtt{SonOfArgentinian} \sqsubseteq \mathtt{Argentinian}$

$\exists\mathtt{isSonOf.Argentinian} \sqsubseteq \mathtt{SonOfArgentinian}$

In the figure depicted on the right we have one atom for each normalized rule:



$\mathcal{A}_4 = \langle \{\mathtt{Alien}, \mathtt{SonOfArgentinian}\}, \mathtt{Argentinian} \rangle$

$\mathcal{A}_5 = \langle \{\exists\mathtt{isSonOf.Argentinian}\}, \mathtt{SonOfArgentinian} \rangle$

The idea of argumental atom $\mathcal{A}_5$ bringing its claim for supporting one of the premises of atom $\mathcal{A}_4$ is to provide probative reason for $\mathcal{A}_4$ to effectively draw its claim. This interplay between pairs of atoms shows an intuition of their interrelation for a further construction of GenAF-arguments.

Since a premise is supported through the claim of other argument/s, both languages $\mathcal{L}_{\mathtt{pr}}$ and $\mathcal{L}_{\mathtt{cl}}$ should be interrelated and their expressivity controlled. In this sense, Definition 3.6 provides a characterization of the argument language $\mathbb{A}_\pi$ throughout restrictions on π in such a way that: 1) the interrelation between $\mathcal{L}_{\mathtt{pr}}$ and $\mathcal{L}_{\mathtt{cl}}$ could be modeled for ensuring that every describable premise would eventually be supported by formulæ expressible through the language for claims, and 2) rules of $\mathcal{L}^{\mathcal{K}}$ can be formatted to preserve an appropriate configuration for building argumental atoms. The objective of the second condition is to make sure that any formula $\varphi \in \mathcal{L}^{\mathcal{K}}$ can be expressed through a set of "atom like" formulæ. That is, since an atom is built from a set of premises for drawing a claim, we specify the pattern of formulæ as $(\bigwedge\Gamma) \rightarrow \alpha$, where $\alpha \in \mathcal{L}_{\mathtt{cl}}$ and Γ is a finite set such that $\Gamma \subseteq \mathcal{L}_{\mathtt{pr}}$. Note that the left-hand side of the pattern ends up describing a conjunction of premises while the right-hand side corresponds to the claim. Additionally, we need to make sure that any formula in $\mathcal{L}^{\mathcal{K}}$ can be represented through a set of "atom like" formulæ and vice versa in order to ensure a sane and complete argument representation language.

From now on, we will use the (usually subindexed) greek letters α to represent claims, ρ for premises, Γ for sets of premises, and φ for formulæ from $\mathcal{L}^{\mathcal{K}}$.

Definition 3.6 (Legal argument language)**.** Given an AL-framework $\pi = \langle \mathcal{L}^{\mathcal{K}}, \mathcal{L}_{\mathtt{cl}}, \mathcal{L}_{\mathtt{pr}}, \mathcal{L}_{\mathtt{a}} \rangle \in \mathbb{L}$, and assuming "$\models$" as the semantic entailment for $\mathcal{L}^{\mathcal{K}}$; the set $\mathbb{A}_\pi$ is a **legal argument language** *iff* π satisfies the following conditions:

1. for any $\rho \in \mathcal{L}_{\mathtt{pr}}$, there is a set $\Phi \subseteq \mathcal{L}_{\mathtt{cl}}$ such that $\Phi \models \rho$
2. for any $\varphi \in \mathcal{L}^{\mathcal{K}}$, there is a set $\Delta \subseteq \mathcal{L}^{\mathcal{K}}$ of formulæ $(\bigwedge\Gamma) \rightarrow \alpha$[6], where $\Gamma \subseteq \mathcal{L}_{\mathtt{pr}}$ and $\alpha \in \mathcal{L}_{\mathtt{cl}}$, *st.* $\Delta \models \varphi$

An argumental atom requires a set of premises to be satisfied –by considering the claims of other atoms– in order to reach its own claim. This interrelation among atoms is part of the GenAF's proof procedure, which relates to the proof procedures of the logical systems of natural deduction (Barker-Plummer, Barwise, & Etchemendy, 2011; Prawitz, 1965). Notice that we give the possibility to support a premise through a set of claims. The following example justifies this decision in an intuitive manner.

Example 3. (Continues from Example 1). We will consider also the following knowledge:

"Alien is everyone who was not born in Argentina".

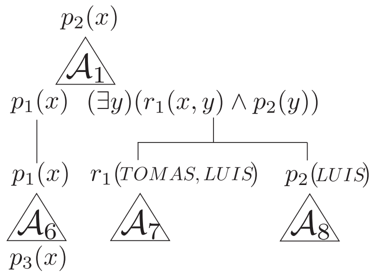
For its representation, we additionally use the unary predicate p_3 standing for "not born in Argentina". We will also consider the assertional knowledge $r_1(TOMAS, LUIS)$ and $p_2(LUIS)$ to logically represent that "Tomas is son of Luis" and that "Luis is Argentinian".

Note that argumental atom $\mathcal{A}_6 = \langle \{p_3(x)\}, p_1(x) \rangle$, standing for the FOL formula $p_3(x) \rightarrow p_1(x)$, supports through its claim to $\mathcal{A}_1$'s premise $p_1(x)$. On the other hand, for the second $\mathcal{A}_1$'s premise, it is necessary to provide more than a single atom to be supported. This kind of "collective" support will be formalized later on.

Note also that atoms $\mathcal{A}_7 = \langle \{\}, r_1(TOMAS, LUIS) \rangle$ and $\mathcal{A}_8 = \langle \{\}, p_2(LUIS) \rangle$ are evidential since they have empty sets of

[6] Referring to a syntax of conjunctive premises for the antecedent and a claim as the consequent. For instance, in DLs, the corresponding syntax would be $(\sqcap\Gamma) \sqsubseteq \alpha$.

premises and ground claims standing for assertional knowledge from the KB.



Now we have a formal instrument to represent knowledge in an "atom like" format in order to construct an argumentation framework whose fundamental elements, *i.e.*, argumental atoms, coincide with the fundamental knowledge structures from the underlying knowledge base (KB), *i.e.*, its formulæ. This allows to think each formula in the KB, as an argumental atom in the GenAF, which turns a KB into a repository of argumental atoms available for the consideration of the GenAF reasoning machinery.

Since a premise (or claim) of an atom is part of the left-hand side $\bigwedge\Gamma$ (or right-hand side α) of a statement in the KB, the languages $\mathcal{L}_{\mathtt{pr}}$ for premises and $\mathcal{L}_{\mathtt{cl}}$ for claims should be flexible enough to be able to model different alternative underlying general logics, identified as $\mathcal{L}^{\mathcal{K}}$. The need for such a flexibility becomes clearer when thinking of the several description logics (DLs) available to express ontologies. Recall that usually in DLs, description languages are specified through the definition of different sublanguages for left-hand and right-hand side formulæ with the goal of controlling expressivity in favor of the efficiency of reasoning. For handling these particular logics, the GenAF shows its full potential by providing the possibility to specify and interrelate the languages for premises and claims through a legal argument language. The following example illustrates an instantiation for the AL-framework which triggers a legal argument language according to Definition 3.6. Such an instantiation would allow the construction of the argumental atoms given in Example 3.

Example 4. (Continues from Example 3). In order to formally construct the argumental atoms given before, we need an instantiation for the AL-framework $\langle\mathcal{L}^{\mathcal{K}}, \mathcal{L}_{\mathtt{cl}}, \mathcal{L}_{\mathtt{pr}}, \mathcal{L}_{\mathtt{a}}\rangle \in \mathbb{L}$. A possibility is to assume $\mathcal{L}^{\mathcal{K}}$ as a logic $\mathcal{L}^2$ (arity ≤ 2), where the language $\mathcal{L}_{\mathtt{a}}$ contains only ground atoms and their negations, and the languages $\mathcal{L}_{\mathtt{cl}}$ and $\mathcal{L}_{\mathtt{pr}}$ are instantiated in a way that $\mathcal{L}_{\mathtt{cl}}$ allows disjunctions but prohibits conjunctions, while $\mathcal{L}_{\mathtt{pr}}$ avoids both conjunctions and disjunctions. For $\mathcal{L}_{\mathtt{pr}}$, it will be also possible to existentially quantify premises. In such a case, $\mathcal{L}_{\mathtt{pr}}$ will only admit a conjunction between a unary predicate and a relation (binary predicate). For simplicity, on absence of quantifier in a formula, universal quantification "$\forall$" is assumed.

$\mathcal{L}_{\mathtt{a}} ::= L \,|\, L(a) \,|\, L(a,b)$
$\mathcal{L}_{\mathtt{cl}} ::= \mathcal{L}_{\mathtt{a}} \,|\, P \,|\, \mathcal{L}_{\mathtt{cl}} \vee \mathcal{L}_{\mathtt{cl}}$
$\mathcal{L}_{\mathtt{pr}} ::= \mathcal{L}_{\mathtt{a}} \,|\, P \,|\, EDom \,|\, EImg$
$L: := p \,|\, \neg p$
$P: := L(x) \,|\, L(x, a) \,|\, L(a, x) \,|\, L(x, y)$
$EDom: := (\exists x)(L(x) \wedge L(x, y)) \,|\, (\exists x)(L(x) \wedge L(x, a))$
$EImg: := (\exists y)(L(y) \wedge L(x, y)) \,|\, (\exists y)(L(y) \wedge L(a, y))$

Now we can formally define the GenAF which will be constructed from an AL-framework π determining a legal argument language $\mathbb{A}_\pi$, and a set of argumental atoms.

Definition 3.7 (GenAF)**.** A **generalized abstract argumentation framework** (or GenAF, for short) is a pair $\langle\pi, \mathbf{A}\rangle$, where the AL-framework $\pi \in \mathbb{L}$ determines a legal argument language $\mathbb{A}_\pi$, and $\mathbf{A} \subseteq \mathbb{A}_\pi$ is a finite set of argumental atoms. The domain of GenAFs is identified through the set $\mathbb{F}$.

The set of arguments as well as the support, subargument, and attack relation sets are not made explicit in the abstract framework, but implicit in the light of the AL-framework through the appropriate constructions. When an underlying KB is specified and the logic $\mathcal{L}^{\mathcal{K}}$ (along with its entailment "$\models$") and a related AL-framework are concretized, an argumentation system can be built, where the set of arguments and the attack relation set will be made explicit, whereas the subargument and support will be implied by the construction of the set of arguments. This will be formalized later in Section 5 through the notion of Generalized Argumentation System (GenAS).

3.3. GenAF formal construction

A simple alternative for reasoning over inconsistent KBs –by relying upon argumentation theory is to build an argumentation system from the corresponding KB. This would involve the recognition of elementary pieces of knowledge from the KB for building argumental atoms. Such process could be computationally expensive not only regarding time, but also space. Complex axioms from the KB could imply the construction of several atoms for a further construction of arguments (and recognition of their interaction through a conflict relation). Moreover, when thinking of a dynamic environment, in which changes could affect the corresponding argumentation, avoiding to consider an argument for reasoning would imply the removal of an atom which would also involve removing some minimal piece of knowledge from the KB.

A better alternative would be to propose a *meta-normalization* for KBs in such a way that the normalized KB turns out being considered the main GenAF repository of argumental atoms. We refer to this process as *meta-normalization* given that we do not expect to give the formal proceeding for obtaining a normalized KB, but some general rules for validating the verification of such property by formulæ and KBs. The main idea is to translate formulæ into an "atom like" form such that considering a single normalized formula would make no practical distinction regarding the consideration of the structure of its related argumental atom. For that reason, we referred to such normalization form as *pre-Argumental*.

Definition 3.8 (pre-Argumental normal form)**.** Given a knowledge base $\Sigma \subseteq \mathcal{L}^{\mathcal{K}}$ and an AL-frame-work $\pi \in \mathbb{L}$; Σ conforms to the **(π)-pre-argumental normal form** (π-pANF) *iff* every formula $\varphi \in \Sigma$ corresponds to the form $(\bigwedge\Gamma) \to \alpha$, where $\alpha \in \mathcal{L}_{\mathtt{cl}}$ and $\Gamma \subseteq \mathcal{L}_{\mathtt{pr}}$. In such a case, each $\varphi \in \Sigma$ as well as the whole Σ, are said to be in π-pANF. In addition, we say $\mathcal{L}^{\mathcal{K}}$ is a π-pANF logic *iff* every $\vartheta \in \mathcal{L}^{\mathcal{K}}$ is in π-pANF.

The usage of a normalization function aims at abstracting away from any particular procedure for normalizing a knowledge base into its pre-argumental normal form. The normalization function $\mathfrak{panf}_\pi$ applied to a KB Σ will be used for referring to a normalized KB $\mathfrak{panf}_\pi(\Sigma)$. Such kind of functionality will be syntax-dependant upon a concrete AL-framework π.

Definition 3.9 (Normalization)**.** Given an AL-framework $\pi \in \mathbb{L}$; a function $\mathfrak{panf}_\pi : \wp(\mathcal{L}^{\mathcal{K}}) \longrightarrow \wp(\mathcal{L}^{\mathcal{K}})$ is a **pre-argumental normalization function** *iff* it translates any knowledge base $\Sigma \subseteq \mathcal{L}^{\mathcal{K}}$ into a logically equivalent π-pANF knowledge base $\mathfrak{panf}_\pi(\Sigma)$.

Observe that formulæ like $\alpha \in \mathcal{L}_{\mathtt{a}}$ or $\alpha \in \mathcal{L}_{\mathtt{cl}}$, are in π-pANF given that they correspond to the form $(\bigwedge\Gamma) \to \alpha$, with $\Gamma = \emptyset$.

Example 5. Let $\Sigma = \{(p_1(x) \wedge p_2(x)) \vee (p_3(x) \wedge p_4(x)) \to q_1(x) \wedge (q_2(x) \vee q_3(x)), (p_1(x) \to \neg p_1(x)), (p_1(x) \vee \neg r(x,y)), r(a,b)\}$ be an $\mathcal{L}^{\mathcal{K}}$ knowledge base. Its normalization (according to the AL-framework $\pi \in \mathbb{L}$ from Example 4) renders the π-pANF knowledge base:

$$\mathfrak{panf}_\pi(\Sigma) = \left\{\begin{array}{ll} \varphi_1: & (p_1(x) \wedge p_2(x) \to q_1(x)) \\ \varphi_2: & (p_1(x) \wedge p_2(x) \to q_2(x) \vee q_3(x)) \\ \varphi_3: & (p_3(x) \wedge p_4(x) \to q_1(x)) \\ \varphi_4: & (p_3(x) \wedge p_4(x) \to q_2(x) \vee q_3(x)) \\ \varphi_5: & (p_1(x) \to \neg p_1(x)) \\ \varphi_6: & (p_1(x) \vee \neg r(x,y)) \\ \varphi_7: & r(a,b) \end{array}\right\}$$

Note that formula $(p_1(x) \vee \neg r(x, y))$ in Example 5 might be transformed into a formula $(\neg p_1(x) \to \neg r(x, y))$, however this will depend on the conventions adopted by the normalization algorithm. We will abstract away from this kind of matters. Given an AL-framework $\pi \in \mathbb{L}$, for any atom $\mathcal{A} \in \mathbb{A}_\pi$, its claim and set of premises are identified by the functions $\mathfrak{cl} : \mathbb{A}_\pi \longrightarrow \mathcal{L}_{\mathtt{cl}}$, and $\mathfrak{pr} : \mathbb{A}_\pi \longrightarrow \wp(\mathcal{L}_{\mathtt{pr}})$, respectively. For instance, given $\mathcal{A} = \langle\{\rho_1, \rho_2\}, \alpha\rangle$, its premises are $\mathfrak{pr}(\mathcal{A}) = \{\rho_1, \rho_2\}$, and its claim, $\mathfrak{cl}(\mathcal{A}) = \alpha$.

We will refer as *atom constraints* to the logical conditions defined upon π-pANF formulæ $\varphi \in \mathcal{L}^\mathcal{K}$ for restricting the construction of φ-related argumental atoms $\mathcal{A}_\varphi \in \mathbb{A}_\pi$.

Definition 3.10 (Atom constraint)**.** Given an AL-framework $\pi \in \mathbb{L}$; an **atom constraint** (**AC**) is a logical condition upon π-pANF formulæ asserted through the use of any **atom constraint function** $\omega_\pi : \mathcal{L}^\mathcal{K} \longrightarrow \{\mathfrak{true}, \mathfrak{false}\}$, such that a formula $\varphi \in \mathcal{L}^\mathcal{K}$ **satisfies** a specific **AC** whenever its associated atom constraint function "ω_π" is such that $\omega_\pi(\varphi) = \mathfrak{true}$. For any π-pANF formula $(\bigwedge\Gamma) \to \alpha$ (or, $(\bigwedge\Gamma)(\bar{y}) \to \alpha(\bar{x})$, for making reference to variables), a non-exhaustive list of atom constraints is detailed next:

- **AC**$_1$. $\Gamma \cup \{\alpha\} \not\models \bot$ (consistency)
- **AC**$_2$. $\Gamma \not\models \alpha$ (non-circularity)
- **AC**$_3$. If $\bar{x} = \langle x_1, \ldots, x_n\rangle$ then $\bar{y} = \langle x_1, \ldots, x_n, y_1, \ldots, y_m\rangle$ (safeness)
- **AC**$_4$. for any subset of premises $\Gamma' \subseteq \Gamma$, if $\Gamma' \vDash \Gamma$ then $\Gamma' = \Gamma$ (premise minimality)

As aforementioned, the list of atom constraints given in Definition 3.10 is not exhaustive. However, it brings the possibility of studying the different alternatives for constructing argumental atoms which will serve for building GenAF-arguments. For instance, formulæ like $p_1(x) \to \neg p_1(x)$, $p_1(x) \to p_1(x)$, $p_1(x) \wedge p_2(x) \to r(x, y)$, and $p_1(x) \wedge (p_1(x) \vee p_2(x)) \to p_3(x)$ –assuming $\mathcal{L}_{\mathtt{pr}}$ admits disjunctions– would violate **AC**$_1$, **AC**$_2$, **AC**$_3$, and **AC**$_4$, respectively. This shows that atom constraints might be useful for discovering knowledge representation errors: for instance, we can suppose that failures of **AC**$_3$ and **AC**$_4$ are due to erroneous representations of formulæ like $p_1(x) \wedge p_2(y) \to r(x, y)$ and $p_1(x) \vee p_2(x) \to p_3(x)$, respectively. Besides, atom constraints are interesting for capturing the desired behavior of the theory, whichever it would be in any particular case, given that the user can define his own atom constraints functions. An *atom constraints set* is a structure containing atom constraints for the construction of atoms.

Definition 3.11 (Atom constraints set)**.** Given an AL-framework $\pi \in \mathbb{L}$; an **atom constraints set** Ω is any set of atom constraint functions ω_π. Given a π-pANF formula $\varphi \in \mathcal{L}^\mathcal{K}$, we say that φ **satisfies** Ω *iff* φ satisfies every **AC** modelled in Ω, *i.e.*, for any $\omega_\pi \in \Omega$, $\omega_\pi(\varphi) = \mathfrak{true}$.

We build atoms from π-pANF formulæ through an *atom translation function* "$\mathfrak{atom}_{[\pi,\Omega]}$" defined as follows.

Definition 3.12 (Atom translation function)**.** Given an AL-framework $\pi \in \mathbb{L}$ and an atom constraints set Ω; an **atom translation function** is a function $\mathfrak{atom}_{[\pi,\Omega]} : \mathcal{L}^\mathcal{K} \longrightarrow \mathbb{A}_\pi \cup \{\epsilon\}$ where $\mathfrak{atom}_{[\pi,\Omega]}(\varphi) = \langle\Gamma, \alpha\rangle$ *iff* $\varphi \in \mathcal{L}^\mathcal{K}$ is a π-pANF formula $(\bigwedge\Gamma) \to \alpha$ satisfying Ω. Otherwise, $\mathfrak{atom}_{[\pi,\Omega]}(\varphi) = \langle\emptyset, \bot\rangle = \epsilon$.

We use an *escape atom* $\epsilon = \langle\emptyset, \bot\rangle$ to identify those cases in which the atom translation function fails in the construction of an argumental atom due to either a non-π-pANF formula or a formula which does not satisfy the atom constraints set Ω. Observe that, although the special atom ϵ verifies the formatting syntax of an argumental atom, it falls outside of the scope of the argumental atom's domain, *i.e.*, $\epsilon \notin \mathbb{A}_\pi$. The reason is that ϵ does not satisfy the consistency property from Definition 3.3.

Example 6. (Continues from Example 5. Assuming a constraint set Ω modeling **AC**$_1$, the results of the function "$\mathfrak{atom}_{[\pi,\Omega]}$" for obtaining argumental atoms from the formulæ contained in $\mathfrak{panf}_\pi(\Sigma)$ is:

$$\mathcal{A}_1 = \mathfrak{atom}_{[\pi,\Omega]}(\varphi_1) = \langle\{p_1(x), p_2(x)\}, q_1(x)\rangle$$
$$\mathcal{A}_2 = \mathfrak{atom}_{[\pi,\Omega]}(\varphi_2) = \langle\{p_1(x), p_2(x)\}, q_2(x) \vee q_3(x)\rangle$$
$$\mathcal{A}_3 = \mathfrak{atom}_{[\pi,\Omega]}(\varphi_3) = \langle\{p_3(x), p_4(x)\}, q_1(x)\rangle$$
$$\mathcal{A}_4 = \mathfrak{atom}_{[\pi,\Omega]}(\varphi_4) = \langle\{p_3(x), p_4(x)\}, q_2(x) \vee q_3(x)\rangle$$
$$\mathcal{A}_5 = \mathfrak{atom}_{[\pi,\Omega]}(\varphi_5) = \langle\{\}, \bot\rangle = \epsilon$$
$$\mathcal{A}_6 = \mathfrak{atom}_{[\pi,\Omega]}(\varphi_6) = \langle\{\}, p_1(x) \vee \neg r(x,y)\rangle (primitive)$$
$$\mathcal{A}_7 = \mathfrak{atom}_{[\pi,\Omega]}(\varphi_7) = \langle\{\}, r(a,b)\rangle (evidence)$$

For the previous example, if we would not that assume **AC**$_1$ is modeled by Ω, then $\mathfrak{atom}_{[\pi,\Omega]}(\varphi_5)$ would trigger the structure $\langle\{p_1(x)\}, \neg p_1(x)\rangle$, however it would still not be an atom given that it fails verifying consistency from Definition 3.3. This shows that **AC**$_1$ is not necessary for ensuring the well behavior of the theory. Nevertheless, for completeness purposes **AC**$_1$ has been specified.

Atoms appear in a GenAF as a consequence of an existing piece of knowledge from the underlying KB. This establishes a relation between premises and claims wrt. the KB.

Remark 3.13. Given an AL-framework $\pi \in \mathbb{L}$ and a π-pANF KB $\Sigma \subseteq \mathcal{L}^\mathcal{K}$, a formula $\varphi \in \Sigma$, and its associated atom $\mathfrak{atom}_{[\pi,\Omega]}(\varphi) = \langle\Gamma, \alpha\rangle$; it follows $\Sigma \models (\bigwedge\Gamma) \to \alpha$, but $\Gamma \vDash \alpha$ does not necessarily hold.

The remark above makes explicit that –given a π-pANF knowledge base $\Sigma \subseteq \mathcal{L}^\mathcal{K}$– having a formula like $(\rho_1 \wedge \ldots \wedge \rho_n \to \alpha) \in \Sigma$ means that $\Sigma \models (\rho_1 \wedge \ldots \wedge \rho_n \to \alpha)$ but it does not necessarily mean that $\{\rho_1, \ldots, \rho_n\} \models \alpha$ holds. This shows that the use of atoms is slightly different from the natural meaning of argument for which its support set (or body) is a minimal set for inferring the claim.

Next we introduce the notion of *generalized argumentation theory* (GenAT) which describes a tuple enclosing the fundamental theoretic elements for the construction of a related GenAF. That is, the $\mathcal{L}^\mathcal{K}$-KB from where the set of argumental atoms will be obtained, a concrete AL-framework specifying a legal argument language along with the general logic $\mathcal{L}^\mathcal{K}$ and the languages for claims, premises, and assertions for specifying the inner structure of atoms, the concretization of a pre-argumental normalization function for obtaining a π-pANF KB as a repository of argumental atoms, an atom constraint set for making explicit the conditions required for the obtention of atoms, and the atom translation function.

Definition 3.14 (Generalized argumentation theory)**.** We say $\tau \in \mathbb{T}$ is a **generalized argumentation theory** (or GenAT, for short) *iff* $\tau = \langle\Sigma, \pi, \mathfrak{panf}_\pi, \Omega, \mathfrak{atom}_{[\pi,\Omega]}\rangle$ is a tuple consisting of a KB $\Sigma \subseteq \mathcal{L}^\mathcal{K}$, an AL-framework $\pi \in \mathbb{L}$, a pre-argumental normalization function $\mathfrak{panf}_\pi$, an atom constraints set Ω, and an atom translation function $\mathfrak{atom}_{[\pi,\Omega]}$. For such a GenAT, we say that τ is **associated** to Σ. The set $\mathbb{T}$ identifies the formal domain of GenATs.

Definition 3.15 (Theory function)**.** Given a GenAT $\tau \in \mathbb{T}$, a function $\mathfrak{genaf} : \mathbb{T} \longrightarrow \mathbb{F}$ is a **theory function** *iff* $\mathfrak{genaf}(\tau) = \langle \pi, \mathbf{A} \rangle$ where for every $\varphi \in \mathfrak{panf}_\pi(\Sigma)$, if $\mathfrak{atom}_{[\pi,\Omega]}(\varphi) \in \mathbb{A}_\pi$ then $\mathfrak{atom}_{[\pi,\Omega]}(\varphi) \in \mathbf{A}$.

Upon a specific GenAT $\tau \in \mathbb{T}$ we defined above the *theory function* "$\mathfrak{genaf}$" to construct the GenAF associated to a knowledge base $\Sigma \subseteq \mathcal{L}^\mathcal{K}$. Note that, in addition, it could be also necessary to consider *contrapositive* formulæ for building atoms. This is natural since contrapositive formulæ are implicitly considered for reasoning in $\mathcal{L}^\mathcal{K}$ through *modus tollens*. Nonetheless, in a GenAF, this needs to be made explicit due to the nature of the notion of atom and the way they interrelate with each other for building arguments –as we will see later in Section 4. From now on, we identify the *contrapositive* of any formula $\varphi \in \mathcal{L}^\mathcal{K}$, by writing φ^-. Note however that it may happen for a given formula $\varphi \in \mathfrak{panf}_\pi(\Sigma)$ that its contrapositive φ^- falls outside the logic $\mathcal{L}^\mathcal{K}$. Thus, we should verify if $\mathfrak{atom}_{[\pi,\Omega]}(\varphi^-) \in \mathbb{A}_\pi$ holds for incorporating $\mathfrak{atom}_{[\pi,\Omega]}(\varphi^-)$ into $\mathbf{A}$, not only for avoiding violation of atom constraints but also for verifying that the resulting atom corresponds to the argument language $\mathbb{A}_\pi$. Consequently, we define the *closure under transposition* as an alternative for theory functions considering contrapositive formulæ.

Definition 3.16 (Closure under transposition)**.** Given a GenAT $\tau \in \mathbb{T}$ and the related GenAF $\delta \in \mathbb{F}$ obtained through a theory function $\delta = \mathfrak{genaf}(\tau) = \langle \pi, \mathbf{A} \rangle$, we say that "$\mathfrak{genaf}$" is a **transpositive theory function** *iff* for every $\varphi \in \mathfrak{panf}_\pi(\Sigma)$, if $\mathfrak{atom}_{[\pi,\Omega]}(\varphi^-) \in \mathbb{A}_\pi$ then $\mathfrak{atom}_{[\pi,\Omega]}(\varphi^-) \in \mathbf{A}$. Finally, a GenAF $\delta \in \mathbb{F}$ is referred as **closed under transposition** *iff* δ is obtained through a transpositive theory function.

The idea of building atoms in pairs is to make sure that, whenever it is possible, for any formula $\varphi \in \mathfrak{panf}_\pi(\Sigma)$ we will have two related atoms $\mathcal{A}_{(\varphi)} \in \mathbb{A}_\pi$ and $\mathcal{A}_{(\varphi^-)} \in \mathbb{A}_\pi$, such that $\mathcal{A}_{(\varphi^-)}$ has premises (claim) constructed from the claim (set of premises) of $\mathcal{A}_{(\varphi)}$. In this sense, the construction of contrapositive formulæ is disambiguated by constructing the left-hand (right-hand) side of φ^- uniquely from the right-hand (left-hand) side of φ. For instance, given a formula φ: $p_1(x) \wedge p_2(x) \rightarrow q_1(x)$, we will consider $\varphi^- : \neg q_1(x) \rightarrow \neg p_1(x) \vee \neg p_2(x)$ as its unique contrapositive formula.

Example 7. (Continues from Example 6). Constituted the corresponding GenAT $\tau \in \mathbb{T}$, a transpositive theory function renders the GenAF $\mathfrak{genaf}(\tau) = \langle \pi, \mathbf{A} \rangle$, where $\mathbf{A} = \{\mathcal{A}_1, \mathcal{A}_2, \mathcal{A}_3, \mathcal{A}_4, \mathcal{A}_6, \mathcal{A}_7, \mathcal{A}'_1, \mathcal{A}'_2, \mathcal{A}'_3, \mathcal{A}'_4\}$.

$$\mathcal{A}'_1 = \langle \{\neg q_1(x)\}, \neg p_1(x) \vee \neg p_2(x) \rangle$$
$$\mathcal{A}'_2 = \langle \{\neg q_2(x), \neg q_3(x)\}, \neg p_1(x) \vee \neg p_2(x) \rangle$$

$$\mathcal{A}'_3 = \langle \{\neg q_1(x)\}, \neg p_3(x) \vee \neg p_4(x) \rangle$$
$$\mathcal{A}'_4 = \langle \{\neg q_2(x), \neg q_3(x)\}, \neg p_3(x) \vee \neg p_4(x) \rangle$$

Note that the results of $\mathfrak{atom}_{[\pi,\Omega]}(\varphi_5)$, $\mathfrak{atom}_{[\pi,\Omega]}((\varphi_5)^-)$, $\mathfrak{atom}_{[\pi,\Omega]}((\varphi_6)^-)$, and $\mathfrak{atom}_{[\pi,\Omega]}((\varphi_7)^-)$, are always the escape atom ϵ given that each of them does not verify $\mathbf{AC}_1$ and afterwards, $\epsilon \notin \mathbb{A}_\pi$, *i.e.*, they are not atoms since consistency from Definition 3.3 is violated.

4. The GenAF argumentation machinery

In this section we provide the fundamentals for building GenAF-arguments and further recognizing conflicts between pairs of arguments. These two constructions give rise to the *generalized argumentation system* (GenAS) that will be specified later in Section 5.

4.1. Argument construction

The idea of generalizing an abstract argumentation framework arises from the need of managing different argument languages specified through FOL fragments. The main purpose is the formalization of a framework which would be capable of handling a wide range of description languages for inconsistency-tolerant ontology reasoning. Thus, assuming an AL-framework $\pi \in \mathbb{L}$, for the argument language $\mathbb{A}_\pi$, several possibilities may arise; for instance, the language for claims may accept disjunction of formulæ. In such a case, it would be possible to infer a formula in $\mathcal{L}_{\mathtt{cl}}$– *i.e.*, a new claim– through considering together the claims of several argumental atoms in the GenAF. For instance, given two argumental atoms $\langle \{p_1(x)\}, q_1(x) \vee q_2(x) \rangle$ and $\langle \{p_2(x)\}, \neg q_2(x) \rangle$, the claim $q_1(x)$ may be inferred. This sort of construction that we referred as *claim-coalition*, will be included in the construction of GenAF-arguments, for bringing the argument's claim.

Before formalizing the notion of claim-coalition, we will discuss the inconveniences that may arise as a result of bringing together atoms in a coalition for a collaborative purpose. We need to make sure that the atoms which will compose a coalition could not trigger inconsistencies by themselves, independently of the way they interrelate each other, or even, independently of the way the coalition could be related with other atoms outside of the coalition. Even more, we need to make sure that the coalition does not stand for an incoherent set of formulæ (see Section 3.1). This will be important for ensuring that the construction of arguments will be possible given that an argument must be a consistent structure.

Three types of *clashes* may arise in a set of atoms: clashes between 1) claims, 2) claims and premises, and 3) premises. For the former case, it is clear that contradictory claims should not be considered together since the construction of an argument will require supporting all the atoms involved in it, which means supporting their premises and thus, ultimately drawing every atom's claim. The second case has two alternatives: a) considering two atoms like $\langle \{p\}, q \rangle$ and $\langle \{\neg q\}, p' \rangle$, supporting one of them implies the impossibility of supporting the second atom (having p implies q and therefore, $\neg q$ cannot be supported; and for the other way around, having $\neg q$ implies that p could not be supported since this would infer q). For the second alternative within type 2, b) having a pair of atoms like $\langle \{p\}, q \rangle$ and $\langle \{q\}, \neg p \rangle$, where the former atom supports the second one, implies that premise p could never be supported since this would infer also $\neg p$. For the latter type of inconsistency, 3) a clash of premises in a pair of atoms like $\langle \{\neg p\}, q \rangle$ and $\langle \{p\}, q' \rangle$ implies once again that supporting one of them makes impossible to support the second one. Hence, it is natural to think that a set of atoms should ensure consistency among the claims and premises of all its elements in order to be safe for a later construction of arguments.

Definition 4.1 (Consistent set of argumental atoms)**.** Given a GenAF $\langle \pi, \mathbf{A} \rangle \in \mathbb{F}$; a set of argumental atoms $\mathbf{A}' \subseteq \mathbf{A}$ is **consistent** *iff* $\mathfrak{prset}(\mathbf{A}') \cup \mathfrak{clset}(\mathbf{A}') \not\models \bot$ holds, where the functions $\mathfrak{clset} : \wp(\mathbb{A}_\pi) \longrightarrow \wp(\mathcal{L}_{\mathtt{cl}})$ and $\mathfrak{prset} : \wp(\mathbb{A}_\pi) \longrightarrow \wp(\mathcal{L}_{\mathtt{pr}})$ are defined as $\mathfrak{clset}(\mathbf{A}') = \{\mathfrak{cl}(\mathcal{A}) | \mathcal{A} \in \mathbf{A}'\}$ and $\mathfrak{prset}(\mathbf{A}') = \bigcup_{\mathcal{A} \in \mathbf{A}'} \mathfrak{pr}(\mathcal{A})$, to respectively identify the set of claims and premises from atoms in $\mathbf{A}'$.

Intuitively, a *claim-coalition* is a structure that might be seen as *a consistent set of argumental atoms which is minimal for inferring a given claim.*

Definition 4.2 (Claim-Coalition)**.** Given a GenAF $\langle \pi, \mathbf{A} \rangle \in \mathbb{F}$, a triple $\mathcal{C} = \langle \Gamma, \mathbf{A}', \alpha \rangle$ is a **claim-coalition** for α *iff* $\Gamma \subseteq \mathcal{L}_{\mathtt{pr}}$ is its set of premises, $\mathbf{A}' \subseteq \mathbf{A}$ its body, and $\alpha \in \mathcal{L}_{\mathtt{cl}}$ its claim; and the following conditions are verified by assuming an interpretation model $\mathcal{I}$ and a substitution v:

1. $\mathfrak{clset}(\mathbf{A}') \models_{(\mathcal{I},v)} \alpha$ **(claim)**
2. $\Gamma = \mathfrak{prset}(\mathbf{A}')[v]$ **(premises)**
3. $\mathbf{A}'$ is a consistent set of argumental atoms **(consistency)**
4. there is no claim-coalition $\mathcal{C}' = \langle \Gamma, \mathbf{A}'', \alpha \rangle$ such that $\mathbf{A}'' \subset \mathbf{A}'$ **(minimality)**

Functions $\mathfrak{pr}$, $\mathfrak{bd}$, and $\mathfrak{cl}$ are overloaded to identify the coalition's premises, body and claim, respectively. A claim-coalition $\mathcal{C}$ is referred as **primitive** *iff* it contains a single atom, *i.e.*, $|\mathfrak{bd}(\mathcal{C})| = 1$.

It is important to note that given a coalition $\mathcal{C}$ for α, the interpretation model $\mathcal{I}$ and substitution v are built in order to ensure that $\mathfrak{clset}(\mathbf{A}') \models \alpha$ holds, *i.e.*, $\mathfrak{clset}(\mathbf{A}') \models_{(\mathcal{I},v)} \alpha$. As a result, the set of premises of the coalition ends up substituted through v. Next, we exemplify such a situation.

Example 8. Given two argumental atoms $\mathcal{A}_1 = \langle \{p_1(x)\}, q_1(x) \vee q_2(x) \rangle$ and $\mathcal{A}_2 = \langle \{\}, \neg q_2(a) \rangle$, a claim-coalition $\mathcal{C}$ for $q_1(a)$ can be constructed, where $\mathfrak{bd}(\mathcal{C}) = \{\mathcal{A}_1, \mathcal{A}_2\}$. It is easy to see that this will be possible only through a substitution $[\{x/a\}]$. Observe that in such a case, the premise $p_1(x)$ of $\mathcal{A}_1$ turns to $p_1(a)$ in $\mathcal{C}$, *i.e.*, while $\mathfrak{pr}(\mathcal{A}_1) = \{p_1(x)\}$ holds, we also have that $\mathfrak{pr}(\mathcal{C}) = (\mathfrak{pr}(\mathcal{A}_1) \cup \mathfrak{pr}(\mathcal{A}_2))[\{x/a\}] = \{p_1(x)\}[\{x/a\}] = \{p_1(a)\}$ hold. Afterwards, $\mathcal{C} = \langle \{p_1(a)\}, \{\mathcal{A}_1, \mathcal{A}_2\}, q_1(a) \rangle$ holds.

Definition 4.3 (Supporter)**.** Given an AL-framework $\pi \in \mathbb{L}$, a premise $\rho \in \mathcal{L}_{\mathtt{pr}}$, an interpretation model $\mathcal{I}$, and a substitution v; a set of claims $\Phi \subseteq \mathcal{L}_{\mathtt{cl}}$ is a **supporter of** ρ **through** $(\mathcal{I}, v)$ *iff* $\Phi \models_{(\mathcal{I},v)} \rho$, $\Phi \not\models_{(\mathcal{I},v)} \perp$ (consistency), and there is no $\Phi' \subset \Phi$ such that $\Phi' \models_{(\mathcal{I},v)} \rho$ (minimality).

We will refer to an unsupported premise as *free*.

Definition 4.4 (Free premises)**.** Given a GenAF $\langle \pi, \mathbf{A} \rangle \in \mathbb{F}$, a consistent set of argumental atoms $\mathbf{A}' \subseteq \mathbf{A}$, a premise $\rho \in \mathcal{L}_{\mathtt{pr}}$, an interpretation model $\mathcal{I}$, and a substitution v; we say ρ is a **free premise through** $(\mathcal{I}, v)$ **wrt. A′** *iff* there is no supporter $\Phi \subseteq \mathfrak{clset}(\mathbf{A}')$ of ρ through $(\mathcal{I}, v)$.

Next we formalize the notion of GenAF-argument through a recursive definition. Intuitively, a GenAF-argument is constructed through a claim-coalition, which provides the argument's claim, and a set of GenAF-arguments for supporting each of the premises of the claim-coalition. However, sometimes it will not be possible to support each of the coalition's premises. This will render a sort of "incomplete arguments" which we will refer as *potential arguments* since they could easily turn to *full arguments* if we would admit a dynamic environment in which new atoms could be introduced. Moreover, the identification of conflicting pairs of potential arguments will be useful for detecting sources of *incoherences* in the underlying knowledge base (see Section 3.1). Next, we formalize the notion of *argumental structure*, as the formal construction from which GenAF-arguments and potential arguments will arise.

Definition 4.5 (Argumental structure)**.** Given a GenAF $\langle \pi, \mathbf{A} \rangle \in \mathbb{F}$, a triple $\mathcal{S} = \langle \Gamma, \mathbf{A}', \alpha \rangle$ is an **argumental structure** (or just, **structure**) for α *iff* $\Gamma \subseteq \mathcal{L}_{\mathtt{pr}}$ is its set of premises, $\mathbf{A}' \subseteq \mathbf{A}$ its body, and $\alpha \in \mathcal{L}_{\mathtt{cl}}$ its claim; and the following conditions are verified by assuming a common interpretation model $\mathcal{I}$ and a substitution v:

1. there is a claim-coalition $\mathcal{C}$ for α *st.* $\mathfrak{bd}(\mathcal{C}) \subseteq \mathbf{A}'$ **(claim)**
2. there is $\Gamma_{\mathcal{C}} \subseteq \mathfrak{pr}(\mathcal{C})$ *st.* for each $\rho \in \Gamma_{\mathcal{C}}$ there are structures $\mathcal{S}_1, \ldots, \mathcal{S}_n$ (**supporting structures**) *st.*:
   (a) $\Phi = \{\mathfrak{cl}(\mathcal{S}_1), \ldots, \mathfrak{cl}(\mathcal{S}_n)\}$ is a supporter of ρ through $(\mathcal{I}, v)$ **(support)**
   (b) $\mathbf{A}_\rho = \mathfrak{bd}(\mathcal{S}_1) \cup \ldots \cup \mathfrak{bd}(\mathcal{S}_n)$
   (c) $\Gamma_\rho = \mathfrak{pr}(\mathcal{S}_1) \cup \ldots \cup \mathfrak{pr}(\mathcal{S}_n)$
3. $\mathbf{A}' = \mathfrak{bd}(\mathcal{C}) \cup \bigcup_{\rho \in \Gamma_{\mathcal{C}}} \mathbf{A}_\rho$ **(body)**
4. $\Gamma = ((\mathfrak{pr}(\mathcal{C}) \setminus \Gamma_{\mathcal{C}}) \cup \bigcup_{\rho \in \Gamma_{\mathcal{C}}} \Gamma_\rho)[v]$ **(premises)**
5. every $\rho \in \Gamma$ is a free premise through $(\mathcal{I}, v)$ wrt. $\mathbf{A}'$ **(free-premises)**
6. $\mathbf{A}'$ is a consistent set of argumental atoms **(consistency)**
7. there is no argumental structure $\mathcal{S}' = \langle \Gamma, \mathbf{A}'', \alpha \rangle$ such that $\mathbf{A}'' \subset \mathbf{A}'$ **(minimality)**
8. $\mathbf{A}_\rho \subset \mathbf{A}'$, for every $\rho \in \Gamma_{\mathcal{C}}$ **(non-circularity)**

Functions $\mathfrak{pr}$, $\mathfrak{bd}$, and $\mathfrak{cl}$ are overloaded to identify the structure's premises, body and claim, respectively. An argumental structure $\mathcal{S}$ is a **full argument** (or just, **argument**) *iff* $\mathfrak{pr}(\mathcal{S}) = \emptyset$. Whereas $\mathcal{S}$ is a **potential argument** *iff* $\mathfrak{pr}(\mathcal{S}) \neq \emptyset$. A structure $\mathcal{S}$ is referred as **primitive** *iff* $|\mathfrak{bd}(\mathcal{S})| = 1$.

The construction of argumental structures relies on the specification of its corresponding triple, where the claim α is provided by the claim-coalition constructed for α, the body is conformed by the argumental atoms conforming both the coalition and the supporting structures, and a set of premises composed by the coalition's unsupported premises and the premises corresponding to the supporting structures. The set of premises must be ensured to be free wrt. the body of the structure (see condition 5 referred as free-premises) given that this guarantees the complete determination of roles played by the atoms of the structure. In addition, an argumental structure must satisfy the properties of consistency, minimality, and non-circularity. Consistency ensures that the argumental structure will be free of inner conflicts, minimality ensures that each atom included in the structure plays a unique and specific role for its construction, and finally, non-circularity ensures that there is possible construction of a structure relying on a cyclic support. For a deep understanding of this latter property, suppose we have two argumental atoms $\mathcal{A}_1 = \langle \{p\}, q \rangle$ and $\mathcal{A}_2 = \langle \{q\}, p \rangle$, and suppose condition 8 (non-circularity) is not available. In such a case, a pair of –malformed– structures $\mathcal{S}_1 = \langle \{\}, \{\mathcal{A}_1, \mathcal{A}_2\}, p \rangle$ and $\mathcal{S}_2 = \langle \{\}, \{\mathcal{A}_1, \mathcal{A}_2\}, q \rangle$ would be constructed given that for $\mathcal{S}_1$, a primitive claim-coalition $\mathcal{C} = \langle \{q\}, \{\mathcal{A}_2\}, p \rangle$ would provide the claim and $\mathfrak{pr}(\mathcal{C})$ would not be free wrt. $\mathfrak{bd}(\mathcal{S}_1) = \{\mathcal{A}_1, \mathcal{A}_2\}$ given that $\mathcal{S}_2$ would support it. Afterwards, it is easy to see that the properties of consistency, and minimality are verified. Observe that, structure $\mathcal{S}_2$ would be constructed analogously. Non-circularity avoids supporting a premise of the claim-coalition through a set of structures whose bodies are constructed through exactly the same body of the main structure. Thus, for the case of $\mathcal{S}_1$, structure $\mathcal{S}_2$ could not support $\mathfrak{pr}(\mathcal{C})$ given that $\mathfrak{bd}(\mathcal{S}_2) \not\subset \mathfrak{bd}(\mathcal{S}_1)$, violating non-circularity.

Note that a primitive structure $\mathcal{S}$ contains a single argumental atom $\mathcal{A}$ in its body, and both $\mathfrak{pr}(\mathcal{A}) = \mathfrak{pr}(\mathcal{S})$ and $\mathfrak{cl}(\mathcal{A}) = \mathfrak{cl}(\mathcal{S})$ hold. However, not every argumental atom has an associated primitive structure; for instance, no primitive structure could contain an argumental atom like $\langle \{p\}, p \rangle$ given that it would be impossible to satisfy condition 8 (non-circularity). Moreover, such an atom cannot be part of any structure, given that it would also violate condition 7 (minimality), given that it will not provide any advance to the support process. Observe however, that whenever an atom constraint function $\omega_\pi \in \Omega$ models $\mathbf{AC}_2$, such kind of argumental atoms would not even be part of the GenAF, given that they would not arise from the application of the translation function $\mathfrak{atom}_{[\pi, \Omega]}$.

Definition 4.6 (Substructure)**.** Given a GenAF $\langle \pi, \mathbf{A} \rangle \in \mathbb{F}$ and two structures $\mathcal{S}_1$ and $\mathcal{S}_2$, $\mathcal{S}_1$ is a **substructure** of $\mathcal{S}_2$, noted as $\mathcal{S}_1 \trianglelefteq \mathcal{S}_2$ *iff* $\mathfrak{bd}(\mathcal{S}_1) \subseteq \mathfrak{bd}(\mathcal{S}_2)$ holds.[7] Similarly, $\mathcal{S}_1$ is a **proper substructure** of $\mathcal{S}_2$, *i.e.*, $\mathcal{S}_1 \triangleleft \mathcal{S}_2$ *iff* $\mathfrak{bd}(\mathcal{S}_1) \subset \mathfrak{bd}(\mathcal{S}_2)$ holds.

Example 9. (Continues from Example 3). Depicted in Fig. 1, we show the construction of an argumental structure stand-

[7] Note that this relation is reflexive, *i.e.*, for any structure $\mathcal{S}$, it follows $\mathcal{S} \trianglelefteq \mathcal{S}$.

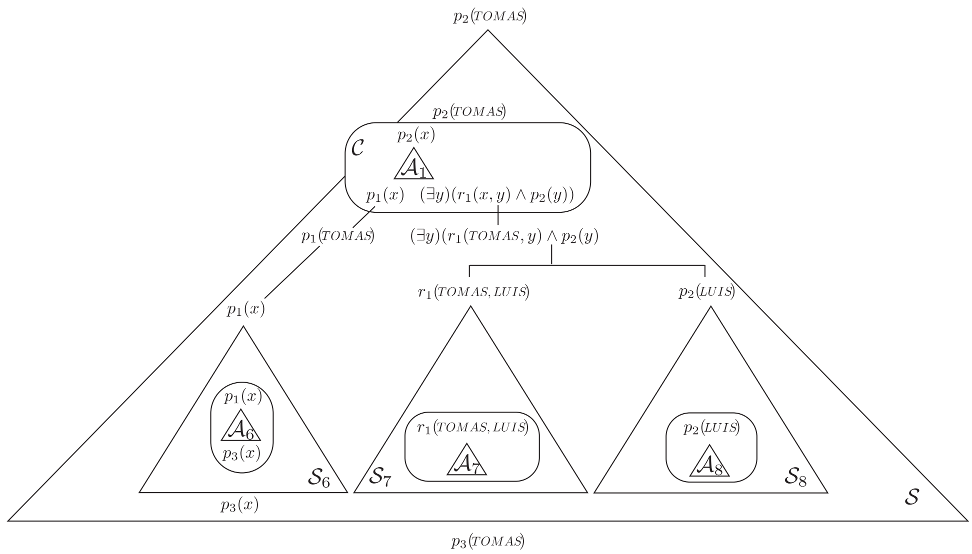


Fig. 1. Structure $\mathcal{S}$ from Example 9.

ing for *TOMAS is Argentinian, i.e.*, a structure $\mathcal{S}$ for $p_2(TOMAS)$. Following Definition 4.5 in detail, we have a claim-coalition $\mathcal{C} = \langle\{p_1(TOMAS), (\exists y)(r_1(TOMAS, y) \wedge p_2(y))\}, \{\mathcal{A}_1\}, p_2(TOMAS)\rangle$, which provides its claim for the main structure $\mathcal{S}$. Premise $\rho_1 = p_1(TOMAS) \in \mathfrak{pr}(\mathcal{C})$ is supported by structure $\mathcal{S}_6 = \langle\{p_3(x)\}, \{\mathcal{A}_6\}, p_1(x)\rangle$ given that $\{p_1(x)\} \models_{(\mathcal{I},\nu)} p_1(TOMAS)$, where $\{x/TOMAS\} \subseteq \nu$, and on the other hand, $\rho_2 = (\exists y)(r_1(TOMAS, y) \wedge p_2(y)) \in \mathfrak{pr}(\mathcal{C})$ is supported by the primitive structures $\mathcal{S}_7 = \langle\{\}, \{\mathcal{A}_7\}, r_1(TOMAS, LUIS)\rangle$ and $\mathcal{S}_8 = \langle\{\}, \{\mathcal{A}_8\}, p_2(LUIS)\rangle$ given that $\{\mathfrak{cl}(\mathcal{S}_7), \mathfrak{cl}(\mathcal{S}_8)\} \models_{(\mathcal{I},\nu)} \rho_2$, considering $\{y/LUIS\} \subseteq \nu$. Observe that structure $\mathcal{S} = \langle\{p_3(TOMAS)\}, \{\mathcal{A}_1, \mathcal{A}_6, \mathcal{A}_7, \mathcal{A}_8\}, p_2(TOMAS)\rangle$ is built through a substitution $\nu = \{x/TOMAS, y/LUIS\}$, and that $\mathcal{S}$ ends up being a potential argument since it has one free-premise, *i.e.*, $p_3(TOMAS)$ cannot be supported considering atoms from $\mathfrak{bd}(\mathcal{S})$. It is easy to see that consistency, minimality, and non-circularity are also verified.

Afterwards, if we assume a new piece of evidence standing for *TOMAS was not born in Argentina*, we would need to consider the additional atom $\mathcal{A}_9 = \langle\{\}, p_3(TOMAS)\rangle$ where $\mathcal{A}_9 \in \mathbf{A}$. The resulting argument $\mathcal{S}' = \langle\{\}, \{\mathcal{A}_1, \mathcal{A}_6, \mathcal{A}_7, \mathcal{A}_8, \mathcal{A}_9\}, p_2(TOMAS)\rangle$ is depicted in Fig. 2. The structure $\mathcal{S}_{10} = \langle\{\}, \{\mathcal{A}_6, \mathcal{A}_9\}, p_1(TOMAS)\rangle$ will be supporting ρ_1 in the context of $\mathcal{S}'$.

Example 10 illustrates the formation of structures where an atom plays more than a particular role. That is, atom $\mathcal{A}_1$ is used as part of a claim-coalition $\mathcal{C}_2$ for the main structure $\mathcal{S}_2$ given the corresponding substitution ν, and a supporting structure $\mathcal{S}_1$ contained in the main one also makes usage of the same atom for its own purpose through its own substitution. This example is interesting for showing how variable instantiations may take place in the construction of structures.

Example 10. Given GenAF $\langle\pi, \mathbf{A}\rangle \in \mathbb{F}$, $\{\mathcal{A}_1, \mathcal{A}_2, \mathcal{A}_3, \mathcal{A}_4\} \subseteq \mathbf{A}$ where $\mathcal{A}_1 = \langle\{p(x)\}, (\exists y)(\neg r(x, y) \vee p(y))\rangle$, $\mathcal{A}_2 = \langle\{\}, r(a, b)\rangle$, $\mathcal{A}_3 = \langle\{\}, p(a)\rangle$, and $\mathcal{A}_4 = \langle\{\}, r(b, c)\rangle$. Structure $\mathcal{C}_1 = \langle\{p(a)\}, \{\mathcal{A}_1, \mathcal{A}_2\}, p(b)\rangle$ is a claim-coalition for $p(b)$ given that $\{(\exists y)(\neg r(x, y) \vee p(y)), r(a, b)\} \models_{(\mathcal{I},\nu)} p(b)$, where the substitution $\nu = \{x/a, y/b\}$ holds. Afterwards, given that $\mathfrak{pr}(\mathcal{C}_1)$ can be supported through a primitive argumental structure $\langle\{\}, \{\mathcal{A}_3\}, p(a)\rangle$, it ends up constructed the argument $\mathcal{S}_1 = \langle\{\}, \{\mathcal{A}_1, \mathcal{A}_2, \mathcal{A}_3\}, p(b)\rangle$ through the substitution ν. On the other hand, argument $\mathcal{S}_1$ is a supporting structure for the premise $p(b)$ of the claim-coalition $\mathcal{C}_2 = \langle\{p(b)\}, \{\mathcal{A}_1, \mathcal{A}_4\}, p(c)\rangle$, which appears through a substitution $\nu' = \{x/b, y/c\}$ in the context of the argumental structure $\mathcal{S}_2 = \langle\{\}, \{\mathcal{A}_1, \mathcal{A}_2, \mathcal{A}_3, \mathcal{A}_4\}, p(c)\rangle$. Fig. 3 illustrates the case.

The *set of structures* of a GenAF δ will enclose all the argumental structures constructible from δ.

Definition 4.7 (Set of structures of a GenAF)**.** Given a GenAF $\delta = \langle\pi, \mathbf{A}\rangle \in \mathbb{F}$, the set $\mathbf{S}_\delta$ is the **set of δ-structures** *iff* $\mathbf{S}_\delta$ is the set of all argumental structures $\mathcal{S}$ constructible from δ.

4.2. Conflict recognition

For recognizing conflicts in a GenAF, we will rely upon the usual notions of *rebuttals, undercut*, and *direct undercut* (Gorogiannis & Hunter, 2011). A slight reformulation of such notions is done by relying upon inconsistency in place of verification of complementary literals. As aforementioned, such a decision is crucial for certain representation languages like description logics, where negation of axioms may fall out of the scope of the description language (see consistency-negation and coherence-negation on page 3). We say that two argumental structures $\mathcal{S}_1$ and $\mathcal{S}_2$ are in conflict whenever their claims cannot be assumed together (*rebuttals*), whenever the claim of $\mathcal{S}_1$ is inconsistent with a premise of $\mathcal{S}_2$ (*direct undercut*), or when $\mathcal{S}_1$ is a rebuttal of some proper substructure of $\mathcal{S}_2$ (*undercut*).

Definition 4.8 (Conflictive structures)**.** Given a GenAF $\delta \in \mathbb{F}$ and its associated set $\mathbf{S}_\delta$ of δ-structures; two structures $\mathcal{S}_1 \in \mathbf{S}_\delta$ and $\mathcal{S}_2 \in \mathbf{S}_\delta$ are in **conflict** *iff* it follows:

(rebuttals) $\{\mathfrak{cl}(\mathcal{S}_1), \mathfrak{cl}(\mathcal{S}_2)\} \models \bot$ (in which case $\mathcal{S}_2$ **rebuts** $\mathcal{S}_1$, and viceversa), or
(direct undercut) $\{\mathfrak{cl}(\mathcal{S}_1), \rho\} \models \bot$, where $\rho \in \mathfrak{pr}(\mathcal{S}_2)$ (in which case $\mathcal{S}_1$ **directly undercuts** $\mathcal{S}_2$), or
(undercut) $\mathcal{S}_1$ rebuts $\mathcal{S}'_2$, where $\mathcal{S}'_2 \lhd \mathcal{S}_2$ (in which case $\mathcal{S}_1$ **undercuts** $\mathcal{S}_2$).

Note that conflictive structures may involve potential arguments. This alternative is important to identify sources of incoherency (see Section 3.1) from the underlying knowledge base. As seen before, an incoherence formula is a potential source of inconsistency that can be triggered by incorporating assertions supporting the left-hand side of the formula. This intuition can be extended to pairs of argumental structures. For instance, two conflictive potential arguments do not stand for a source of inconsistency

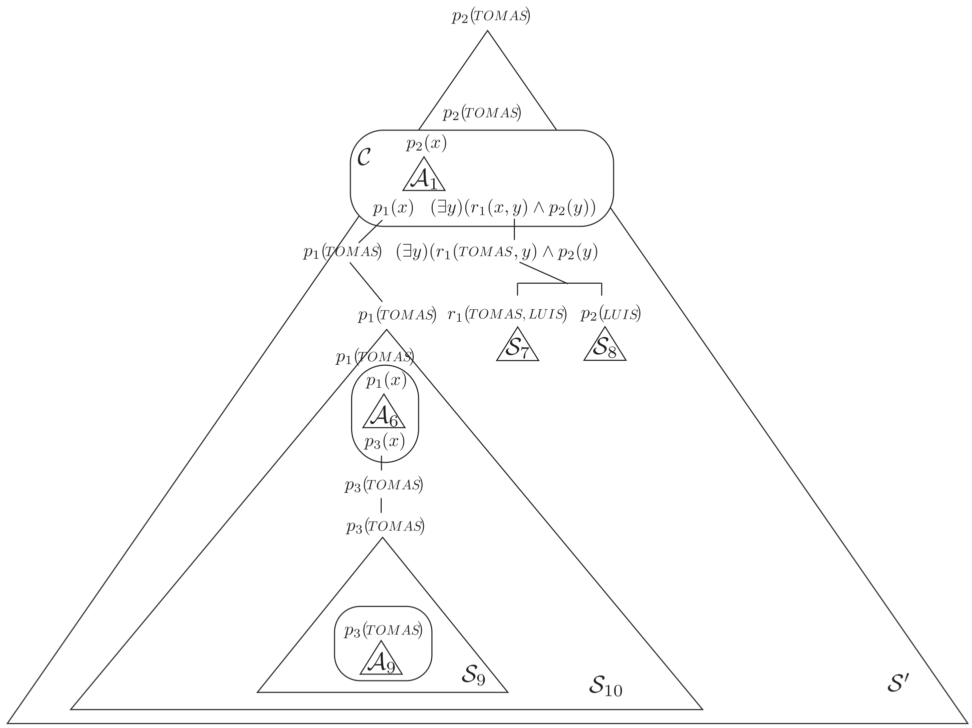


Fig. 2. Structure $\mathcal{S}'$ from Example 9.

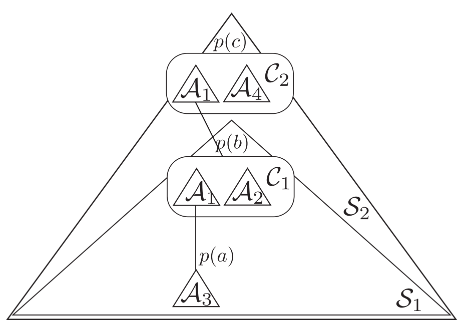


Fig. 3. Structures from Example 10.

unless their premises are simultaneously supported through the incorporation of new argumental atoms. In such a case, the potential arguments turn into full conflicting arguments. In this way, the recognition of conflicting potential arguments is a way to anticipate conflicts in dynamic environments. Standard logic-based argumentation systems do not handle conflicts at a coherency level. This is an interesting contribution of GenAFs for a wide range of research areas like ontology debugging. However, we will not go deeper on this matter here since it is not the main focus of this article. Investigations on this subject are underway.

In order to definitely decide which structure succeeds from a conflictive pair of rebuttals, an *atom comparison criterion* $\succeq \subseteq \mathbf{A} \times \mathbf{A}$ is assumed to be determined from a comparison criterion among formulæ in the KB –which could be defined for instance, upon relevance or reliability of knowledge, *i.e.*, a measure of relevance can be assigned to formulæ in the KB; by default any two formulæ in a KB are assumed to be equally reliable unless the contrary is stated. We will overload the definition of the atom comparison criterion extending its usage to structures, *i.e.*, $\succeq \subseteq \mathbf{S}_\delta \times \mathbf{S}_\delta$; hence, two structures $\mathcal{S}_1$ and $\mathcal{S}_2$ are assumed to be ordered by following some *structure ordering methodology* defined upon the criterion "$\succeq$". The expression $\mathcal{S}_1 \succcurlyeq \mathcal{S}_2$ means that $\mathcal{S}_1$ is more important/reliable than $\mathcal{S}_2$. Different structure ordering methodologies can be defined. In this article we will assume a simple one: $\mathcal{S}_1 \succcurlyeq \mathcal{S}_2$ implies either that 1) both structures are incomparable, which means that there is no $\mathcal{A} \in \mathfrak{bd}(\mathcal{S}_1)$ and no $\mathcal{A}' \in \mathfrak{bd}(\mathcal{S}_2)$ such that $\mathcal{A} \succcurlyeq \mathcal{A}'$; or 2) there is some $\mathcal{A} \in \mathfrak{bd}(\mathcal{S}_1)$ such that for every $\mathcal{A}' \in \mathfrak{bd}(\mathcal{S}_2)$ it follows $\mathcal{A} \succcurlyeq \mathcal{A}'$. The attack relation $\mathbf{R}_\delta$ of a GenAF $\delta \in \mathbb{F}$, is finally adjudicated in terms of the atom comparison criterion "$\succeq$".

Definition 4.9 (Attack relation)**.** Given a GenAF $\delta \in \mathbb{F}$ and its associated set $\mathbf{S}_\delta$ of δ-structures; the set $\mathbf{R}_\delta \subseteq \mathbf{S}_\delta \times \mathbf{S}_\delta$ is the **attack relation** *iff* $\mathbf{R}_\delta = \{(\mathcal{S}_1, \mathcal{S}_2) | (\mathcal{S}_1 \in \mathbf{S}_\delta$ rebuts $\mathcal{S}_2 \in \mathbf{S}_\delta$ and $\mathcal{S}_1 \succcurlyeq \mathcal{S}_2)$ or $(\mathcal{S}_1 \in \mathbf{S}_\delta$ undercuts, or directly undercuts, $\mathcal{S}_2 \in \mathbf{S}_\delta)\}$. The infix notation is used, writing $\mathcal{S}_1 \mathbf{R}_\delta \mathcal{S}_2$, to state $\mathcal{S}_1$ **attacks/defeats** $\mathcal{S}_2$.

Example 11. Let $\delta = \langle \pi, \mathbf{A} \rangle \in \mathbb{F}$ be a GenAF, where $\mathbf{A} = \{\mathcal{A}_1, \mathcal{A}_2, \mathcal{A}_3\}$ where $\mathcal{A}_1 = \langle \{p_1(x)\}, p_2(x) \rangle$, $\mathcal{A}_2 = \langle \{p_1(x)\}, \neg p_3(x) \rangle$, $\mathcal{A}_3 = \langle \{p_1(x), \neg p_2(x)\}, p_3(x) \rangle$. Consider the primitive structures $\mathcal{S}_1 = \langle \{p_1(x)\}, \{\mathcal{A}_1\}, p_2(x) \rangle$, $\mathcal{S}_2 = \langle \{p_1(x)\}, \{\mathcal{A}_2\}, \neg p_3(x) \rangle$, and $\mathcal{S}_3 = \langle \{p_1(x), \neg p_2(x)\}, \{\mathcal{A}_3\}, p_3(x) \rangle$, configuring three potential arguments. We obtain the following conflicts: $\mathcal{S}_1$ directly undercuts $\mathcal{S}_3$, $\mathcal{S}_2$ rebuts $\mathcal{S}_3$, and $\mathcal{S}_3$ rebuts $\mathcal{S}_2$. However, the rebuttal determined by $\mathcal{S}_2$ and $\mathcal{S}_3$ is avoided to be symmetric within $\mathbf{R}_\delta$ by relying upon the atom comparison criterion, assuming $\mathcal{A}_3 \succcurlyeq \mathcal{A}_2$. Thus, the attack set –whose graph is depicted in Fig. 4– ends up being $\mathbf{R}_\delta = \{(\mathcal{S}_3, \mathcal{S}_2), (\mathcal{S}_1, \mathcal{S}_3)\}$.

Now suppose we have a second GenAF $\delta' = \langle \pi, \mathbf{A}' \rangle \in \mathbb{F}$ where $\mathbf{A}' = \mathbf{A} \cup \{\mathcal{A}_4, \mathcal{A}_5\}$, and where $\mathcal{A}_4 = \langle \{\}, p_1(a) \rangle$ and $\mathcal{A}_5 = \langle \{\}, \neg p_2(a) \rangle$. In addition, we will have the full arguments $\mathcal{S}_4 = \langle \{\}, \{\mathcal{A}_4, \mathcal{A}_1\}, p_2(a) \rangle$, $\mathcal{S}_5 = \langle \{\}, \{\mathcal{A}_4, \mathcal{A}_2\}, \neg p_3(a) \rangle$, and $\mathcal{S}_6 = \langle \{\}, \{\mathcal{A}_4, \mathcal{A}_5, \mathcal{A}_3\}, p_3(a) \rangle$. Also, the following primitive arguments appear $\mathcal{S}_7 = \langle \{\}, \{\mathcal{A}_4\}, p_1(a) \rangle$, and $\mathcal{S}_8 = \langle \{\}, \{\mathcal{A}_5\}, \neg p_2(a) \rangle$. In this case, the resulting conflicts are: $\mathcal{S}_4$ undercuts $\mathcal{S}_6$ given that $\mathcal{S}_4$ rebuts $\mathcal{S}_8$ and $\mathcal{S}_8 \lhd \mathcal{S}_6$, $\mathcal{S}_8$ rebuts $\mathcal{S}_4$, $\mathcal{S}_6$ rebuts $\mathcal{S}_5$, and $\mathcal{S}_5$ rebuts $\mathcal{S}_6$.

However, the rebuttals determined by $\mathcal{S}_5$ and $\mathcal{S}_6$, and $\mathcal{S}_4$ and $\mathcal{S}_8$, are avoided to be symmetric by relying upon the atom comparison criterion, assuming $\mathcal{A}_3 \succcurlyeq \mathcal{A}_2$ and $\mathcal{A}_1 \succcurlyeq \mathcal{A}_5$. Thus, the at-

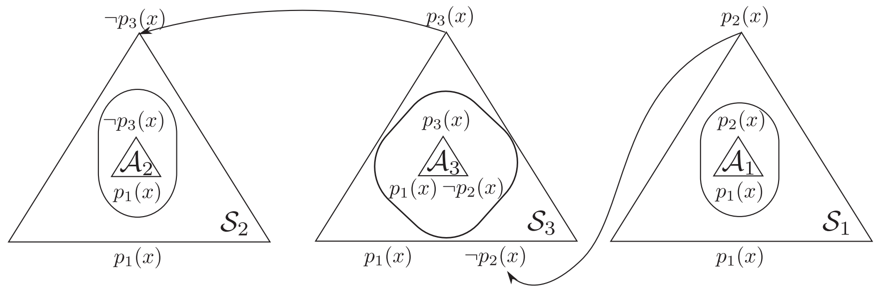


Fig. 4. Graph of structures and attacks for $\langle \pi, \mathbf{A}\rangle$ – Example 11.

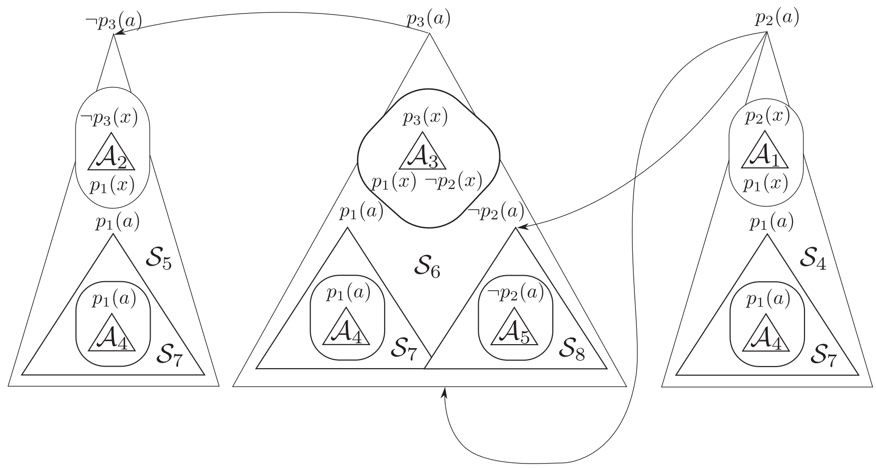


Fig. 5. Graph of structures and attacks for $\langle \pi, \mathbf{A}'\rangle$ – Example 11.

tack set –whose graph is depicted in Fig. 5– ends up being $\mathbf{R}_{\delta'} = \{(\mathcal{S}_6, \mathcal{S}_5), (\mathcal{S}_4, \mathcal{S}_8), (\mathcal{S}_4, \mathcal{S}_6)\}$.

The definition of the relation $\mathbf{R}_\delta$ allows to recognize sources of information for both inconsistencies and incoherencies from the knowledge base Σ. However, it could be also necessary to identify only the cases of inconsistencies. To this end, we define the *consistency attack relation* $\mathbf{C}_\delta \subseteq \mathbf{R}_\delta$. In addition, we define the *base consistency attack relation* $\mathbf{C}^b_\delta \subseteq \mathbf{C}_\delta$ as the set containing the smaller conflicting pairs from $\mathbf{C}_\delta$.

Definition 4.10 (Consistency Attack)**.** Given a GenAF $\delta \in \mathbb{F}$ and its associated sets $\mathbf{S}_\delta$ and $\mathbf{R}_\delta$, the set $\mathbf{C}_\delta \subseteq \mathbf{R}_\delta$ is the **consistency attack relation** *iff* $\mathbf{C}_\delta = \{(\mathcal{S}_1, \mathcal{S}_2) \in \mathbf{R}_\delta | \mathcal{S}_1$ and $\mathcal{S}_2$ are full arguments$\}$. In addition, the set $\mathbf{C}^b_\delta \subseteq \mathbf{R}_\delta$ is the **base consistency attack relation** *iff* $\mathbf{C}^b_\delta = \{(\mathcal{S}_1, \mathcal{S}_2) \in \mathbf{C}_\delta |$ for any $\mathcal{S}'_1 \trianglelefteq \mathcal{S}_1$ and any $\mathcal{S}'_2 \trianglelefteq \mathcal{S}_2$ *st.* $(\mathcal{S}'_1, \mathcal{S}'_2) \in \mathbf{C}_\delta$, it holds $\mathcal{S}'_1 = \mathcal{S}_1$ and $\mathcal{S}'_2 = \mathcal{S}_2\}$.

It is easy to see that any base consistency attack is a rebutting conflict between full arguments.

Remark 4.11. Given a GenAF $\delta \in \mathbb{F}$ and its associated sets $\mathbf{S}_\delta$ and $\mathbf{R}_\delta$, $(\mathcal{S}_1, \mathcal{S}_2) \in \mathbf{C}^b_\delta$ *iff* $\mathcal{S}_1 \in \mathbf{S}_\delta$ and $\mathcal{S}_2 \in \mathbf{S}_\delta$ are full arguments *st.* $\mathcal{S}_1$ rebuts $\mathcal{S}_2$.

5. The GenAF reasoner

A full computation of the set of δ-structures and the attack relation enables the theoretical analysis of our proposal. Through the reference of a GenAS it will be possible to study acceptability of arguments and their properties upon rationality postulates for argumentation as given in Amgoud (2014).

Definition 5.1 (Generalized argumentation system)**.** A triple $\langle \delta, \mathbf{S}_\delta, \mathbf{R}_\delta\rangle$ is a **generalized argumentation system** (or GenAS, for short) *iff* $\delta \in \mathbb{F}$ is a GenAF, $\mathbf{S}_\delta$ is the set of δ-structures, and $\mathbf{R}_\delta \subseteq \mathbf{S}_\delta \times \mathbf{S}_\delta$ is the attack relation between pairs of conflicting structures from $\mathbf{S}_\delta$. The set $\mathbb{G}$ stands for identifying the GenAS-domain.

Given a GenAT $\tau \in \mathbb{T}$, the following *constructor* is proposed for building the related GenAS.

Definition 5.2 (GenAS constructor)**.** Given a GenAT $\tau \in \mathbb{T}$, a function $\mathfrak{genas} : \mathbb{T} \longrightarrow \mathbb{G}$ is a GenAS**constructor** *iff* $\mathfrak{genas}(\tau) = \langle \delta, \mathbf{S}_\delta, \mathbf{R}_\delta\rangle$ is the GenAS constructed from the GenAF $\delta = \mathfrak{genaf}(\tau)$.

The *base function* is defined for identifying the set of $\mathcal{L}^\mathcal{K}$-formulæ from which a set of argumental structures can be constructed. This will be useful for showing GenAS properties.

Definition 5.3 (Base function)**.** Given a GenAT $\tau = \langle \Sigma, \pi, \mathfrak{panf}_\pi, \Omega, \mathfrak{atom}_{[\pi,\Omega]}\rangle \in \mathbb{T}$ and the associated GenAS $\mathfrak{genas}(\tau) = \langle \delta, \mathbf{S}_\delta, \mathbf{R}_\delta\rangle \in \mathbb{G}$, a function $\mathfrak{base} : \wp(\mathbf{S}_\delta) \longrightarrow \wp(\mathcal{L}^\mathcal{K})$ is the **base function** *iff* for any set $\mathbf{S} \subseteq \mathbf{S}_\delta$, $\mathfrak{base}(\mathbf{S}) \subseteq \mathfrak{panf}_\pi(\Sigma)$ is defined as: $\mathfrak{base}(\mathbf{S}) = \{\varphi \in \mathfrak{panf}_\pi(\Sigma) |$ for every $\mathcal{S} \in \mathbf{S}$ and every $\mathcal{A} \in \mathfrak{bd}(\mathcal{S})$, $\mathcal{A} = \mathfrak{atom}_{[\pi,\Omega]}(\varphi)$ or $\mathcal{A} = \mathfrak{atom}_{[\pi,\Omega]}(\varphi^-)\}$

The following properties evaluate the relation between sources of inconsistency/incoherence from an underlying knowledge base regarding the existence of attack relations in a related GenAS.

Lemma 5.4. *Given a* GenAT $\tau = \langle \Sigma, \pi, \mathfrak{panf}_\pi, \Omega, \mathfrak{atom}_{[\pi,\Omega]}\rangle \in \mathbb{T}$ *and the* GenAS $\mathfrak{genas}(\tau) = \langle \delta, \mathbf{S}_\delta, \mathbf{R}_\delta\rangle$*; if* Σ *is consistent then* $\mathbf{C}_\delta = \emptyset$.

Applying contrapositive reasoning to the previous property we can infer the following corollary.

Corollary 5.5. *Given a* GenAT $\tau = \langle \Sigma, \pi, \mathfrak{panf}_\pi, \Omega, \mathfrak{atom}_{[\pi,\Omega]}\rangle \in \mathbb{T}$ *and the* GenAS $\mathfrak{genas}(\tau) = \langle \delta, \mathbf{S}_\delta, \mathbf{R}_\delta\rangle$*; if* $(\mathcal{S}_1, \mathcal{S}_2) \in \mathbf{C}^b_\delta$ *then* $\mathfrak{base}(\{\mathcal{S}_1, \mathcal{S}_2\})$ *is a source of inconsistency in* $\mathfrak{panf}_\pi(\Sigma)$.

Theorem 5.6. *Given a* GenAT $\tau = \langle \Sigma, \pi, \mathfrak{panf}_\pi, \Omega, \mathfrak{atom}_{[\pi,\Omega]} \rangle \in \mathbb{T}$ *and the* GenAS $\mathfrak{genas}(\tau) = \langle \delta, \mathbf{S}_\delta, \mathbf{R}_\delta \rangle$*; if* Σ *is coherent and consistent then* $\mathbf{R}_\delta = \emptyset$.

Corollary 5.7. *Given a* GenAT $\tau = \langle \Sigma, \pi, \mathfrak{panf}_\pi, \Omega, \mathfrak{atom}_{[\pi,\Omega]} \rangle \in \mathbb{T}$ *and the* GenAS $\mathfrak{genas}(\tau) = \langle \delta, \mathbf{S}_\delta, \mathbf{R}_\delta \rangle$*; if* $(\mathcal{S}_1, \mathcal{S}_2) \in \mathbf{R}_\delta$ *then* $\mathfrak{base}(\{\mathcal{S}_1, \mathcal{S}_2\})$ *is either a source of incoherence or inconsistency in* $\mathfrak{panf}_\pi(\Sigma)$.

The converse of each property given above is not true in any case, since a formula $\varphi \in \Sigma$ may be such that $\varphi \vDash \perp$, and therefore $\mathfrak{atom}_{[\pi,\Omega]}(\varphi) \notin \mathbb{A}_\pi$. This means that $\mathbf{R}_\delta$ may still be empty but Σ not necessarily consistent and/or coherent. More on this matter will be discussed later in Section 7.

From Corollary 5.7, it is clear that each attack in a GenAS implies a source of inconsistency/incoherence in the underlying Σ. Since the objective of a GenAF is to reason over inconsistent knowledge bases, there is a need for a mechanism for allowing to obtain from the related GenAS those arguments that prevail over the rest, *i.e.*, 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* (Baroni & Giacomin, 2007; Dung, 1995). These semantics ensure the obtention of conflict-free sets of arguments, namely *extensions*; that is, the set of accepted arguments calculated following any of these semantics is such that no pair of conflictive arguments appears in that same extension. Next we introduce the Dung's standard semantics adapted for dealing with the GenAS specification.

Definition 5.8 (Conflict-freeness and defense)**.** Let $\langle \delta, \mathbf{S}_\delta, \mathbf{R}_\delta \rangle \in \mathbb{G}$ be a GenAS and $\mathbf{E} \subseteq \mathbf{S}_\delta$ a set of argumental structures.

- **E** is **conflict-free** *iff* there exist no pair of structures $\mathcal{S}_1 \in \mathbf{E}$ and $\mathcal{S}_2 \in \mathbf{E}$ such that $\mathcal{S}_1 \mathbf{R}_\delta \mathcal{S}_2$.
- **E defends** a structure $\mathcal{S} \in \mathbf{S}_\delta$ ($\mathcal{S}$ is **acceptable** wrt. **E**) *iff* for any structure $\mathcal{S}_1 \in \mathbf{S}_\delta$ such that $\mathcal{S}_1 \mathbf{R}_\delta \mathcal{S}$ there exists a structure $\mathcal{S}_2 \in \mathbf{E}$ such that $\mathcal{S}_2 \mathbf{R}_\delta \mathcal{S}_1$. A set of argumental structures is acceptable wrt. **E** when each structure in it is acceptable wrt. **E**. Finally, **E** is called **self-acceptable** when **E** is acceptable wrt. **E**, *i.e.*, **E** defends all its structures.

Definition 5.9 (Admissible sets)**.** Let $\langle \delta, \mathbf{S}_\delta, \mathbf{R}_\delta \rangle \in \mathbb{G}$ be a GenAS, a set $\mathbf{E} \subseteq \mathbf{S}_\delta$ of argumental structures is said **admissible** *iff* **E** is both self-acceptable and conflict-free.

The expression "Dung's standard semantics" is usually used to refer to *complete, grounded, preferred*, and *stable semantics* which appeared in Dung (1995)[8]. Besides, the unqualified term *extension* can be also used to refer to a complete, grounded, preferred, or stable extension. The importance of the notion of admissible sets is reflected by the fact that every extension under any of the Dung's standard semantics is admissible.

Definition 5.10 (Dung's standard semantics for GenAFs)**.** Let $\langle \delta, \mathbf{S}_\delta, \mathbf{R}_\delta \rangle \in \mathbb{G}$ be a GenAS, $\mathbf{E} \subseteq \mathbf{S}_\delta$ a conflict-free set of argumental structures, and $\mathcal{F} : \wp(\mathbf{S}_\delta) \longrightarrow \wp(\mathbf{S}_\delta)$ the **characteristic function** defined as $\mathcal{F}(\mathbf{E}) = \{\mathcal{S} \in \mathbf{S}_\delta \mid \mathbf{E} \text{ defends } \mathcal{S}\}$. The following are the **(Dung's) standard semantics** for GenAFs:

- **E** is a **complete extension** *iff* it is admissible and it holds $\mathbf{E} = \mathcal{F}(\mathbf{E})$.
- **E** is a **grounded extension** *iff* it is the minimal (wrt. set-inclusion from $\mathbf{S}_\delta$) complete extension.
- **E** is a **preferred extension** *iff* it is a maximal (wrt. set-inclusion from $\mathbf{S}_\delta$) complete extension.
- **E** is a **stable extension** *iff* it is a preferred extension and for any structure $\mathcal{S} \in (\mathbf{S}_\delta \setminus \mathbf{E})$ there is a structure $\mathcal{S}' \in \mathbf{E}$ such that $\mathcal{S}' \mathbf{R}_\delta \mathcal{S}$.

Multiple extensions may arise from any semantics except for the *grounded* whose outcome is always a unique single extension. This might require to make a choice among the resulting extensions. On the other hand, the outcome of both the *grounded* and *stable* semantics may be the empty set. In the sequel and just for simplicity, we refer to a *semantics* "$\mathfrak{s}$" for identifying some specific standard semantics according to Definition 5.10. Moreover, we refer to a $\mathfrak{s}$*-extension* to identify an extension obtained according to the semantics "$\mathfrak{s}$".

Definition 5.11 (Set of Extensions of a GenAS)**.** Given a GenAS $\sigma = \langle \delta, \mathbf{S}_\delta, \mathbf{R}_\delta \rangle \in \mathbb{G}$ and a semantics "$\mathfrak{s}$", a function $\mathfrak{ext}_\mathfrak{s} : \mathbb{G} \longrightarrow \wp(\wp(\mathbf{S}_\delta))$ identifies the **set of** $\mathfrak{s}$**-extensions** of σ *iff* $\mathfrak{ext}_\mathfrak{s}(\sigma)$ is the set of all extensions $\mathbf{E} \subseteq \mathbf{S}_\delta$ modeling the semantics "$\mathfrak{s}$". In addition, we say that $\mathfrak{ext}_\mathfrak{s}(\sigma^{\mathbf{C}})$ is the set of all **consistency** $\mathfrak{s}$**-extensions** $\mathbf{E} \subseteq \mathbf{S}_\delta$ of the GenAS $\sigma^{\mathbf{C}} = \langle \delta, \mathbf{S}_\delta, \mathbf{C}_\delta \rangle$. In such a case, we write $\mathfrak{ext}^{\mathbf{C}}_\mathfrak{s}(\sigma)$ to refer to $\mathfrak{ext}_\mathfrak{s}(\sigma^{\mathbf{C}})$.

Queries made to the argumentation framework are resolved by looking for argumental structures supporting them inside extensions built according to the adopted argumentation semantics. A satisfied query renders a *warranted* formula. Note that the expressivity of queries will correspond to $\mathcal{L}^\kappa$ but fulfilling the requirements of a π-pANF formula.

Definition 5.12 (Warrant)**.** Given a knowledge base $\Sigma \subseteq \mathcal{L}^\kappa$, the associated GenAT $\tau \in \mathbb{T}$, and a semantics $\mathfrak{s}$; a π-pANF formula $\vartheta \in \mathcal{L}^\kappa$ is said to be **warranted** from Σ *iff* there is some argumental structure $\mathcal{S} \in \mathbf{E}$ in some extension $\mathbf{E} \in \mathfrak{ext}_\mathfrak{s}(\mathfrak{genas}(\tau))$, such that $(\bigwedge \mathfrak{pr}(\mathcal{S})) \to \mathfrak{cl}(\mathcal{S}) = \vartheta$. We will write $\Sigma \mid\!\approx_{[\tau,\mathfrak{s}]} \vartheta$ to state that ϑ is warranted from Σ according to the associated GenAT τ and the semantics $\mathfrak{s}$. In addition, we will write $\Sigma \mid\!\approx^{\mathbf{C}}_{[\tau,\mathfrak{s}]} \vartheta$ to state that the argumental structure $\mathcal{S}$ is in some extension $\mathbf{E} \in \mathfrak{ext}^{\mathbf{C}}_\mathfrak{s}(\mathfrak{genas}(\tau))$. In such a case, we say ϑ is said to be **consistently warranted** from *KB*.

The following example shows the intuitions behind a warrant condition of a formula.

Example 12. Suppose we have a KB $\Sigma = \{p_1, (p_1 \wedge p_2 \to q)\}$. A structure $\mathcal{S} = \langle \{p_2\}, \{\mathcal{A}_1, \mathcal{A}_2\}, q \rangle$ would be constructed in the related GenAF, where $\mathcal{A}_1 = \langle \{p_1, p_2\}, q \rangle$ and $\mathcal{A}_2 = \langle \{\}, p_1 \rangle$. In the case where $\mathcal{S}$ is contained in some extension, it means that the structure has been accepted by the adopted argumentation semantics $\mathfrak{s}$. Hence, it is natural to expect the GenAF obtained through the GenAT τ, to warrant $p_2 \to q$, *i.e.*, $\Sigma \mid\!\approx_{[\tau,\mathfrak{s}]} (p_2 \to q)$. On the other hand, by assuming the primitive structure $\mathcal{S}' = \langle \{\}, \{\mathcal{A}_2\}, p_1 \rangle$ is contained in some $\mathfrak{s}$-extension, we would obtain $\Sigma \mid\!\approx_{[\tau,\mathfrak{s}]} p_1$, where p_1 ends up warranted.

The following proposition makes explicit the existing relation between GenAF-structures and formulæ from an underlying π-pANF knowledge base. This intuition was previously introduced in Section 3.3 and discussed through Remark 3.13. Proposition 5.13 formally shows the well construction of structures in accordance to such intuitions.

Proposition 5.13. *Given a* GenAS $\langle \delta, \mathbf{S}_\delta, \mathbf{R}_\delta \rangle \in \mathbb{G}$*, if* $\mathcal{S} \in \mathbf{S}_\delta$ *then* $\mathfrak{base}(\{\mathcal{S}\}) \models (\bigwedge \mathfrak{pr}(\mathcal{S})) \to \mathfrak{cl}(\mathcal{S})$.

Given a knowledge base $\Sigma \subseteq \mathcal{L}^\kappa$, it is possible to reason over inconsistency by relying upon the operator "$\mid\!\approx_{[\tau,\mathfrak{s}]}$". Note that

[8] Other argumentation semantics, like the semi-stable and ideal semantics, could also be applied for GenAFs without inconvenience. For simplicity, we just consider the four original semantics appearing in Dung's seminal work (Dung, 1995).

under consistent knowledge bases, "$\approx_{[\tau,\mathfrak{s}]}$" behaves as the classical entailment "$\vDash$".

Theorem 5.14. *Given a knowledge base $\Sigma \subseteq \mathcal{L}^{\kappa}$ and a π-pANF formula $\vartheta \in \mathcal{L}^{\kappa}$, if Σ is coherent and consistent then $\Sigma \vDash \vartheta$ iff $\Sigma \approx_{[\tau,\mathfrak{s}]} \vartheta$.*

The *set of warranted conclusions* will be the set of sets containing all the warranted formulæ from each extension.

Definition 5.15 (Set of warranted conclusions)**.** Given a knowledge base $\Sigma \subseteq \mathcal{L}^{\kappa}$, the associated GenAT $\tau \in \mathbb{T}$, and a semantics $\mathfrak{s}$; a function $\mathfrak{warrant}_{[\tau,\mathfrak{s}]} : \wp(\mathcal{L}^{\kappa}) \longrightarrow \wp(\wp(\mathcal{L}^{\kappa}))$ is a **warrant function** *iff* $\mathfrak{warrant}_{[\tau,\mathfrak{s}]}(\Sigma) = \{\{(\bigwedge \mathfrak{pr}(\mathcal{S})) \rightarrow \mathfrak{cl}(\mathcal{S}) | \mathcal{S} \in \mathbf{E}\} | \mathbf{E} \in \mathfrak{ext}_{\mathfrak{s}}(\mathfrak{genas}(\tau))\}$. For referring to the **set of warranted consistent conclusions**, we write $\mathfrak{warrant}^{\mathbf{C}}_{[\tau,\mathfrak{s}]}(\Sigma)$ which is analogously constructed from all extensions $\mathbf{E} \in \mathfrak{ext}^{\mathbf{C}}_{\mathfrak{s}}(\mathfrak{genas}(\tau))$.

Some authors claim that a set of warranted conclusions as given above may be too credulous and require a more skeptical alternative. For such matter, we propose the definition of *plausible conclusions* by relying upon a similar notion given in Amgoud (2014).

Definition 5.16 (Set of Plausible Conclusions)**.** Given a knowledge base $\Sigma \subseteq \mathcal{L}^{\kappa}$, the associated GenAT $\tau \in \mathbb{T}$, and a semantics $\mathfrak{s}$; a function $\mathfrak{plausible}_{[\tau,\mathfrak{s}]} : \wp(\mathcal{L}^{\kappa}) \longrightarrow \wp(\mathcal{L}^{\kappa})$ is a **plausible function** *iff* $\mathfrak{plausible}_{[\tau,\mathfrak{s}]}(\Sigma) = \bigcap \mathfrak{warrant}_{[\tau,\mathfrak{s}]}(\Sigma)$. In addition, the **set of plausible consistent conclusions** is identified through the set $\mathfrak{plausible}^{\mathbf{C}}_{[\tau,\mathfrak{s}]}(\Sigma) = \bigcap \mathfrak{warrant}^{\mathbf{C}}_{[\tau,\mathfrak{s}]}(\Sigma)$.

The result sets defined above (Definitions 5.12, 5.15, and 5.16) which are constructed only from consistency $\mathfrak{s}$-extensions will be useful for reasoning over argumentation systems that only tackle consistency clashes, leaving unresolved coherence inconveniences from the underlying knowledge base.

6. Applying the GenAF to $\mathcal{ALC}$ DLs for reasoning over inconsistent ontologies

The use of argumentation techniques to reason over inconsistent ontologies has been gaining attention lately. Some works have been presented exploring different variations of this confluence of areas (see Section 8), and although research on the matter is at its very initial stages, its usability is quite promising. Mostly, for areas of application like medicine and law, where knowledge bases are expected to allow inconsistencies and thus reasoning processes are designed upon non-classical methodologies.

Considering that in Lutz, Sattler, and Wolter (2001) an extension of the $\mathcal{ALC}$ DL was presented and shown to be equivalent to $\mathcal{L}^2$ (see Baader, 1999; Borgida, 1996 for more details), we propose a reification of $\mathcal{L}^{\kappa}$ to the description logic (DL) $\mathcal{ALC}$, and to such end, concepts descriptions in $\mathcal{ALC}$ will be translated into $\mathcal{L}^2$ formulæ. As a result, we introduce the definition of an $\mathcal{ALC}$-GenAF aiming at reasoning over potentially inconsistent/incoherent $\mathcal{ALC}$ ontologies.

We first give a brief overview of $\mathcal{ALC}$ DLs in which we will rely on for the specification of the $\mathcal{ALC}$-GenAF used for reasoning over inconsistent ontologies. To this end, we will refer to Bienvenu's work on $\mathcal{ALC}$ prime implicates as a recommended normal form for named concepts. A π-pANF $\mathcal{ALC}$ ontology can be obtained by transforming $\mathcal{ALC}$ axioms like $C \sqsubseteq D$ into subsumptions between C's and D's prime implicates; thus, we might not only obtain the $\mathcal{ALC}$-GenAF, but also benefit from the well known advantages of prime implicates as a form of knowledge compilation.

6.1. $\mathcal{ALC}$ overview

Before presenting the reification details, we will give a brief overview of the $\mathcal{ALC}$ DLs (for details on $\mathcal{ALC}$ and other DLs, see Baader, Calvanese, McGuinness, Nardi, & Patel-Schneider, 2003). 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 $C, D ::= A \mid \bot \mid \top \mid \neg C \mid C \sqcap D \mid C \sqcup D \mid \forall R.C \mid \exists R.C$ 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 $(\forall R.C)^{\mathcal{I}} = \{a \in \Delta^{\mathcal{I}} \mid \forall b.(a, b) \in R^{\mathcal{I}} \rightarrow b \in C^{\mathcal{I}}\}$; and the full existential quantification as $(\exists R.C)^{\mathcal{I}} = \{a \in \Delta^{\mathcal{I}} \mid \exists b.(a, b) \in R^{\mathcal{I}} \wedge b \in C^{\mathcal{I}}\}$.

An ontology is a pair $\mathcal{O} = \langle \mathrm{T}, \mathrm{A} \rangle$, where T represents the TBox, containing the terminologies (or axioms) of the application domain, and A, the ABox, which contains assertions about named individuals in terms of these terminologies. Regarding the TBox 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 T if $\mathcal{I}$ satisfies all the axioms in T; thus, the TBox T is said to be satisfiable if it admits a model. Also, for the ABox A, $\mathcal{I}$ satisfies $C(a)$ if $a \in C^{\mathcal{I}}$, and $R(a, b)$ if $(a, b) \in R^{\mathcal{I}}$, and $\mathcal{I}$ is said to be a model of the ABox A if every assertion of A is satisfied by $\mathcal{I}$; hence, the ABox A is said to be satisfiable if it admits a model. Finally, regarding the entire ontology $\mathcal{O}$, an interpretation $\mathcal{I}$ is said to be a model of $\mathcal{O}$ if every statement in it 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 *incoherence* presented in Flouris et al. (2006). Thus, 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 named concept C in $\mathcal{O}$, and a concept C is *unsatisfiable iff* for each interpretation $\mathcal{I}$ of $\mathcal{O}$, $C^{\mathcal{I}} = \emptyset$ holds. While incoherence is considered a form 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. Just for simplicity, we shall abuse notation writing $\mathcal{O} = \mathrm{T} \cup \mathrm{A}$ (*e.g.*, $\mathcal{O} = \{C \sqsubseteq D, A(a)\}$) to identify an ontology $\mathcal{O} = \langle \mathrm{T}, \mathrm{A} \rangle$ (*e.g.*, $\mathcal{O} = \langle \{C \sqsubseteq D\}, \{A(a)\} \rangle$).

6.2. Specifying the $\mathcal{ALC}$-GenAF

One of the first inconveniences we face towards the specification of the $\mathcal{ALC}$-GenAF is the construction of a π-pANF $\mathcal{ALC}$ ontology and the question of what we will obtain from that should be answered. Bienvenu's work on *$\mathcal{ALC}$ prime implicate normal form* (Bienvenu, 2008; 2009), has been shown as enjoying several properties which are quite interesting; for instance, $\mathcal{ALC}$ concepts in prime implicate normal form are much better behaved computationally than arbitrary $\mathcal{ALC}$ concepts: whether a concept in prime implicate normal form is satisfiable or tautologous can be tested in constant time, and whether two concepts in prime implicate normal form are equivalent or if one subsumes the other can be tested in quadratic time.

Knowledge compilation requires an initial cost of preprocessing for computing the normal form. The transformation of an $\mathcal{ALC}$ concept to its prime implicate normal form can be achieved through the algorithm introduced in Bienvenu (2008), and was shown to result in at most a doubly-exponential blowup in concept size. However, a structural subsumption algorithm was presented for deciding subsumption between concepts in prime implicate normal form. Such algorithm was shown to be correct, complete, and to make its decisions in linear time in the size of the input, *i.e.*, for a query $C \sqsubseteq D$, it terminates in linear time in $|C||D|$, and hence in quadratic time in $|C| + |D|$. This signifies a considerable improvement wrt. the cost of subsumption with non-normalized $\mathcal{ALC}$ axioms (known to be EXPTIME-complete or PSPACE-complete for unfoldable $\mathcal{ALC}$, which also requires an initial preprocessing cost) and it is supposed to offset the initial cost of the transformation with such computational savings made on later queries.

We will show that the grammars used for prime implicates can match well the π-pANF defined here, and thus, the $\mathcal{ALC}$-GenAF can benefit from the aforementioned prime implicate nice properties.

L: $:= \top \mid \bot \mid A \mid \neg A \mid \forall R.D \mid \exists R.D$

Cl: $:= L \mid Cl \sqcup Cl$

Cb: $:= L \mid Cb \sqcap Cb$

D: $:= \top \mid \bot \mid A \mid \neg A \mid D \sqcap D \mid D \sqcup D \mid \forall R.D \mid \exists R.D$

Given the $\mathcal{ALC}$ prime implicate grammar we can specify a π-pANF $\mathcal{ALC}$ ontology.

Definition 6.1 (. π-pANF $\mathcal{ALC}$Ontology) Given an AL-framework $\pi \in \mathbb{L}$ and an $\mathcal{ALC}$ ontology $\mathcal{O} = \langle T, A \rangle$, we say $\mathcal{O}$ is in π-pANF *iff* $T \subseteq \mathcal{L}_T$ and $A \subseteq \mathcal{L}_a$, where $\mathcal{L}_T ::= Cb \sqsubseteq Cl$ identifies axioms and $\mathcal{L}_a ::= Cl(a) | R(a, b)$, (ground) assertions, with a and b as individual names. The set $\mathcal{L}_T \times \mathcal{L}_a$ identifies the domain of π-pANF $\mathcal{ALC}$ ontologies.

The AL-framework $\pi \in \mathbb{L}$ that we adopt to construct the $\mathcal{ALC}$-GenAF, will be specified by making $\pi = \langle \mathcal{ALC}, \mathcal{L}_{\mathtt{cl}}, \mathcal{L}_{\mathtt{pr}}, \mathcal{L}_{\mathtt{a}} \rangle$, where:
$\mathcal{L}_{\mathtt{pr}} ::= L$

$\mathcal{L}_{\mathtt{cl}} ::= Cl \mid Cl(a) \mid R(a, b)$

$\mathcal{L}_{\mathtt{a}} ::= Cl(a) \mid R(a, b)$

Proposition 6.2. Given the AL-framework $\pi \in \mathbb{L}$ for $\mathcal{ALC}$, the languages $\mathcal{L}_{\mathtt{pr}} ::= L$ for the premises, and $\mathcal{L}_{\mathtt{cl}} ::= Cl \mid Cl(a) \mid R(a, b)$ for the claims, determine a legal argument language $\mathbb{A}_\pi$.

Observe that according to Definition 3.3, Definition 6.1 which specifies the languages for premises and claims, and the grammar for prime implicate $\mathcal{ALC}$ concepts, every axiom/assertion in a π-pANF ontology $\mathcal{O}$ is an argumental atom itself in the GenAF, unless the consistency property in Definition 3.3 is violated. It is important to note that the algorithms for building prime implicates of $\mathcal{ALC}$ concepts presented in Bienvenu (2008) will produce new $\mathcal{ALC}$ concepts conforming the grammar given above and it will contain neither unnecessary atomic concepts or roles, nor redundant or irrelevant subconcepts. This is another of the remarkable advantages of the $\mathcal{ALC}$ prime implicate normal form, making easier for humans to read and understand them; even more interesting for us is that π-pANF $\mathcal{ALC}$ axioms/assertions, as specified by Definition 6.1, will trivially conform the conditions given in Definition 3.3 to build argumental atoms from π-pANF formulæ.

It is important to mention that the detailed use of prime implicates (and/or other type of normal forms) exceeds the scope of the article, and is proposed as an alternative to benefit from the nice properties of knowledge compilation, facing the necessity to transform $\mathcal{ALC}$ ontologies into π-pANF. Moreover, the versatility of the GenAF allows to adapt its argument language to different normal forms proposed for $\mathcal{ALC}$, like disjunctive form (ten Cate, Conradie, Marx, & Venema, 2006), and linkless normal form (Furbach, Günther, & Obermaier, 2009; Furbach & Obermaier, 2007). Implementations of the $\mathcal{ALC}$-GenAF should consider algorithms for the transformation of concepts according to the adopted normal form, in combination with the appropriate grammars to obtain π-pANF $\mathcal{ALC}$ ontologies. In what follows, we will confine ourselves to π-pANF ontologies satisfying the grammars given above –without considering the prime implicates of concepts used in examples– although the reader should keep in mind that other grammars might adapt well for constructing a different, but still appropriate $\mathcal{ALC}$-GenAF.

Some particularities appear for $\mathcal{ALC}$ DLs; for instance, concept equivalences as $C_1 \equiv C_2$, are transformed into two π-pANF axioms $C_1 \sqsubseteq C_2$ and $C_2 \sqsubseteq C_1$. Also, two π-pANF axioms like $\bot \sqsubseteq C$ and $C \sqsubseteq \bot$, will only create two argumental atoms from their contrapositives π-pANF axioms $\neg C \sqsubseteq \top$ and $\top \sqsubseteq \neg C$, respectively, given that argumental atoms cannot accept $\bot$ in any of their components (see the consistency property in Definition 3.3). Finally, any assertion $Cl(a)$ (resp., $R(a, b)$) produces an evidence $\langle \{\}, Cl(a) \rangle$ (resp., $\langle \{\}, R(a, b) \rangle$).

The function "$\mathfrak{panf}_\pi$" (Definition 3.8), is used to translate any $\mathcal{ALC}$ ontology $\mathcal{O}$ into an equivalent $\mathcal{L}_T \times \mathcal{L}_a$ ontology $\mathfrak{panf}_\pi(\mathcal{O})$. A desirable property of a π-pANF $\mathcal{ALC}$ ontology is that each statement in it generates a single argumental atom (actually a pair, if it is the case that its contrapositive falls within the language) in its related GenAF. This statement holds except for unsatisfiable concept inclusions as $A \sqsubseteq \neg A$, which are filtered by the consistency property of Definition 3.3 –giving no related argumental atom in the GenAF.

Given an $\mathcal{ALC}$ ontology $\mathcal{O}$, the associated GenAF $\langle \pi, \mathbf{A} \rangle$ is obtained by means of the theory function $\mathfrak{genaf}(\mathcal{O})$ (see Definition 3.15). Note that cyclic terminologies like $A \equiv B$ will not be part of any structure due to non-circularity in Definition 4.5 (condition 8). A similar situation occurs with axioms like $A \sqsubseteq A$.

Next, we exemplify a reasoning process upon an $\mathcal{ALC}$ ontology with a complex construction of structures where some argumental atoms are reintroduced within a same supporting-chain due to different variables instantiations.

Example 13. Given the $\mathcal{ALC}$ ontology $\mathcal{O} = \{R(a,b), R(b,c), R(c,d), A(a), \neg A(c), \neg A(d), A \sqsubseteq \forall R.A\}$. We will assume a closed under transposition $\mathcal{ALC}$ GenAF $\delta = \langle \pi, \mathbf{A} \rangle$, where $\mathbf{A} = \{\mathcal{A}_1, \mathcal{A}_2, \mathcal{A}_3, \mathcal{A}_4, \mathcal{A}_5, \mathcal{A}_6, \mathcal{A}_7, \mathcal{A}'_7\}$, such that $\mathcal{A}_1 = \langle \{\}, R(a,b) \rangle$, $\mathcal{A}_2 = \langle \{\}, R(b,c) \rangle$, $\mathcal{A}_3 = \langle \{\}, R(c,d) \rangle$, $\mathcal{A}_4 = \langle \{\}, A(a) \rangle$, $\mathcal{A}_5 = \langle \{\}, \neg A(c) \rangle$, $\mathcal{A}_6 = \langle \{\}, \neg A(d) \rangle$, $\mathcal{A}_7 = \langle \{A(x)\}, (\forall R.A)(x) \rangle$, and $\mathcal{A}'_7 = \langle \{(\exists R.\neg A)(x)\}, \neg A(x) \rangle$. The related GenAS $\sigma = \langle \delta, \mathbf{S}_\delta, \mathbf{R}_\delta \rangle$ can be built. For simplicity, we will only consider "some" full arguments which are relevant for this example.

$\mathcal{S}_1 = \langle \{\}, \{\mathcal{A}_4, \mathcal{A}_7, \mathcal{A}_1\}, A(b) \rangle$

$\mathcal{S}_2 = \langle \{\}, \{\mathcal{A}_4, \mathcal{A}_7, \mathcal{A}_1, \mathcal{A}_2\}, A(c) \rangle$

$\mathcal{S}_3 = \langle \{\}, \{\mathcal{A}_4, \mathcal{A}_7, \mathcal{A}_1, \mathcal{A}_2, \mathcal{A}_3\}, A(d) \rangle$

$\mathcal{S}_4 = \langle \{\}, \{\mathcal{A}_3, \mathcal{A}_6, \mathcal{A}'_7\}, \neg A(c) \rangle$

$\mathcal{S}_5 = \langle \{\}, \{\mathcal{A}_3, \mathcal{A}_6, \mathcal{A}'_7, \mathcal{A}_2\}, \neg A(b) \rangle$

$\mathcal{S}_6 = \langle \{\}, \{\mathcal{A}_3, \mathcal{A}_6, \mathcal{A}'_7, \mathcal{A}_2, \mathcal{A}_1\}, \neg A(a) \rangle$

Let us also consider the following primitive arguments: $\mathcal{S}'_1 = \langle \{\}, \{\mathcal{A}_1\}, R(a,b) \rangle$

$\mathcal{S}'_2 = \langle \{\}, \{\mathcal{A}_2\}, R(b,c) \rangle$

$\mathcal{S}'_3 = \langle \{\}, \{\mathcal{A}_3\}, R(c,d) \rangle$

$\mathcal{S}'_4 = \langle \{\}, \{\mathcal{A}_4\}, A(a) \rangle$

$\mathcal{S}'_5 = \langle \{\}, \{\mathcal{A}_5\}, \neg A(c) \rangle$

$\mathcal{S}'_6 = \langle \{\}, \{\mathcal{A}_6\}, \neg A(d) \rangle$

Let us analyze firstly the construction of the mentioned structures: the set $\{\mathcal{A}_7, \mathcal{A}_1\}$ is the claim-coalition for $A(b)$ in $\mathcal{S}_1$, $\{\mathcal{A}_7, \mathcal{A}_2\}$ is the claim-coalition for $A(c)$ in $\mathcal{S}_2$, and $\{\mathcal{A}_7, \mathcal{A}_3\}$ is the claim-coalition for $A(d)$ in $\mathcal{S}_3$. Observe also that $\mathcal{S}_1$ is a supporting structure in $\mathcal{S}_2$, and $\mathcal{S}_2$ is a supporting structure in $\mathcal{S}_3$; and also that both $\mathcal{S}'_3$ and $\mathcal{S}'_6$ are supporting structures of the claim-coalition's premise $(\exists R.\neg A)(x)$ in $\mathcal{S}_4$, $\mathcal{S}'_2$ and $\mathcal{S}_4$ are supporting structures of the claim-coalition's premise $(\exists R.\neg A)(x)$ in $\mathcal{S}_5$, and $\mathcal{S}'_1$ and $\mathcal{S}_5$ are supporting structures of the claim-coalition's premise $(\exists R.\neg A)(x)$ in $\mathcal{S}_6$ (refer to Fig. 6).

Afterwards, for avoiding symmetric rebuttals we will rely on an atom comparison criterion '$\succeq$' such that the following rebuttals

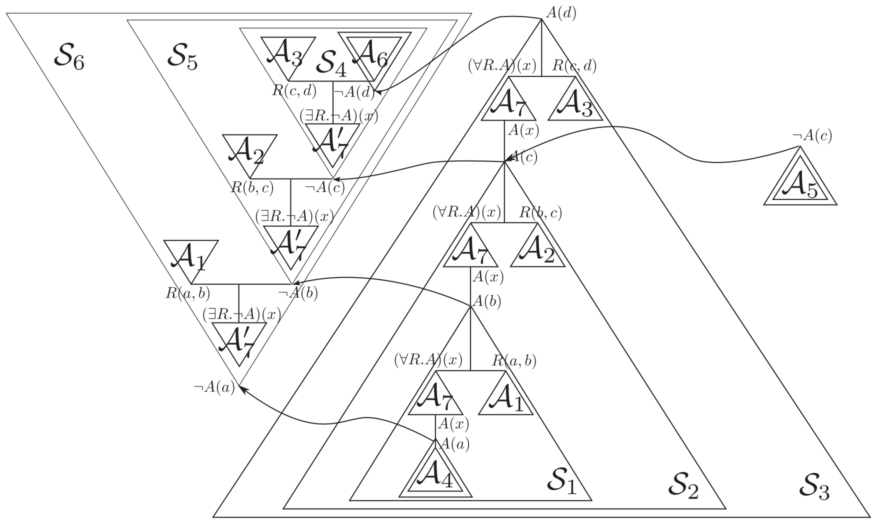


Fig. 6. Graph of rebuttals from Example 13. Multiple occurrences of an argumental atom within a structure are due to the particular substitution within each corresponding supporting structure.

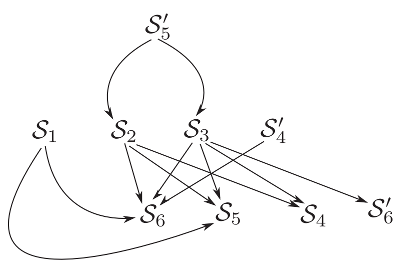


Fig. 7. Complete graph of attacks from Example 13.

appear: $\mathcal{S}_1\mathbf{C}^b_\delta\mathcal{S}_5$, $\mathcal{S}_2\mathbf{C}^b_\delta\mathcal{S}_4$, $\mathcal{S}_3\mathbf{C}^b_\delta\mathcal{S}'_6$, $\mathcal{S}'_4\mathbf{C}^b_\delta\mathcal{S}_6$, and $\mathcal{S}'_5\mathbf{C}^b_\delta\mathcal{S}_2$. Fig. 6 illustrates both he construction of the referred structures and rebuttals.

In addition, Fig. 7 illustrates the complete graph of attacks considering both rebuttals and undercuts. The following undercuts also appear: $\mathcal{S}_1\mathbf{C}_\delta\mathcal{S}_6$, $\mathcal{S}_2\mathbf{C}_\delta\mathcal{S}_6$, $\mathcal{S}_2\mathbf{C}_\delta\mathcal{S}_5$, $\mathcal{S}_3\mathbf{C}_\delta\mathcal{S}_6$, $\mathcal{S}_3\mathbf{C}_\delta\mathcal{S}_5$, $\mathcal{S}_3\mathbf{C}_\delta\mathcal{S}_4$, and $\mathcal{S}'_5\mathbf{C}_\delta\mathcal{S}_3$.

Since we are not considering potential arguments, direct undercuts will not appear here. Thus, we are going to analyze only consistency $\mathfrak{s}$-extensions, where $\mathfrak{s}$ implements the complete semantics. The set $\{\mathcal{S}_1, \mathcal{S}'_4, \mathcal{S}_4, \mathcal{S}'_6, \mathcal{S}'_5\}$ is part of a complete extension in $\mathfrak{ext}^{\mathbf{C}}_{\mathfrak{s}}(\sigma)$, hence $\mathcal{O}\ \approx\!\!|^{\mathbf{C}}_{[\tau,\mathfrak{s}]} A(b)$ holds and thus we say that $A(b)$ is consistently warranted by the `GenAF` reasoner.

7. Equivalence with logic-based argumentation systems

In this section we study `GenAF`s by contrasting them with standard logic-based argumentation. Therefore, we firstly provide simple definitions for logic-based arguments and logic-based argumentation systems, and afterwards we define a translation procedure for a subsystem of a `GenAS` into a standard logic-based argumentation system. This will allow to study argumentation postulates for standard logic-based argumentation systems and afterwards, it brings the possibility of studying such postulates for a `GenAS` under certain specific conditions.

Definition 7.1 (Logic-based argument)**.** Given the logic $\mathcal{L}^\kappa$, a structure $\mathcal{B} = \langle\Psi, \alpha\rangle$, where $\Psi \subseteq \mathcal{L}^\kappa$ (body) and $\alpha \in \mathcal{L}^\kappa$ (claim), is a **logic-based argument** *iff* $\Psi \vDash \alpha$, $\Psi \not\vDash \bot$, and there is no $\Psi' \subset \Psi$ such that $\Psi' \vDash \alpha$. The set $\mathbb{B}$ identifies the domain $\wp(\mathcal{L}^\kappa) \times \mathcal{L}^\kappa$ of logic-based arguments.

We will overload the usage of the functions $\mathfrak{bd}$ and $\mathfrak{cl}$ to respectively identify the body and claim of logic-based arguments. Similarly, the substructure operator $\trianglelefteq$ will be used for identifying logic-based subarguments such that for any pair $\mathcal{B}, \mathcal{B}' \in \mathbb{B}$, $\mathcal{B} \trianglelefteq \mathcal{B}'$ *iff* $\mathfrak{bd}(\mathcal{B}) \subseteq \mathfrak{bd}(\mathcal{B}')$. In addition, we will overload the base function for being applied over logic-based arguments such that given a set of arguments $\Theta \subseteq \mathbb{B}$, $\mathfrak{base}(\Theta) = \{\vartheta \in \mathfrak{bd}(\mathcal{B}) | \mathcal{B} \in \Theta\}$. Note that given a logic-based argument $\mathcal{B} \in \mathbb{B}$ it holds $\mathfrak{bd}(\mathcal{B}) = \mathfrak{base}(\{\mathcal{B}\})$.

Definition 7.2 (Logic-based AS)**.** Given a knowledge base $\Sigma \subseteq \mathcal{L}^\kappa$, $AS(\Sigma)$ is a **logic-based argumentation system** (or just, **logic-based AS**) from Σ *iff* $AS(\Sigma) = \langle Args(\Sigma), \mathbf{R}\rangle$ where $Args(\Sigma)$ is the maximal set of logic-based arguments $\mathcal{B} \in \mathbb{B}$ constructed from *KB* *i.e.*, $Args(\Sigma) = \{\mathcal{B} \in \mathbb{B} | \mathfrak{bd}(\mathcal{B}) \subseteq \Sigma\}$) and $\mathbf{R} \subseteq Args(\Sigma) \times Args(\Sigma)$ is the attack relation. The domain of logic-based ASs is identified through the set $\mathbb{S}$.

Definition 7.3 (Set of extensions of a logic-based AS)**.** Given a knowledge base $\Sigma \subseteq \mathcal{L}^\kappa$, its associated logic-based AS $AS(\Sigma) = \langle Args(\Sigma), \mathbf{R}\rangle$, and a standard semantics $\mathfrak{s}$; $Ext_{\mathfrak{s}}(AS(\Sigma))$ is the **set of $\mathfrak{s}$-extensions** of $AS(\Sigma)$ *iff* for every $\mathcal{E} \in Ext_{\mathfrak{s}}(AS(\Sigma))$ it holds $\mathcal{E} \subseteq Args(\Sigma)$ and $\mathcal{E}$ is an $\mathfrak{s}$-extension.

Now we provide the following functions for the construction of a logic-based AS from a `GenAS`.

Definition 7.4 (Logic-based argument translation)**.** Given a `GenAS` $\langle\delta, \mathbf{S}_\delta, \mathbf{R}_\delta\rangle \in \mathbb{G}$, a function $\mathfrak{lbarg} : \mathbf{S}_\delta \longrightarrow \mathbb{B}$ is a **logic-based argument function** *iff* $\mathfrak{lbarg}(\mathcal{S}) = \langle \mathfrak{base}(\{\mathcal{S}\}), (\bigwedge \mathfrak{pr}(\mathcal{S}) \rightarrow \mathfrak{cl}(\mathcal{S}))\rangle$, for any argumental structure $\mathcal{S} \in \mathbf{S}_\delta$.

The following example shows the intuitions behind the logic-based argument translation function.

Example 14. Suppose we have an argumental structure $\mathcal{S} = \langle\{p_1, p_2\}, \{\mathcal{A}_1\}, q\rangle$, where $\mathcal{A}_1 = \langle\{p_1, p_2\}, q\rangle$ is originated through a formula $p_1 \wedge p_2 \rightarrow q$. Thus, it is natural to expect an equivalent logic-based argument $\mathfrak{lbarg}(\mathcal{S}) = \mathcal{B} = \langle\{p_1 \wedge p_2 \rightarrow q\}, p_1 \wedge p_2 \rightarrow q\rangle$. Now suppose we have the atom $\mathcal{A}_2 = \langle\{\}, p_1\rangle$. The structure $\mathcal{S}' = \langle\{p_2\}, \{\mathcal{A}_1, \mathcal{A}_2\}, q\rangle$ would be constructed. In such a case, the translation would end up as $\mathfrak{lbarg}(\mathcal{S}') = \mathcal{B}' = \langle\{p_1, (p_1 \wedge p_2 \rightarrow q)\}, p_2 \rightarrow q\rangle$. Observe that if we construct the `GenAF`-argument $\mathcal{S}'' = \langle\{\}, \{\mathcal{A}_1, \mathcal{A}_2, \mathcal{A}_3\}, q\rangle$, where $\mathcal{A}_3 = \langle\{\}, p_2\rangle$, the translated logic-based argument would end up as $\mathfrak{lbarg}(\mathcal{S}'') = \mathcal{B}'' = \langle\{p_1, p_2, (p_1 \wedge p_2 \rightarrow q)\}, q\rangle$.

We show that the logic-based argument function, as defined above, effectively translates every argumental structure into a logic-based argument.

Proposition 7.5. *Given a* GenAS $\langle\delta, \mathbf{S}_\delta, \mathbf{R}_\delta\rangle \in \mathbb{G}$, *if* $\mathcal{S} \in \mathbf{S}_\delta$ *then* $\mathfrak{lbarg}(\mathcal{S}) \in \mathbb{B}$ *is a logic-based argument.*

Remark 7.6. Observe that given an argumental structure $\mathcal{S} \in \mathbf{S}_\delta$, if $\mathfrak{lbarg}(\mathcal{S}) = \mathcal{B}$ then $\mathfrak{base}(\{\mathcal{S}\}) = \mathfrak{bd}(\mathcal{B})$.

Definition 7.7 (Logic-based AS translation)**.** Given a GenAS $\sigma = \langle\delta, \mathbf{S}_\delta, \mathbf{R}_\delta\rangle \in \mathbb{G}$, a function $\mathfrak{lbas} : \mathbb{G} \longrightarrow \mathbb{S}$ is an **logic-based AS function** *iff* $\mathfrak{lbas}(\sigma) = \langle\mathbf{B}, \mathbf{T}\rangle$ where:

$\mathbf{B} = \{\mathfrak{lbarg}(\mathcal{S}) |$for every argumental structure $\mathcal{S} \in \mathbf{S}_\delta\}$, and

$\mathbf{T} = \{(\mathcal{B}_1, \mathcal{B}_2) |$for every pair $(\mathcal{S}_1, \mathcal{S}_2) \in \mathbf{C}_\delta$, where $\mathcal{B}_1 = \mathfrak{lbarg}(\mathcal{S}_1)$ and $\mathcal{B}_2 = \mathfrak{lbarg}(\mathcal{S}_2)\}$.

The following lemma proposes the conditions under which a GenAS which is translated into a logic-based AS contains all the arguments constructible from its underlying knowledge base. In an intuitive manner, ensuring that $\mathcal{L}^\kappa$ is a π-pANF logic guarantees that all the formulæ in the KB (including its inferences) corresponds to a π-pANF form. Besides, the requirement of closure under transposition ensures that *modus tollens* is modeled for the construction of arguments. Finally, the set of constraints is reduced to its minimal form in order to ensure that all possible formulæ will be considered. Tautologies cannot be plausible conclusions of any logic-based argumentation system. The inconveniences related with such a drawback are treated in Amgoud (2014): tautologies may depreciate the utility of the argumentation postulate controlling the consequences of the system. Note that we will not allow tautologies in the underlying knowledge base for ensuring the well behavior of any logic-based AS $AS(\Sigma)$, however, for a GenAF this would not be necessary given that it has the ability to restrict the construction of tautologic argumental atoms by modeling the non-circularity constraint.

Lemma 7.8. *Given a* GenAT $\tau = \langle\Sigma, \pi, \mathfrak{panf}_\pi, \Omega, \mathfrak{atom}_{[\pi,\Omega]}\rangle \in \mathbb{T}$, *the* GenAS $\mathfrak{genas}(\tau) = \langle\delta, \mathbf{S}_\delta, \mathbf{R}_\delta\rangle \in \mathbb{G}$, *and the translated logic-based AS* $\mathfrak{lbas}(\mathfrak{genas}(\tau)) = \langle\mathbf{B}, \mathbf{T}\rangle$*; if* $\mathcal{L}^\kappa$ *is a* π-pANF *logic,* Σ *has no tautologies, the* GenAF δ *is closed under transposition, and* $\Omega \subseteq \{\omega^1_\pi, \omega^2_\pi\}$ *where* ω^1_π *and* ω^2_π *are atom constraint functions modeling* $\mathbf{AC}_1$ *and* $\mathbf{AC}_2$, *respectively, then* $\mathbf{B} = Args(\Sigma)$.

The GenAF entails the same conclusions that its translated logic-based version.

Lemma 7.9. *Given a* GenAT $\tau = \langle\Sigma, \pi, \mathfrak{panf}_\pi, \Omega, \mathfrak{atom}_{[\pi,\Omega]}\rangle \in \mathbb{T}$, *the* GenAS $\mathfrak{genas}(\tau) = \langle\delta, \mathbf{S}_\delta, \mathbf{R}_\delta\rangle \in \mathbb{G}$, *and the translated logic-based AS* $\mathfrak{lbas}(\mathfrak{genas}(\tau)) = \langle\mathbf{B}, \mathbf{T}\rangle$*; if* $\mathcal{L}^\kappa$ *is a* π-pANF *logic,* Σ *has no tautologies, the* GenAF δ *is closed under transposition, and* $\Omega \subseteq \{\omega^1_\pi, \omega^2_\pi\}$ *where* ω^1_π *and* ω^2_π *are atom constraint functions modeling* $\mathbf{AC}_1$ *and* $\mathbf{AC}_2$, *respectively, then* $\mathfrak{warrant}^{\mathbf{C}}_{[\tau,\mathfrak{s}]}(\Sigma) = \{\mathfrak{clset}(\mathcal{E}) | \mathcal{E} \in Ext_{\mathfrak{s}}(\mathfrak{lbas}(\mathfrak{genas}(\tau)))\}$.

Build from the previous lemmata, the following corollary presents the conditions under which a GenAS is equivalent to a logic-based argumentation system, that means that both argumentation systems shares: the same arguments (arguments from one AS can be translated into the other and viceversa), the same attack relations (based on consistency), and the same consequences.

Corollary 7.10. *Given a* GenAT $\tau = \langle\Sigma, \pi, \mathfrak{panf}_\pi, \Omega, \mathfrak{atom}_{[\pi,\Omega]}\rangle \in \mathbb{T}$, *the* GenAS $\mathfrak{genas}(\tau) = \langle\delta, \mathbf{S}_\delta, \mathbf{R}_\delta\rangle \in \mathbb{G}$, *and the translated logic-based AS* $\mathfrak{lbas}(\mathfrak{genas}(\tau)) = \langle\mathbf{B}, \mathbf{T}\rangle$*; if* $\mathcal{L}^\kappa$ *is a* π-pANF *logic,* Σ *has no tautologies, the* GenAF δ *is closed under transposition, and* $\Omega \subseteq \{\omega^1_\pi, \omega^2_\pi\}$ *where* ω^1_π *and* ω^2_π *are atom constraint functions modeling* $\mathbf{AC}_1$ *and* $\mathbf{AC}_2$, *respectively, then* $\mathfrak{lbas}(\mathfrak{genas}(\tau))$ *is equivalent to* $AS(\Sigma)$.

However, nothing is still said about the attack relation. In what follows we will concentrate on the properties of the attack relation modeled through GenAFs. For such purpose, we will refer to the work by Amgoud (Amgoud, 2014) where a set of properties for logic-based argumentation systems has been proposed with the objective of characterizing the conflict relation between pairs of logic-based arguments. Such a characterization allow the study of the *argumentation postulates* that we will see later in Definition 7.16. Before formalizing the mentioned conflict characterization, we will say that a set $X \subseteq \Sigma$ is a *minimal source of inconsistency* of a knowledge base $\Sigma \subseteq \mathcal{L}^\kappa$ *iff* $X \vDash \bot$ and there is no $X' \subset X$ such that $X' \vDash \bot$ holds. In addition, we will refer to the set $\mathtt{mi}(\Sigma) \subseteq \wp(\mathcal{L}^\kappa)$ as the *set of minimal sources of inconsistencies of a knowledge base* Σ *iff* $\mathtt{mi}(\Sigma)$ contains every minimal source of inconsistency of Σ.

Definition 7.11 (Conflict characterization (Amgoud, 2014))**.** Given $AS(\Sigma) = \langle Args(\Sigma), \mathbf{R}\rangle$, for any $\mathcal{B}_1, \mathcal{B}_2, \mathcal{B}_3 \in Args(\Sigma)$, the attack relation $\mathbf{R}$ may verify any of the following properties:

(R_1) if $\mathfrak{bd}(\mathcal{B}_1) \subseteq \mathfrak{bd}(\mathcal{B}_2)$ and $(\mathcal{B}_1, \mathcal{B}_3) \in \mathbf{R}$ then $(\mathcal{B}_2, \mathcal{B}_3) \in \mathbf{R}$.
(R_2) if $\mathfrak{bd}(\mathcal{B}_1) \subseteq \mathfrak{bd}(\mathcal{B}_2)$ and $(\mathcal{B}_3, \mathcal{B}_1) \in \mathbf{R}$ then $(\mathcal{B}_3, \mathcal{B}_2) \in \mathbf{R}$.
(conflict-dependent) if $(\mathcal{B}_1, \mathcal{B}_2) \in \mathbf{R}$ then $\mathfrak{bd}(\mathcal{B}_1) \cup \mathfrak{bd}(\mathcal{B}_2) \models \bot$.
(conflict-sensitive) if $\mathfrak{bd}(\mathcal{B}_1) \cup \mathfrak{bd}(\mathcal{B}_2) \models \bot$ then $(\mathcal{B}_1, \mathcal{B}_2) \in \mathbf{R}$ or $(\mathcal{B}_2, \mathcal{B}_1) \in \mathbf{R}$.
(conflict-exhaustive) if $X \in \mathtt{mi}(\Sigma)$ such that $|X| > 1$ then there is $X_1, X_2 \subset X$ such that $X = X_1 \cup X_2$ and $\mathcal{B}_1, \mathcal{B}_2 \in Args(\Sigma)$ such that $\mathfrak{bd}(\mathcal{B}_1) = X_1$ and $\mathfrak{bd}(\mathcal{B}_2) = X_2$, and either $(\mathcal{B}_1, \mathcal{B}_2) \in \mathbf{R}$ or $(\mathcal{B}_2, \mathcal{B}_1) \in \mathbf{R}$.

It was shown in Amgoud (2014) (see Proposition 7.18) that an attack relation simultaneously verifying R_1 and R_2 (see item 1), or conflict-dependent and conflict-sensitive (see item 2), allow the verification of the argumentation postulate of (closure under sub-arguments) (see Definition 7.16). However, the specification of the GenAF's conflict relation (see Definition 4.8) does not allow the verification of properties R_1 nor conflict-sensitive. There could be alternatives for redefining the conflict relation in order to satisfy such properties. Nevertheless, what would this mean? For instance, consider Example 11, Fig. 5. Let us suppose for such a case that we have inversed the attack between $\mathcal{S}_8$ and $\mathcal{S}_4$. Hence, assuming that $(\mathcal{S}_8, \mathcal{S}_4) \in \mathbf{R}_\delta$ holds, it seems difficult to justify the existence of an attack relation from $\mathcal{S}_6$ to $\mathcal{S}_4$ (considering that $\mathcal{S}_8 \trianglelefteq \mathcal{S}_6$) since an attack relation is usually directed from the claim of an argument to the defeated argument. This is made clearer if we consider that the claim of $\mathcal{S}_6$ is $p_3(a)$, and that there is nothing in $\mathcal{S}_4$ which could be related to p_3. We will see later in this section, that the argumentation postulates that are carried through the verification of such two properties can be achieved in a more intuitive form. Next, we show that the GenAF's attack relation verifies the properties of conflict-dependent and R_2.

Proposition 7.12. *Given a* GenAT $\tau = \langle\Sigma, \pi, \mathfrak{panf}_\pi, \Omega, \mathfrak{atom}_{[\pi,\Omega]}\rangle \in \mathbb{T}$, *the* GenAS $\mathfrak{genas}(\tau) = \langle\delta, \mathbf{S}_\delta, \mathbf{R}_\delta\rangle \in \mathbb{G}$, *and the translated logic-based AS* $\mathfrak{lbas}(\mathfrak{genas}(\tau)) = \langle Args(\Sigma), \mathbf{R}\rangle$, *the attack relation* $\mathbf{R}$ *verifies* R_2.

Proposition 7.13. *Given a* GenAT $\tau = \langle\Sigma, \pi, \mathfrak{panf}_\pi, \Omega, \mathfrak{atom}_{[\pi,\Omega]}\rangle \in \mathbb{T}$, *the* GenAS $\mathfrak{genas}(\tau) = \langle\delta, \mathbf{S}_\delta, \mathbf{R}_\delta\rangle \in \mathbb{G}$, *and the translated logic-based AS* $\mathfrak{lbas}(\mathfrak{genas}(\tau)) = \langle Args(\Sigma), \mathbf{R}\rangle$, *the attack relation* $\mathbf{R}$ *is conflict-dependent.*

For showing conflict-exhaustive attack sets we need to evaluate certain circumstances in which the inconsistency from the underlying knowledge base cannot be represented through an attack between two arguments. This may appear for certain logics in which it is not possible to represent the complementary value of every formula. This is also the main inconvenience by which the converse of Corollaries 5.5 and 5.7 do not hold. Next, we formalize the notion of parallel-complementary logics and afterwards, we illustrate the situation through Example 15.

Definition 7.14 (Parallel-complementary logics)**.** A logic $\mathcal{L}$ is **parallel-complementary** *iff* for any formula $\vartheta \in \mathcal{L}$ there is a complementary formula $\vartheta' \in \mathcal{L}$ such that $\{\vartheta, \vartheta'\} \models \bot$.

Example 15. Let $\mathcal{L}_{c1}$ be a logic such that for any formula $\vartheta \in \mathcal{L}_{c1}$, ϑ can be a literal, *i.e.*, an atom or its negation, or a clause, *i.e.*, a disjunction of literals. For instance, $\Phi = \{\neg p, \neg q, p \vee q\}$ is a set of formulæ in $\mathcal{L}_{c1}$. Note that $\mathcal{L}_{c1}$ is not parallel-complementary given that a formula like $p \vee q$ has no parallel-complementary formula. That is, there is no formula $\vartheta \in \mathcal{L}_{c1}$ such that $\{(p \vee q), \vartheta\} \models \bot$ can be verified. Let us assume a logic-based AS $AS(\Sigma) = \langle Args(\Sigma), \mathbf{R} \rangle$, where $Args(\Sigma) = \{\mathcal{B}_1, \mathcal{B}_2, \mathcal{B}_3\}$ such that $\mathfrak{cl}(\mathcal{B}_1) = p \vee q$, $\mathfrak{cl}(\mathcal{B}_2) = \neg p$, and $\mathfrak{cl}(\mathcal{B}_3) = \neg q$. Observe that argument $\mathcal{B}_1$ cannot be attacked by any other argument, even more, note that $\mathbf{R}$ will be empty. Note however that Φ is a minimal set verifying $\Phi \models \bot$.

Now we have a condition for modeling a GenAF, ensuring a conflict-exhaustive attack relation for it.

Proposition 7.15. *Given a* GenAT $\tau = \langle \Sigma, \pi, \mathfrak{panf}_\pi, \Omega, \mathfrak{atom}_{[\pi,\Omega]} \rangle \in \mathbb{T}$, *the* GenAS $\mathfrak{genas}(\tau) = \langle \delta, \mathbf{S}_\delta, \mathbf{R}_\delta \rangle \in \mathbb{G}$, *and the translated logic-based AS* $\mathfrak{lbas}(\mathfrak{genas}(\tau)) = \langle Args(\Sigma), \mathbf{R} \rangle$, *if* $\mathcal{L}_{c1}$ *is parallel-complementary then* $\mathbf{R}$ *is conflict-exhaustive.*

The following postulates for a logic-based argumentation system were proposed by Amgoud in Amgoud (2014). We firstly study them and their implications according to the notation of the present article, and analyze afterwards their relation with regards to a GenAS translated to a standard logic-based AS, and the properties it should satisfy in order to guarantee each postulate.

Definition 7.16 (Postulates for a logic-based AS (Amgoud, 2014))**.** Given $AS(\Sigma) = \langle Args(\Sigma), \mathbf{R} \rangle$:

(closure under consequence) for all $\mathcal{E} \in Ext_\mathfrak{s}(AS(\Sigma))$, if $\mathfrak{clset}(\mathcal{E}) \models \vartheta$ then $\vartheta \in \mathfrak{clset}(\mathcal{E})$.
(closure under subarguments) for all $\mathcal{E} \in Ext_\mathfrak{s}(AS(\Sigma))$, if $\mathcal{B} \in \mathcal{E}$ then $\mathcal{B}' \in \mathcal{E}$, for all $\mathcal{B}' \trianglelefteq \mathcal{B}$.
(consistency) for all $\mathcal{E} \in Ext_\mathfrak{s}(AS(\Sigma))$, $\mathfrak{clset}(\mathcal{E}) \not\models \bot$.
(exhaustiveness) for all $\mathcal{E} \in Ext_\mathfrak{s}(AS(\Sigma))$ and all $\mathcal{B} \in Args(\Sigma)$, if $\mathfrak{bd}(\mathcal{B}) \cup \mathfrak{cl}(\mathcal{B}) \subseteq \mathfrak{clset}(\mathcal{E})$ then $\mathcal{B} \in \mathcal{E}$.
(free precedence) for all $\mathcal{E} \in Ext_\mathfrak{s}(AS(\Sigma))$, $Args(\Sigma \setminus \bigcup \mathfrak{mi}(\Sigma)) \subseteq \mathcal{E}$.

The following properties shown in Propositions 7.17, 7.18, and 7.19 are due to Amgoud in Amgoud (2014).

Proposition 7.17 ((Amgoud, 2014))**.** *Given* $AS(\Sigma) = \langle Args(\Sigma), \mathbf{R} \rangle$, *if* $\mathbf{R}$ *is conflict-dependent and* $Ext_\mathfrak{s}(AS(\Sigma)) \neq \emptyset$ *then* $AS(\Sigma)$ *satisfies the postulate of* (free precedence) *for any semantics* $\mathfrak{s}$.

Proposition 7.18 ((Amgoud, 2014))**.** *Given* $AS(\Sigma) = \langle Args(\Sigma), \mathbf{R} \rangle$, *if any of the following is satisfied then* $AS(\Sigma)$ *satisfies the postulate of* (closure under subarguments) *for any semantics* $\mathfrak{s}$*:*

1. *An attack relation* $\mathbf{R}$ *satisfying* R_1 *and* R_2,
2. *An attack relation* $\mathbf{R}$ *which is shown to be conflict-dependent and sensitive,*
3. *An AS* $AS(\Sigma)$ *satisfying* $Ext_\mathfrak{s}(AS(\Sigma)) \neq \emptyset$ *and for all* $\mathcal{E} \in Ext_\mathfrak{s}(AS(\Sigma))$, *it holds* $\mathcal{E} = Args(\mathfrak{base}(\mathcal{E}))$,

Proposition 7.19 ((Amgoud, 2014))**.** *Given* $AS(\Sigma) = \langle Args(\Sigma), \mathbf{R} \rangle$, *if* $\mathbf{R}$ *is conflict-exhaustive and* $\mathcal{E} = Args(\mathfrak{base}(\mathcal{E}))$ *then* $AS(\Sigma)$ *satisfies the postulates of* (closure under consequence) *and* (consistency) *for any semantics* $\mathfrak{s}$.

Proposition 7.17 is perfectly applied to GenAFs given that we have shown it to be conflict-dependent in Proposition 7.13. Thus, GenAFs guarantee the postulate of (free precedence) for any semantics excepting the stable whenever it triggers the empty set. Regarding the postulate of (closure under subarguments), Proposition 7.18 shows it through any of its three alternative items. As we have seen before, GenAFs do not satisfy the attack properties of R_1 and conflict-sensitive, which makes impossible to verify the mentioned postulate through items 1 and 2. Thus, the alternative seems to be the verification of item 3. Moreover, verifying such item would also allow guaranteeing the postulates of (closure under consequence) and (consistency), according to Proposition 7.19. However, the following example shows that verifying $\mathcal{E} = Args(\mathfrak{base}(\mathcal{E}))$ may be not so trivial, and moreover, it will be clear that this is actually a main inconvenience that relies upon any of the standard semantics for reasoning over logic-based argumentation systems.

Example 16. Assume a KB $\Sigma = \{p, q, \neg p \vee \neg q\}$ where $\mathcal{L}^\kappa$ is a π-pANF logic, the resulting set of logic-based arguments will be $Args(\Sigma) = \{\mathcal{B}_1, \mathcal{B}_2, \mathcal{B}_3, \mathcal{B}_4, \mathcal{B}_5, \mathcal{B}_6\}$, where $\mathcal{B}_1 = \langle \{p\}, p \rangle$, $\mathcal{B}_2 = \langle \{q\}, q \rangle$, and $\mathcal{B}_3 = \langle \{\neg p \vee \neg q\}, \neg p \vee \neg q \rangle$ are the primitive arguments and $\mathcal{B}_4 = \langle \{p, q\}, p \wedge q \rangle$, $\mathcal{B}_5 = \langle \{p, \neg p \vee \neg q\}, p \wedge (\neg p \vee \neg q) \rangle$, and $\mathcal{B}_6 = \langle \{q, \neg p \vee \neg q\}, q \wedge (\neg p \vee \neg q) \rangle$ are constructed from combining the primitive ones. Assume also, the following attacks: $\mathbf{R} = \{(\mathcal{B}_1, \mathcal{B}_6), (\mathcal{B}_2, \mathcal{B}_5), (\mathcal{B}_3, \mathcal{B}_4), (\mathcal{B}_4, \mathcal{B}_5), (\mathcal{B}_4, \mathcal{B}_6), (\mathcal{B}_5, \mathcal{B}_4), (\mathcal{B}_6, \mathcal{B}_4)\}$. Observe that the set $\Theta = \{\mathcal{B}_1, \mathcal{B}_2, \mathcal{B}_3\}$ is admissible although $\mathfrak{base}(\Theta) \models \bot$.

The problem presented in Example 16 relies on the construction of logic-based AFs from arbitrary sets of arguments. It is necessary to build all possible arguments, including sub and super arguments, in order to ensure that the resulting AF will deliver rational responses through an argumentation semantics. We say that a set of arguments is closed whenever it contains all the sub- and super-arguments that can be constructed from its arguments. This ensures an exhaustive construction of arguments from an initial base of arguments. We provide such implementation through an *argumentation closure operator* $\mathbb{C}$.

Definition 7.20 (Argumentation closure)**.** A set $\Theta \subseteq Args(\Sigma)$ is **closed** *iff* it holds ($\mathcal{B} \in \Theta$ *iff* $\mathcal{B}' \in \Theta$, for all $\mathcal{B}' \trianglelefteq \mathcal{B}$).

The following proposition shows that the closure of a set Θ of arguments triggers the complete set of arguments that can be constructed using the formulae involved in arguments contained in Θ.

Proposition 7.21. *Given a set* $\Theta \subseteq Args(\Sigma)$, Θ *is closed iff* $\Theta = Args(\mathfrak{base}(\Theta))$.

As been shown in Example 16, the admissibility condition of a set of arguments violates its own spirit when applied over logic-based argumentation systems, *i.e.*, it does not follow its original intention: to recognize a set of arguments which is conflict-free and self acceptable, where conflict-free should be understood as a set of arguments whose bodies do not entail inconsistency. The reason of such a drawback is probably that the notion of admissibility was originally formulated for abstract argumentation. Based on the notion of argumentation closure, we reformulate the abstract notion of admissibility into logic-based admissibility by additionally requiring the argumentation closure condition.

Definition 7.22 (Logic-based admissibility)**.** Given a logic-based AS $\langle Args(\Sigma), \mathbf{R} \rangle$, for any set of arguments $\Theta \subseteq Args(\Sigma)$ we say that Θ is **(logic-based) admissible** *iff* Θ is closed, conflict-free, and defends all its members. For simplicity, when no confusion arise, we get rid of the prefixal term “logic-based”.

Taking into consideration this new form of admissibility for logic-based arguments, we reformulate the usual notion of standard semantics. We will say that a semantics $\mathfrak{s}$ is a *logic-based semantics iff* $\mathfrak{s}$ responds to some of the standard argumentation semantics but replacing the reference to standard admissibility by logic-based admissibility. Thus, for instance, $\mathcal{E} \in Ext_{\mathfrak{s}}(AS(\Sigma))$ is a **complete (logic-based) extension** *iff* it is (logic-based) admissible and it holds $\mathcal{E} = \mathcal{F}(\mathcal{E})$, where $\mathcal{F}$ is the characteristic function. Clearly, the set $Ext_{\mathfrak{s}}(AS(\Sigma))$ is the set of all the logic-based $\mathfrak{s}$-extensions of $AS(\Sigma)$ *iff* $\mathfrak{s}$ refers to some logic-based semantics.

Example 17. (Continues from Example 16). Under the new definition of admissibility, we have that Θ cannot be admissible since it is not closed. The following admissible sets appear: $\{\mathcal{B}_1\}$, $\{\mathcal{B}_2\}$, and $\{\mathcal{B}_3\}$. Note that the sets $\{\mathcal{B}_1, \mathcal{B}_2, \mathcal{B}_4\}$, $\{\mathcal{B}_1, \mathcal{B}_3, \mathcal{B}_5\}$, and $\{\mathcal{B}_2, \mathcal{B}_3, \mathcal{B}_6\}$, are not admissible given that although they are closed and conflict-free, none of them defends all its members.

In Examples 16 and 17, argument $\mathcal{B}_3$ attacks $\mathcal{B}_4$, however, note that although $\mathcal{B}_4$ is constructed by only referring to arguments $\mathcal{B}_1$ and $\mathcal{B}_2$, the set $\{\mathcal{B}_1, \mathcal{B}_2, \mathcal{B}_3\}$ is an admissible extension (under classical admissibility) and also closed under subarguments. This depreciates the utility of the postulate of closure under subarguments, which does not accomplish with its objective proposed in Amgoud (2014). The inconvenience seems to be its inability to make reference to superarguments. For that reason, we propose to harness such postulate –making it more restrictive– by relying upon argumentation closure as follows.

Definition 7.23 (Closure Postulate for a logic-based AS)**.** Given $AS(\Sigma) = \langle Args(\Sigma), \mathbf{R}\rangle$:

(closure) for all $\mathcal{E} \in Ext_{\mathfrak{s}}(AS(\Sigma))$, $\mathcal{B} \in \mathcal{E}$ *iff* $\mathcal{B}' \in \mathcal{E}$, for all $\mathcal{B}' \trianglelefteq \mathcal{B}$.

Remark 7.24. Given $AS(\Sigma) = \langle Args(\Sigma), \mathbf{R}\rangle$, if $AS(\Sigma)$ guarantees the postulate of (closure) then it also guarantees the postulate of (closure under subarguments).

The following lemmata show the implications of relying upon any logic-based semantics regarding the argumentation postulates.

Lemma 7.25. *Given $AS(\Sigma) = \langle Args(\Sigma), \mathbf{R}\rangle$, if $\mathfrak{s}$ is some logic-based semantics then $AS(\Sigma)$ guarantees the postulates of (closure), (closure under consequence), and (exhaustiveness).*

Lemma 7.26. *Given $AS(\Sigma) = \langle Args(\Sigma), \mathbf{R}\rangle$, if $\mathfrak{s}$ is some logic-based semantics and* **R** *is conflict-exhaustive then $AS(\Sigma)$ guarantees the postulate of (consistency).*

The following theorems expose the conditions by which a `GenAS` guarantees the argumentation postulates.

Theorem 7.27. *Given a* `GenAT` $\tau = \langle \Sigma, \pi, \mathfrak{panf}_\pi, \Omega, \mathfrak{atom}_{[\pi,\Omega]}\rangle \in \mathbb{T}$ *and the* `GenAS` $\mathfrak{genas}(\tau) = \langle \delta, \mathbf{S}_\delta, \mathbf{R}_\delta\rangle \in \mathbb{G}$, *if $\mathcal{L}^\kappa$ is a π-*`pANF` *logic, $\mathcal{L}_{\mathtt{cl}}$ is parallel-complementary logic, Σ has no tautologies, the* `GenAF` *δ is closed under transposition, $\Omega \subseteq \{\omega^1_\pi, \omega^2_\pi\}$ where ω^1_π and ω^2_π are atom constraint functions modeling* $\mathbf{AC}_1$ *and* $\mathbf{AC}_2$*, respectively, and $\mathfrak{s}$ is a logic-based semantics then $\mathfrak{lbas}(\mathfrak{genas}(\tau)) = \langle \mathbf{B}, \mathbf{T}\rangle$ is a logic-based AS guaranteeing the argumentation postulates of (closure), (closure under consequence), (consistency), and (exhaustiveness).*

Theorem 7.28. *Given a* `GenAT` $\tau = \langle \Sigma, \pi, \mathfrak{panf}_\pi, \Omega, \mathfrak{atom}_{[\pi,\Omega]}\rangle \in \mathbb{T}$ *and the* `GenAS` $\mathfrak{genas}(\tau) = \langle \delta, \mathbf{S}_\delta, \mathbf{R}_\delta\rangle \in \mathbb{G}$, *if $\mathcal{L}^\kappa$ is a π-*`pANF` *logic, $\mathcal{L}_{\mathtt{cl}}$ is parallel-complementary logic, Σ has no tautologies, the* `GenAF` *δ is closed under transposition, $\Omega \subseteq \{\omega^1_\pi, \omega^2_\pi\}$ where ω^1_π and ω^2_π are atom constraint functions modeling* $\mathbf{AC}_1$ *and* $\mathbf{AC}_2$*, respectively, $\mathfrak{s}$ is a logic-based semantics, and $Ext_{\mathfrak{s}}(AS(\Sigma)) \neq \emptyset$ then $\mathfrak{lbas}(\mathfrak{genas}(\tau)) = \langle \mathbf{B}, \mathbf{T}\rangle$ is a logic-based AS guaranteeing the five argumentation postulates.*

8. Related and future work

As we have mentioned in the introductory section, this article presents the definitive results of our proposal of generalization of Dung's AFs through Generalized Argumentation Frameworks (`GenAF`s). The preliminary approaches to a `GenAF` were given in Moguillansky et al. (2009) where we proposed the idea of generalization through the consideration of an argument language. Afterwards, in Moguillansky et al. (2010a) we proposed the analysis of `GenAF`s upon $\mathcal{ALC}$ DLs. To that end, we considered Bienvenu's studies upon prime implicates as a way of normalizing the underlying knowledge base for constructing the corresponding `GenAF`. The proposal done in Rotstein et al. (2010) simply refers to the construction of argumental structures as a way for structuring arguments. However, the focus of that article is different, since it relies upon the study of dynamics in abstract argumentation. In the present article, we introduce a new recursive perspective for the construction of argumental structures and a final refinement on the proposals made in Moguillansky et al. (2010a, 2009). Finally, in Section 7, the argumentation machinery is studied on the light of the argumentation postulates proposed in Amgoud (2014).

Several argumentation systems inherit Dung's semantics and use this approach to obtain the accepted arguments; among them we can mention Assumption-Based Argumentation (ABA) (Bondarenko et al., 1997; Bondarenko et al., 1993) and ASPIC+ (Prakken, 2010). ABA is a framework developed to cover approaches to default reasoning formulated in the early 90s (Bondarenko et al., 1997; Bondarenko et al., 1993); represents a generalization of THEORIST (Poole, 1988) which allows any theory formulated in a monotonic logic to be extended by a defeasible set of assumptions. ABA combines Dung's preferred extension semantics for logic programming in argumentation-theoretic terms, and the foundations of abstract argumentation; since ABA is an instance of abstract argumentation, all semantic notions for deciding the "acceptability" of arguments readily apply to arguments produced in ABA. Furthermore, ABA is a general-purpose argumentation framework that can be instantiated to support various applications and specialized frameworks, including: most default reasoning frameworks and problems in legal reasoning, game-theory, practical reasoning, and decision-theory. However, ABA builds actual arguments as deductions supported by assumptions by using inference rules in an underlying logic. The ASPIC+ system (Prakken, 2010) is a general framework for argumentation-based reasoning which aims to define a wide class of argumentation frameworks and, like ABA, uses Dung's semantics as the mechanism to decide acceptability. The system makes use of strict and defeasible rules, and arguments are defined as inference trees formed by applying these two forms of inference rules. As in ABA and ASPIC+, `GenAF` relies on Dung's semantics, still we can find several differences. When comparing `GenAF` with these two systems a particular difference that is worth to mention is the level of abstraction regarding the representation language; by taking advantage of this characteristic, `GenAF` can be instantiated on different representation languages in a similar way as we did here for the $\mathcal{ALC}$ DL.

Abstract Dialectical Frameworks (ADFs) (Brewka, Strass, Ellmauthaler, Wallner, & Woltran, 2013; Brewka & Woltran, 2010), represent a research line that generalizes Dung's abstract argumentation frameworks by associating to each node in the argumentation graph an acceptance condition. An ADF is a directed graph where the nodes represent statements or positions which can be accepted or not. The edges in the graph represent dependencies: the status of a node depends just on the status of its parents, which are the nodes with a direct link to it; additionally, each node has an associated acceptance condition that specify the exact conditions under which the node is accepted. This framework permits modeling different forms of dependencies among the different arguments such as support and attack, as well as more general forms of argument interaction can be represented, e.g. some form of accrual can be introduced; to obtain the acceptability semantics, the standard Dung's semantics is generalized to ADFs. From this description, it is clear that GenAF represents a different approach where the structure of the arguments matters. Nevertheless, GenAF can be used below an ADF by producing the arguments that label the nodes. This could be considered an interesting merge of both lines of research.

Logic-associated abstract argumentation frameworks (LAAF) (Dung & Thang, 2014) are defined by associating abstract argumentation with abstract logics which serve well for the representation of arguments' claims in an abstract manner. This characteristic is very similar to the idea we propose for the specification of an AL-Framework. As we did here through the use of satisfiability checking, abstract logics' properties like weak and strong absurdity seem to fit well for specifying conflicts by abstracting away from referring to complementary literals. (Dung & Thang, 2014) shows that LAAFs capture ASPIC-like systems (without preferences) and ABA, and even provides an equivalent translation from the former to the latter. In addition, authors present certain properties under which LAAFs guarantee closure and consistency properties in a quite similar way as done here regarding conflict-dependence and conflict-exhaustive attack relations, and parallel-complementary logics. The main difference between LAAFs and GenAFs is that in our approach the generalization of the language brings the alternative to concentrate on structuring inner components of arguments for establishing direct relations with normalization methods for the underlying knowledge base. On the other hand, although we did not provide translations to ASPIC+ or ABA, GenAFs has been shown to guarantee some logic-based argumentation postulates under specific conditions. This allows to presume GenAFs' equivalence with any other argumentation system which can be shown to behave as a standard logic-based argumentation system.

The GenAF shares some characteristics with Bipolar Argumentation Frameworks (BAF) (Amgoud, Cayrol, & Lagasquie-Schiex, 2004; Amgoud et al., 2008; Cayrol & Lagasquie-Schiex, 2005). Bipolarity refers to two kinds of interactions between arguments: support and conflict. In a Coalition Argumentation Framework (CAF) (Cayrol & Lagasquie-Schiex, 2010), coalitions of arguments are built to effect support and attack. Acceptability semantics are developed preserving some properties of Dung's AF. CAFs intend to gather as many arguments as possible in a coalition, which cannot be broken afterwards for defense purposes. This is a significant difference with the GenAF, in which argumental atoms within claim-coalitions may participate simultaneously in other claim-coalitions for support purposes only. Another difference with GenAFs is that BAFs and CAFs are defined at an almost complete abstract level, *i.e.*, no assumptions are made on the nature of arguments; thus, no language for arguments is provided. As in Dung's seminal work on AFs, this allows to concentrate mostly on the study of the argumentation semantics; however, the problem of reasoning over inconsistent knowledge bases is not analyzed.

A collective argumentation framework (Bochman, 2003; Nielsen & Parsons, 2006) is an abstract framework built from a set of arguments and an attack relation between sets of arguments. There, a set of arguments can attack other arguments, but contrary to our approach, this relation is not reducible to inner attacks. This is more similar to the assumptions done in CAFs, where the notion of coalition is considered as a whole and its members cannot be used separately in an attack relation. Although both proposals are similar (both define semantics on subsets of arguments), Nielsen and Parsons' (Nielsen & Parsons, 2006) allows sets of arguments to attack single arguments only, and uses similar Dung's semantics, whereas Bochman's (Bochman, 2003) proposal for argumentation semantics gives new specific definitions for stable and admissible sets of arguments. Differences with the GenAF are similar to those with CAFs, based on the abstraction level and the atomicity of coalitions. On the other hand, similarities with Bochman's approach rely on the objective of building arguments from a type of restricted knowledge bases, namely disjunctive logic programs. By taking into consideration the similarities of collective argumentation frameworks and CAFs, an interesting related line of research for enriching GenAFs is to define an attack relation upon sets of arguments. This sort of attack relation may be quite advantageous for non-parallel-complementary logics. As shown in Section 7, a conflict-exhaustive attack relation only can be ensured under parallel-complementary logics (see Proposition 7.15). However, an attack relation based on *collections of argumental structures* may bring a solution by modeling conflicts that appear from/to a set of arguments in contrast to the restrictive attack relation between pairs of arguments.

A similar approach is the one presented by García and Simari (2004), under the name of Defeasible Logic Programming (DeLP): an argumentation based machinery for reasoning about a specialized horn-style knowledge bases, *defeasible logic programs*. Similar to the construction of argumental structures in the GenAF, an argument in DeLP is built from a defeasible logic program as a self conclusive piece of knowledge achieving its claim. Semantics are query-based, that is, upon a query an argument supporting it is built, and thereafter a tree (rooted in the query supporter) of defeaters is constructed; then, the tree is analyzed through an acceptability criterion by marking its nodes, *i.e.*, the arguments. These trees are a sub-graph of the ones resulting from a complete attack relation *à la* Dung, and constitutes a different sort of semantics which are better suited for real applications due to its construction upon queries. Future work on GenAFs' semantics also involves the application of dialectical trees of GenAF-arguments.

In Gorogiannis and Hunter (2011), some problems were investigated for preventing bad uses of conflict relations; for instance, with classical logic and direct undercuts without restrictions, if the knowledge base is inconsistent any argument can be shown to be in a preferred extension. This means that such a combination may end up being too credulous and therefore the instantiation of the framework would not tell us anything useful. However, these sort of drawbacks usually come from the use of an unrestricted logic. GenAFs are able to avoid them through an appropriate specification of the argument language. For instance, the mentioned property holds for direct undercuts by verifying that the claim of an argument entails the negation of some formula included in the second argument (postulate $D1'$ in Gorogiannis and Hunter (2011)). GenAFs avoids this inconvenience by checking unsatisfiability of a set containing a premise of an argumental structure and the claim of a second argumental structure. In this manner, we avoid recognizing conflicts via opposed formulæ which is a problem for reasoning about description logics: negation of DL-axioms may fall out of the scope of the language. For that reason, we rely upon unsatisfiability checking of dismembered parts of formulæ, which allows us to abstract away from such inconveniences of syntax.

A study about characterizations of conflicts, argumentation rationality and their implications has been done for GenAFs in this article by considering the general behavior characterizations for logic-based argumentation proposed in Amgoud (2014). There, Amgoud develops a quite exhaustive analysis on the matter of argumentation postulates. That work has been essential for showing the behavior of instantiated GenAFs (*i.e.*, GenASs). In Section 7 we develop a deep analysis on this subject with interesting conclusions. For instance, the specification of theoretic conditions under which a GenAF is equivalent to a logic-based argumentation system for guaranteeing the five argumentation postulates proposed in Amgoud (2014). Future work on this matter involves the refinement of the essential set of argumentation postulates in order to achieve complete representation theorems. That would allow the construction of logic-based argumentation systems by only ensuring the conditions stated through the postulates.

First-order argumentation was previously studied in Besnard and Hunter (2005, 2008). The main difference with the GenAF is that as any classic logic-based argumentation system, they build arguments directly from subsets of the knowledge base, by relying upon the proof procedure corresponding to the modeled representation language. These kind of frameworks look simpler than ours; in contrast, the GenAF enjoys a higher versatility, being possible to set the argument language framework (AL-framework) for modeling classical logic-based argumentation systems. As aforementioned, Section 7 covers the conditions for such constructions. On the other hand, GenAFs have higher granularity which makes them better for identifying minimal inconsistency/incoherence sources, and even also for handling knowledge base dynamics such as debugging or change operations. In this sense, our proposal is more similar to that in Vreeswijk (1997), were GenAF's argumental atoms are comparable to their arguments since they are interpreted as material conditionals (implications " $\rightarrow$ ") in classical logic. Nevertheless, no notion of deduction nor conflict for FOL interpretations are analyzed in Vreeswijk (1997).

Knowledge base compilation towards efficient argumentation was also studied by Besnard and Hunter in Besnard and Hunter (2006). These authors rely upon their first-order argumentation system (Besnard & Hunter, 2005) to apply compilation techniques to construct arguments more efficiently from the given knowledge base. The argumentation semantics they adopt is based on trees of arguments, similar to García and Simari (2004). We are interested in their techniques for knowledge base compilation and algorithms for constructing arguments to eventually implement an applied GenAF.

The usage of a comparison criterion as done for GenAFs, might be related to Preference-based Argumentation Frameworks (PAFs) (Amgoud & Cayrol, 2002). However, the main objective here is only to provide a tool for avoiding, if necessary, the construction of symmetric rebuttals. Moreover, PAFs relying on non-symmetric attack relations (Amgoud & Vesic, 2009) may avoid modeling attacks between conflicting arguments. This is a clear difference with our usage of a comparison criterion. GenAFs, however, cannot avoid the construction of symmetric undercuts. The justification of such a problem is also related to the discussion presented in Section 7 with regards to the direction of attacks for constructing conflict-sensitive attack relations.

Argumentation is an alternative for reasoning over inconsistent ontologies. This can be implemented by referring to standard DLs reasoners for reasoning about consistent contexts, while an argumentation machinery would provide the specialized extension for dealing with inconsistent blocks of knowledge from the base. For the standard (consistent-based) reasoning machinery, although reasoning about DLs is mostly of a high computational complexity, specialized algorithms perform well in practice for the general case. Novel less expressive description languages appear for improving the computational cost of reasoning. Theories for reasoning over inconsistent ontologies look for optimization methods with considerable success for diminishing computational costs. Knowledge compilation provides an interesting alternative for improving reasoning efficiency admitting an initial preprocessing cost; for instance, the case of unfoldable $\mathcal{ALC}$ constitutes another example of knowledge base compilation for ontologies, where the subsumption problem turns from EXPTIME-Complete to PSPACE-Complete in the case of unfolded axioms. The Bienvenu's *prime implicate normal form* for $\mathcal{ALC}$ DLs (Bienvenu, 2008) is a nice alternative; while the transformation involves an at most doubly-exponential blowup in concept length, the subsumption problem turns into polynomial.

On the other hand, the GenAF machinery here proposed for inconsistency-tolerant reasoning, can be an argumentation-based alternative for the well-known inconsistency-tolerant semantics used for ontology reasoning. Inconsistency-tolerant semantics (Lembo, Lenzerini, Rosati, Ruzzi, & Savo, 2010; Lembo & Ruzzi, 2007) are also referred as repair-based semantics (as seen in Database Theory (Chomicki, 2006)), where a repair is obtained by applying a minimal set of changes for restoring consistency. Since many possible repairs can appear, the approach decides that what is true should be so in all possible repairs of the base. Thus, inconsistency-tolerant query answering computes the answer to a query in all possible repairs. That property is fundamental for the *ABox Repair (AR) Semantics* and all its variants: *Intersection ABox Repair (IAR), Closed ABox Repair (CAR)*, and *Intersection Closed ABox Repair (ICAR)*. In Rosati (2011) the computational complexity for reasoning over inconsistent $\mathcal{ALC}$ ontologies has been shown to correspond to EXPTIME under any of the repair-based semantics. The obvious difference between repair-based semantics and the argumentation-based alternative is that consistency restoration is avoided in the latter. This property is mandatory in some research areas in which inconsistencies are not necessarily representational errors. For instance, knowledge bases representing concepts of medicine or law are probably the most notorious ones for which consistency restoration is specially avoided. However, the computational complexity that is reduced by avoiding calculating each repair is faced once again when constructing the set of arguments for deciding the query.

Moguillansky, Wassermann, and Falappa (2010b) proposed a methodology for constructing the defeaters of a given $\mathcal{ALC}$-argument based on the well known axiom pinpointing techniques proposed in Schlobach et al. (2007) for debugging unfoldable $\mathcal{ALC}$ ontologies. That methodology is shown to be in PSPACE, which corresponds to the same complexity class of Schlobach's debugging methodology. This is useful for reasoning upon dialectical trees, where the tree is constructed upon a root argument supporting the query. On the other hand, argumentation semantics based on grah of arguments like Dung-standards' face a new computational instance beyond the graph construction. For example, the complexity results in abstract argumentation for stable semantics are due to Dimopoulos and Torres (1996), where credulous acceptance has been shown to be in NP-Complete, while for the skeptical case, it corresponds to co-NP-Complete. In this sense, the research for the case of logic-based argumentation is still under way, and is part of the motivation for new appearing restrictions on the underlying language for representing knowledge. For instance, the construction of defeaters proposed in Moguillansky et al. (2010b) relies upon Schloback's axiom pinpointing which is based on a tableaux technique, a sort of chase procedure. A well known inconvenience with chase procedures applied to unrestricted FOL is that they might not end. For controlling such kind of situations some restrictions are usually imposed to the adopted description language and/or to the query expressiveness. That is the purpose, for instance, for restricting the usage of functional symbols in FOL or for relying upon sets of FOL formulæ enjoying the *bounded-treewidth model property* (Goncalves & Grädel, 2000) for ensuring that checking satisfiability keeps being decidable. For more details on decidability of query evaluations and non-termination of chase procedures, the interested reader is referred to Calì, Gottlob, and Kifer (2013). Future work in this direction includes the proposal of optimized algorithms for the construction of arguments and defeaters towards a complete analysis of the computational costs for constructing the argumentation machinery and processing an argumentation semantics.

Regarding software implementations, several argumentation-based applications are actually available for studying argumentation semantics' empirical behavior while pursuing computational efficiency. For a brief compendium, the reader can refer to Thimm et al. (2016). However, to our best knowledge, no machinery is actually running for reasoning over inconsistent ontologies, beyond the experimental partial application reported in Rahwan and Banihashemi (2008). Recently, in Bex, Snaith, Lawrence, and Reed (2014) a software was reported for building an argumentation base from web blogs for constructing debate and discussions. However, the reasoning machinery is not based upon description logics.

9. Conclusions

A novel argumentation framework was presented as a generalization of the classical Dung's AF (Dung, 1995), named through the acronym GenAF. The proposal of the GenAF keeps the abstraction on the logic used to represent knowledge inside arguments while specifying an *argument language* to give some inner-structure to arguments. This allows a generalized form of abstract argumentation which is able to adapt to any knowledge representation language known to conform to some first-order logic (FOL) fragment. Therefore, a GenAF aims at providing a straightforward reifiable argumentation framework for reasoning about inconsistent knowledge bases (KBs). A main characteristic of GenAFs is that formulæ from the underlying KB are interpreted as argumental atoms. To this end, the KB is required to conform to a *pre-argumental normal form* for obtaining a related GenAF. Upon such necessity, we proposed the usage of knowledge compilation techniques, such us (Bienvenu, 2008; Darwiche & Marquis, 2002) among others, as a way of transforming a knowledge base towards efficient querying; in this manner, although an expensive preprocessing cost is faced, compensation will be acquired through computational savings made on later reasoning process upon GenAF.

Applying a GenAF to handle a specific representation language for reasoning over an underlying knowledge base, requires the concretization of the argument language, which involves both concrete sublanguages for claims and premises. We introduce the notion of *coalition*, which is a structure capable of grouping several atoms for inferring a new claim. From this discussion, becomes clear that GenAF's argumental atoms play a smaller role than arguments in classic argumentation frameworks: they are aggregated into *structures* towards the achievement of a specific claim. The idea behind the aggregation of atoms within a structure is similar to that of sub-arguments (Martínez et al., 2007). A GenAF considers two different kinds of structures: *full arguments*, which are self-conclusive, as usual arguments in frameworks like (Besnard & Hunter, 2005; 2008) among others; and *potential arguments*, which cannot achieve their claims unless their premises are supported. Note that instantiation of variables within a potential argument may occur as a consequence of its premises being supported, giving rise to full arguments.

Defeats in a GenAF are defined from pairs of conflictive structures rendering two different classes of attack relations: between full arguments and between potential arguments. The two types of attacks stand for identifying two kinds of conflicting sources of information from the underlying knowledge base; that is, inconsistency and incoherence (Flouris et al. 2006). Intuitively, a source of incoherence from a KB is a piece of information that, although it does not infer inconsistency, it admits only empty interpretations for any model of the KB; thus, in a consistent KB, incoherence can appear unveiling a kind of "pre-inconsistency". Coping with incoherences is an important matter in the area of ontology debugging and repairing.

As aforementioned, GenAF's versatility allows to cope with different logics for arguments. In particular, the reification of the abstract argument language into the $\mathcal{ALC}$ description language renders an interesting methodology for ontology reasoning without the need to repair nor debug the ontology, thus bringing a way to reason on top of inconsistencies. This allows to avoid loosing knowledge which is of utmost relevance in areas of application like medicine and law where knowledge bases are naturally expected to be inconsistent.

The argumentation machinery proposed here is semantically determined –by the standard set theoretic Tarskian semantics. This allows to propose further implementations relying on some preferred reasoner engine; for instance, an $\mathcal{ALC}$-GenAF may be handled via tableaux techniques which are usually used to implement ontology reasoners. On the other hand, the implementation of an $\mathcal{ALC}$-GenAF could be developed straightforwardly, over the normalized $\mathcal{ALC}$ ontology, emulating the argumentation machinery. This is possible given that $\mathcal{ALC}$ axioms are interpreted as argumental atoms in the GenAF, and therefore, the normalized ontology serves as a repository of atoms for the argumentation machinery to work.

Regarding argumentation rationality, the five postulates proposed in Amgoud (2014) has been shown to be guaranteed for a GenAS under specific conditions. A GenAS constructed upon a π-pANF logic, brings a direct relation between the inner structure of arguments and formulæ in the underlying knowledge base, but also ensures a manageable set of arguments. In this sense, the number of alternative conclusions that an argument's body may trigger is kept considerably low since claims are restricted to a specific normal form. Also, tautologies has been shown in Amgoud (2014) to be a real drawback in logic-based argumentation systems since they cannot be plausible conclusions. A GenAS guaranteeing the five postulates prohibits tautologies for avoiding being part of the conclusions of a system. In addition, the underlying GenAF when providing the constraints of consistency and non-circularity may avoid unnecessary construction of inconsistent and/or tautologic argumental atoms. Finally, we have shown that the five argumentation postulates are guaranteed under *logic-based semantics*. This specialized semantics for logic-based argumentation systems ensures a complete construction of extensions. Standard semantics has been formalized mainly for abstract argumentation. For that reason, under certain circumstances classical admissibility may bring sets of arguments constructed from an underlying source of inconsistency. The problem is solved through the definition of logic-based admissibility which ensures extensions where all the possible arguments are involved.

Acknowledgements

We thank the anonymous reviewers for their many insightful comments and suggestions, which helped us improving this manuscript. We also thank Nicolás D. Rotstein and Marcelo A. Falappa for their contribution to preliminary GenAF versions. This work has been partially supported by Universidad Nacional del Sur (UNS) and National Council of Scientific and Technical Research (CONICET).

Appendix

Lemma 5.4. *Given a* GenAT $\tau = \langle \Sigma, \pi, \mathfrak{panf}_\pi, \Omega, \mathfrak{atom}_{[\pi,\Omega]} \rangle \in \mathbb{T}$ *and the* GenAS $\mathfrak{genas}(\tau) = \langle \delta, \mathbf{S}_\delta, \mathbf{R}_\delta \rangle$*; if* Σ *is consistent then* $\mathbf{C}_\delta = \emptyset$.

Proof. By *reductio ad absurdum*, we assume Σ is consistent but $\mathbf{C}_\delta \neq \emptyset$. We also assume $(\mathcal{S}_1, \mathcal{S}_2) \in \mathbf{C}_\delta$, where $\mathcal{S}_1 \in \mathbf{S}_\delta$ and $\mathcal{S}_2 \in \mathbf{S}_\delta$ are full arguments, and we know both structures are either rebuttals or undercuts. However, in the case they are undercuts, from Definition 4.8, we know there is some $\mathcal{S}_2' \trianglelefteq \mathcal{S}_2$ such that $(\mathcal{S}_1, \mathcal{S}_2') \in \mathbf{C}_\delta$ is a rebuttal. This means that $\{\mathfrak{cl}(\mathcal{S}_1), \mathfrak{cl}(\mathcal{S}_2')\} \models \perp$. Afterwards, from Definition 5.3, we know that $\mathfrak{base}(\{\mathcal{S}_1, \mathcal{S}_2'\}) \subseteq \mathfrak{panf}_\pi(\Sigma)$ holds and since $\mathfrak{base}(\{\mathcal{S}_1\}) \models \mathfrak{cl}(\mathcal{S}_1)$ and $\mathfrak{base}(\{\mathcal{S}_2'\}) \models \mathfrak{cl}(\mathcal{S}_2')$, afterwards $\mathfrak{base}(\{\mathcal{S}_1, \mathcal{S}_2'\}) \models \{\mathfrak{cl}(\mathcal{S}_1), \mathfrak{cl}(\mathcal{S}_2')\}$ and thus $\mathfrak{base}(\{\mathcal{S}_1, \mathcal{S}_2'\}) \models \perp$ hold.

It is easy to see that $\Sigma \models \bot$, *i.e., KB* s inconsistent which is absurd. Finally, $\mathbf{C}_\delta = \emptyset$ holds. □

Theorem 5.6. *Given a* GenAT $\tau = \langle \Sigma, \pi, \mathfrak{panf}_\pi, \Omega, \mathfrak{atom}_{[\pi,\Omega]} \rangle \in \mathbb{T}$ *and the* GenAS $\mathfrak{genas}(\tau) = \langle \delta, \mathbf{S}_\delta, \mathbf{R}_\delta \rangle$*; if* Σ *is coherent and consistent then* $\mathbf{R}_\delta = \emptyset$*.*

Proof. By *reductio ad absurdum*, we will assume Σ is coherent and consistent, but $\mathbf{R}_\delta \neq \emptyset$. From Lemma 5.4, we know that $\mathbf{C}_\delta = \emptyset$ hence, from Definition 4.10 we know that for any $(\mathcal{S}_1, \mathcal{S}_2) \in \mathbf{R}_\delta$, at least one of the structures $\mathcal{S}_1$ or $\mathcal{S}_2$ is a potential argument. Moreover, from Definition 4.8, three options arise: either (1) $\mathcal{S}_1$ rebuts $\mathcal{S}_2$, (2) $\mathcal{S}_1$ directly undercuts $\mathcal{S}_2$, or (3) $\mathcal{S}_1$ undercuts $\mathcal{S}_2$.

We will assume without loss of generality, that $(\mathcal{S}_1, \mathcal{S}_2) \in \mathbf{R}_\delta$ is a base attack, that is, for any $\mathcal{S}'_1 \trianglelefteq \mathcal{S}_1$ and any $\mathcal{S}'_2 \trianglelefteq \mathcal{S}_2$ such that $(\mathcal{S}'_1, \mathcal{S}'_2) \in \mathbf{R}_\delta$, it holds $\mathcal{S}'_1 = \mathcal{S}_1$ and $\mathcal{S}'_2 = \mathcal{S}_2$. Said this, we know that case 3) is subsumed by case 1).

For (1), we know $\{\mathfrak{cl}(\mathcal{S}_1), \mathfrak{cl}(\mathcal{S}_2)\} \models \bot$ and that there is at least one of both structures which is a potential argument. This means that $\mathfrak{pr}(\mathcal{S}_1) \cup \mathfrak{pr}(\mathcal{S}_2)$ is non-empty, which means that for each $\rho \in \mathfrak{pr}(\mathcal{S}_1) \cup \mathfrak{pr}(\mathcal{S}_2)$ we can assume the existence of a formula $\varphi \in \mathfrak{base}(\{\mathcal{S}_1, \mathcal{S}_2\})$ such that φ has the form $\rho \wedge \rho_1 \wedge \ldots \wedge \rho_n \to \beta$ (π-pANF formula). It is easy to see that for any model $\mathcal{I}$ of $\mathfrak{base}(\{\mathcal{S}_1, \mathcal{S}_2\})$, we have that $(\mathfrak{pr}(\mathcal{S}_1) \cup \mathfrak{pr}(\mathcal{S}_2))^\mathcal{I} = \emptyset$. Afterwards, for every formula φ, we have that $\rho^\mathcal{I} \cap \rho_1^\mathcal{I} \cap \ldots \cap \rho_n^\mathcal{I} = \emptyset$. This means that the free premises cannot be supported since it would trigger the inconsistency determined by the conflict of claims $\{\mathfrak{cl}(\mathcal{S}_1), \mathfrak{cl}(\mathcal{S}_2)\} \models \bot$. Hence, Σ is incoherent which is again absurd, and hence, $\mathcal{S}_1$ cannot rebut $\mathcal{S}_2$.

For (2), we have that $\{\mathfrak{cl}(\mathcal{S}_1), \rho\} \models \bot$, where $\rho \in \mathfrak{pr}(\mathcal{S}_2)$ (see Definition 4.8), and thus $\{\mathfrak{cl}(\mathcal{S}_1), \rho\}$ is unsatisfiable, which means that $\mathfrak{cl}(\mathcal{S}_1)^\mathcal{I} \cap \rho^\mathcal{I} = \emptyset$ for every model $\mathcal{I}$ of $\mathfrak{base}(\{\mathcal{S}_1, \mathcal{S}_2\})$. Hence, we can assume there is a formula $\varphi \in \mathfrak{base}(\{\mathcal{S}_2\})$ that has the form $\rho \wedge \rho_1 \wedge \ldots \wedge \rho_n \to \beta$ (π-pANF formula). It is clear that $\rho^\mathcal{I} \cap \rho_1^\mathcal{I} \cap \ldots \cap \rho_n^\mathcal{I} = \emptyset$. Hence, Σ is incoherent which is absurd. Finally, $\mathcal{S}_1$ cannot be a direct undercut of $\mathcal{S}_2$.

Finally, it is clear that $\mathbf{R}_\delta = \emptyset$ holds. □

Proposition 5.13. *Given a* GenAS $\langle \delta, \mathbf{S}_\delta, \mathbf{R}_\delta \rangle \in \mathbb{G}$*, if* $\mathcal{S} \in \mathbf{S}_\delta$ *then* $\mathfrak{base}(\{\mathcal{S}\}) \models (\bigwedge \mathfrak{pr}(\mathcal{S})) \to \mathfrak{cl}(\mathcal{S})$*.*

Proof. By *reductio ad absurdum*, we assume that $\mathfrak{base}(\{\mathcal{S}\}) \models (\bigwedge \mathfrak{pr}(\mathcal{S}) \to \mathfrak{cl}(\mathcal{S}))$ does not hold. This means that there is an interpretation $\mathcal{I}$ that makes all the formulæ in $\mathfrak{base}(\{\mathcal{S}\})$ true and that the same interpretation does not make true $(\bigwedge \mathfrak{pr}(\mathcal{S}) \to \mathfrak{cl}(\mathcal{S}))$. Hence, for all $\vartheta \in \mathfrak{base}(\{\mathcal{S}\})$, $\vartheta^\mathcal{I}$ is non-empty but $(\bigwedge \mathfrak{pr}(\mathcal{S}) \to \mathfrak{cl}(\mathcal{S}))^\mathcal{I} = \emptyset$. Let's analyze the construction of $\mathcal{S}$. From Definition 4.5, we have that there is a claim-coalition $\mathcal{C}$ for $\mathfrak{cl}(\mathcal{S})$, and from Definition 4.2, that $\mathfrak{clset}(\mathfrak{bd}(\mathcal{C})) \models_{(\mathcal{I},v)} \mathfrak{cl}(\mathcal{S})$ and $\mathfrak{pr}(\mathcal{C}) = \mathfrak{prset}(\mathfrak{bd}(\mathcal{C}))[v]$, which means that the left-hand side of every formulæ in $\mathfrak{base}(\mathfrak{bd}(\mathcal{C}))$ has been substituted in the same manner that their right-hand sides has and that the coalition's claim has for reaching $\mathfrak{cl}(\mathcal{S})$. All the substituted left-hand sides are the premises of $\mathcal{C}$. In Definition 4.5, some of such premises are supported by a set $\Phi \subseteq \mathcal{L}_{\mathtt{cl}}$, where each $\alpha \in \Phi$ is the claim of another argumental structure, *i.e.*, the supporting structures of $\mathcal{S}$. (We rely upon a set Φ given that we are working with a legal argument language, thus from Definition 3.6, condition 1, we have that for any $\rho \in \Gamma$, there is a set $\Phi \subseteq \mathcal{L}_{\mathtt{cl}}$ such that $\Phi \models \rho$.) By Definition 4.3 we have that $\Phi \models_{(\mathcal{I},v)} \rho$, where $\rho \in \mathfrak{pr}(\mathcal{C})$. This means that there is an interpretation $\mathcal{I}$ and a substitution v through which each claim in Φ is true and also is made true ρ. The same interpretation is also used for each of the premises of $\mathcal{C}$ that is supported via a set of supporting structures. Afterwards, the unsupported premises of $\mathcal{C}$ along with the premises of the supporting structures constitute the premises of the structure $\mathcal{S}$ which are still unsupported (free premises wrt. $\mathfrak{bd}(\mathcal{S})$). This means that every interpretation $\mathcal{I}$ which is a model of each formula $\vartheta \in \mathfrak{base}(\{\mathcal{S}\})$ is also a model of $\mathfrak{cl}(\mathcal{S})$ and therefore, $\mathfrak{cl}(\mathcal{S})^\mathcal{I}$ is non-empty as well. Observe that for each premise $\rho \in \mathfrak{pr}(\mathcal{S})$ there is a formula $(\rho \wedge \rho_1 \wedge \ldots \wedge \rho_n \to \alpha) \in \mathfrak{base}(\{\mathcal{S}\})$, that as we said, has a non-empty interpretation $\mathcal{I}$, *i.e.*, $(\rho \wedge \rho_1 \wedge \ldots \wedge \rho_n \to \alpha)^\mathcal{I} \neq \emptyset$. Afterwards, $(\rho \wedge \rho_1 \wedge \ldots \wedge \rho_n)^\mathcal{I} \subseteq \alpha^\mathcal{I}$, and thus $(\bigwedge \mathfrak{pr}(\mathcal{S}))^\mathcal{I} \subseteq \mathfrak{cl}(\mathcal{S})^\mathcal{I}$, for a non-empty interpretation model $\mathcal{I}$, which is equivalent to say that $\mathcal{I}$ is a model of $(\bigwedge \mathfrak{pr}(\mathcal{S})) \to \mathfrak{cl}(\mathcal{S})$. This means that $(\bigwedge \mathfrak{pr}(\mathcal{S}) \to \mathfrak{cl}(\mathcal{S}))^\mathcal{I} \neq \emptyset$, which is absurd. □

Theorem 5.14. *Given a knowledge base* $\Sigma \subseteq \mathcal{L}^\mathcal{K}$ *and a* π-pANF *formula* $\vartheta \in \mathcal{L}^\mathcal{K}$*, if* Σ *is coherent and consistent then* $\Sigma \models \vartheta$ *iff* $\Sigma \mid\!\approx_{[\tau,\mathfrak{s}]} \vartheta$*.*

Proof. Assuming $\mathfrak{s}$ as any of the standard semantics in Definition 5.10, we have at least the grounded extension $\mathbf{E} \in \mathfrak{ext}_\mathfrak{s}(\mathfrak{genas}(\Sigma))$ which is the least fixed point of the characteristic function, *i.e.*, $\mathcal{F}(\mathbf{E}) = \mathbf{E}$. Afterwards, it is easy to see that $\mathbf{E} = \mathbf{S}_\delta$ given that $\mathbf{R}_\delta = \emptyset$ (see Theorem 5.6). For any formula $\vartheta \in \mathcal{L}^\mathcal{K}$, if $\Sigma \models \vartheta$ holds then we can assume there is an argumental structure $\mathcal{S} \in \mathbf{S}_\delta$ such that $(\bigwedge \mathfrak{pr}(\mathcal{S})) \to \mathfrak{cl}(\mathcal{S}) = \vartheta$. Finally, since $\mathbf{E} = \mathbf{S}_\delta$, we have that $\mathcal{S} \in \mathbf{E}$ and given that $\mathbf{E}$ is an extension, we have that $\mathcal{S}$ has no defeaters in $\mathbf{E}$ which means that ϑ is warranted and thus $\Sigma \mid\!\approx_{[\tau,\mathfrak{s}]} \vartheta$.

On the other hand, if $\Sigma \mid\!\approx_{[\tau,\mathfrak{s}]} \vartheta$ then from Definition 5.12, we know that there is a structure $\mathcal{S} \in \mathbf{S}_\delta$ such that $(\bigwedge \mathfrak{pr}(\mathcal{S})) \to \mathfrak{cl}(\mathcal{S}) = \vartheta$ is warranted. From Proposition 5.13 we know that $\mathfrak{base}(\{\mathcal{S}\}) \models \vartheta$, and since Σ is consistent/coherent and $\mathfrak{base}(\{\mathcal{S}\}) \subseteq \mathfrak{panf}_\pi(\Sigma)$, we also know that $\Sigma \models \vartheta$. □

Proposition 6.2. *Given the AL-framework* $\pi \in \mathbb{L}$ *for* $\mathcal{ALC}$*, the languages* $\mathcal{L}_{\mathtt{pr}} ::= L$ *and* $\mathcal{L}_{\mathtt{cl}} ::= Cl|Cl(a)|R(a,b)$ *determine a legal argument language* $\mathbb{A}_\pi$*.*

Proof. According to Definition 3.6, for any $\rho \in \mathcal{L}_{\mathtt{pr}}$ there is a set $\Phi \subseteq \mathcal{L}_{\mathtt{cl}}$ such that $\Phi \models \rho$. Since *Cl* is a disjunction of *L*, it is always possible to verify $\Phi \models \rho$. For the second condition in Definition 3.6, we need to show that for any $\mathcal{ALC}$ axiom φ (according to the original $\mathcal{ALC}$ grammar given in Section 6.1) there is a set Δ of axioms $(\bigwedge \Gamma) \to \alpha$, where $\Gamma \subseteq \mathcal{L}_{\mathtt{pr}}$ and $\alpha \in \mathcal{L}_{\mathtt{cl}}$, such that $\Delta \models \varphi$. It is easy to see that $(\bigwedge \Gamma) \to \alpha$ will correspond to an axiom $(\bigwedge L) \to Cl$, which is equivalent to $Cb \to Cl$, and considering the $\mathcal{ALC}$ semantics given in Section 6.1, it is also equivalent to $Cb \sqsubseteq Cl$, which is a π-pANF $\mathcal{ALC}$ axiom according to Definition 6.1. Hence, Δ will contain π-pANF axioms $Cb \sqsubseteq Cl$. We need to show that $\Delta \models \varphi$. To this end, assuming φ as $C \sqsubseteq D$; if *D* is disjunctive and *C* is conjunctive, then $C \sqsubseteq D$ has the form $Cb \sqsubseteq Cl$ and therefore $\Delta \models \varphi$. Besides, if 1) *D* is a conjunctive concept and/or 2) *C* is a disjunctive concept, two alternatives arise: assuming for 1) $D = D_1 \sqcap D_2$, it is easy to see that each conjunctive clause in *D* triggers a different π-pANF axiom, *i.e.*, $C \sqsubseteq D_1$ and $C \sqsubseteq D_2$. Afterwards, if both D_1 and D_2 are disjunctive concepts (and *C* is a conjunctive concept), then $C \sqsubseteq D_1$ and $C \sqsubseteq D_2$ are π-pANF axioms $Cb \sqsubseteq Cl$. The case for 2), assuming $C = C_1 \sqcup C_2$, is solved similarly with a pair of axioms $C_1 \sqsubseteq D$ and $C_2 \sqsubseteq D$, which will be π-pANF axioms $Cb \sqsubseteq Cl$ if it is the case that C_1 and C_2 are both conjunctive concepts (and *D* is a disjunctive concept). Finally, it is easy to see that $\Delta \models \varphi$, and hence, $\mathbb{A}_\pi$ is a legal argument language. □

Proposition 7.5. *Given a* GenAS $\langle \delta, \mathbf{S}_\delta, \mathbf{R}_\delta \rangle \in \mathbb{G}$*, if* $\mathcal{S} \in \mathbf{S}_\delta$ *then* $\mathfrak{lbarg}(\mathcal{S}) \in \mathbb{B}$ *is a logic-based argument.*

Proof. We need to show that $\mathfrak{lbarg}(\mathcal{S})$ satisfies the three conditions of a logic-based argument: 1) $\mathfrak{base}(\{\mathcal{S}\}) \models (\bigwedge \mathfrak{pr}(\mathcal{S}) \to \mathfrak{cl}(\mathcal{S}))$, 2) $\mathfrak{base}(\{\mathcal{S}\}) \not\models \bot$, and 3) there is no $X \subset \mathfrak{base}(\{\mathcal{S}\})$ such that $X \models (\bigwedge \mathfrak{pr}(\mathcal{S}) \to \mathfrak{cl}(\mathcal{S}))$.

1) has been proven in Proposition 5.13.

For 2), by *reductio ad absurdum*, we assume that $\mathfrak{base}(\{\mathcal{S}\}) \models \bot$. This means that $\mathfrak{base}(\{\mathcal{S}\})$ is unsatisfiable and therefore, that there is no interpretation model for $\mathfrak{base}(\{\mathcal{S}\})$ which is an absurd given that we have shown in Proposition 5.13 that such model exists for reaching the argument's claim from its premises. Thus, 2) has been proven.

To show 3) it is sufficient to refer to minimality (condition 7) in Definition 4.5. Afterwards, since there is no structure $\mathcal{S}'$ such that $\mathfrak{bd}(\mathcal{S}') \subset \mathfrak{bd}(\mathcal{S})$, $\mathfrak{cl}(\mathcal{S}) = \mathfrak{cl}(\mathcal{S}')$ and $\mathfrak{pr}(\mathcal{S}) = \mathfrak{pr}(\mathcal{S}')$, we know that there is no subset $X \subset \mathfrak{base}(\{\mathcal{S}\})$ such that $X \models (\bigwedge \mathfrak{pr}(\mathcal{S})) \rightarrow \mathfrak{cl}(\mathcal{S})$.

Finally, it is clear that $\mathfrak{lbarg}(\mathcal{S}) \in \mathbb{B}$ holds. □

Lemma 7.8. *Given a* GenAT $\tau = \langle \Sigma, \pi, \mathfrak{panf}_\pi, \Omega, \mathfrak{atom}_{[\pi,\Omega]} \rangle \in \mathbb{T}$, *the* GenAS $\mathfrak{genas}(\tau) = \langle \delta, \mathbf{S}_\delta, \mathbf{R}_\delta \rangle \in \mathbb{G}$, *and the translated logic-based AS* $\mathfrak{lbas}(\mathfrak{genas}(\tau)) = \langle \mathbf{B}, \mathbf{T} \rangle$*; if* $\mathcal{L}^\kappa$ *is a* π-pANF *logic,* Σ *has no tautologies, the* GenAF δ *is closed under transposition, and* $\Omega \subseteq \{\omega_\pi^1, \omega_\pi^2\}$ *where* ω_π^1 *and* ω_π^2 *are atom constraint functions modeling* $\mathbf{AC}_1$ *and* $\mathbf{AC}_2$*, respectively, then* $\mathbf{B} = Args(\Sigma)$.

Proof. Since $\mathcal{L}^\kappa$ is a π-pANF logic and we know that $\Sigma \subset \mathcal{L}^\kappa$, we have that Σ is a π-pANF knowledge base and thus, it follows that $\mathfrak{panf}_\pi(\Sigma) = \Sigma$ holds. We need to show that 1) $\mathbf{B} \subseteq Args(\Sigma)$ and 2) $Args(\Sigma) \subseteq \mathbf{B}$.

For 1), by *reductio ad absurdum* we will assume there is an argument $\mathcal{B} \in \mathbf{B}$ such that $\mathcal{B} \notin Args(\Sigma)$. From Definition 7.2 we know that $Args(\Sigma) = \{\mathcal{B} \in \mathbb{B} | \mathfrak{bd}(\mathcal{B}) \subseteq \Sigma\}$. From Proposition 7.5 we know that $\mathcal{B} \in \mathbb{B}$, so the only alternative for $\mathcal{B} \notin Args(\Sigma)$ is to assume that there is a formula $\vartheta \in \mathfrak{bd}(\mathcal{B})$ such that $\vartheta \notin \Sigma$ in order to show that $\mathfrak{bd}(\mathcal{B}) \nsubseteq \Sigma$ holds. From Proposition 7.5 we know that $\mathcal{B}$ is constructed from a structure $\mathcal{S} \in \mathbf{S}_\delta$ where $\mathfrak{base}(\{\mathcal{S}\}) = \mathfrak{bd}(\mathcal{B})$, so we have that $\vartheta \in \mathfrak{base}(\{\mathcal{S}\})$. Afterwards, from Definition 5.3, we know that $\vartheta \in \mathfrak{panf}_\pi(\Sigma)$, and since $\mathfrak{panf}_\pi(\Sigma) = \Sigma$ holds, we have that $\vartheta \in \Sigma$ which is an absurd. Finally, $\mathbf{B} \subseteq Args(\Sigma)$ holds.

For 2), once again, by *reductio ad absurdum* we will assume that there is an argument $\mathcal{B} \in Args(\Sigma)$ such that $\mathcal{B} \notin \mathbf{B}$. From Proposition 7.5 we know that there is no structure $\mathcal{S} \in \mathbf{S}_\delta$ such that $\mathfrak{lbarg}(\mathcal{S}) = \mathcal{B}$. Hence, either there is a formula $\vartheta \in \mathfrak{bd}(\mathcal{B})$ such that (a) $\vartheta \notin \mathfrak{base}(\{\mathcal{S}\})$, or (b) $\mathfrak{cl}(\mathcal{B}) \neq (\bigwedge \mathfrak{pr}(\mathcal{S}) \rightarrow \mathfrak{cl}(\mathcal{S}))$.

For a), we know that $\mathfrak{bd}(\mathcal{B}) \subseteq \Sigma$, thus, $\vartheta \in \Sigma$ and since we know $\mathfrak{panf}_\pi(\Sigma) = \Sigma$, we also have that $\vartheta \in \mathfrak{panf}_\pi(\Sigma)$. Afterwards, from Definition 5.3, we have that for every $\mathcal{A} \in \mathfrak{bd}(\mathcal{S})$, either $\mathcal{A} = \mathfrak{atom}_{[\pi,\Omega]}(\varphi)$ or $\mathcal{A} = \mathfrak{atom}_{[\pi,\Omega]}(\varphi^-)$ holds for any $\varphi \in \mathfrak{panf}_\pi(\Sigma)$. Hence, the alternatives are that $\mathcal{A} \notin \mathbf{A}$ where $\mathcal{A} = \mathfrak{atom}_{[\pi,\Omega]}(\vartheta^-)$ –which is absurd given that by hypothesis we know δ is closed under transposition– or neither $\mathcal{A} = \mathfrak{atom}_{[\pi,\Omega]}(\vartheta)$ nor $\mathcal{A} = \mathfrak{atom}_{[\pi,\Omega]}(\vartheta^-)$ can be verified. Hence, $\mathcal{A} \notin \mathbb{A}_\pi$ given that $\mathcal{A} = \epsilon$ (see Definition 3.12). The alternatives are, either $\mathcal{A}$ is circular (*i.e.*, $\langle \Gamma, \alpha \rangle$, where $\Gamma \vDash \alpha$) implying that ϑ is tautologic which would violate the hypothesis, or $\mathcal{A}$ is not consistent (Definition 3.3) implying $\vartheta \vDash \bot$ which would violate consistency from Definition 7.1 and hence $\mathcal{B} \notin Args(\Sigma)$ which is absurd, or there is some constraint in Ω which $\mathcal{A}$ cannot verify –which again is absurd given that by hypothesis Ω models at most $\mathbf{AC}_1$ (consistency) and $\mathbf{AC}_2$ (non-circularity), for which case we would be violating again either consistency from Definition 7.1 and hence $\mathcal{B} \notin Args(\Sigma)$ which is absurd or the hypothesis requiring non-tautologic formulæ. Finally, $\vartheta \in \mathfrak{base}(\{\mathcal{S}\})$.

For b), since $\mathfrak{bd}(\mathcal{B}) = \mathfrak{base}(\{\mathcal{S}\})$, in order to verify $\mathfrak{cl}(\mathcal{B}) \neq (\bigwedge \mathfrak{pr}(\mathcal{S}) \rightarrow \mathfrak{cl}(\mathcal{S}))$ we only have the alternative that $\mathfrak{cl}(\mathcal{B})$ *iff* $(\bigwedge \mathfrak{pr}(\mathcal{S}) \rightarrow \mathfrak{cl}(\mathcal{S}))$. Afterwards, since by hypothesis we know that $\mathcal{L}^\kappa$ is a π-pANF logic, we necessarily have that there is no structure $\mathcal{S}' \in \mathbf{S}_\delta$ such that $\mathfrak{base}(\{\mathcal{S}'\}) \models \mathfrak{cl}(\mathcal{B})$, however since $\mathfrak{bd}(\mathcal{B}) \models \mathfrak{cl}(\mathcal{B})$ and $\mathfrak{bd}(\mathcal{B}) = \mathfrak{base}(\{\mathcal{S}\})$ we know that $\mathfrak{base}(\{\mathcal{S}\}) \models \mathfrak{cl}(\mathcal{B})$. Finally, $\mathcal{S}' = \mathcal{S}$ which is absurd. Hence, $Args(\Sigma) \subseteq \mathbf{B}$ holds.

Finally, $\mathbf{B} = Args(\Sigma)$ holds. □

Lemma 7.9. *Given a* GenAT $\tau = \langle \Sigma, \pi, \mathfrak{panf}_\pi, \Omega, \mathfrak{atom}_{[\pi,\Omega]} \rangle \in \mathbb{T}$, *the* GenAS $\mathfrak{genas}(\tau) = \langle \delta, \mathbf{S}_\delta, \mathbf{R}_\delta \rangle \in \mathbb{G}$, *and the translated logic-based AS* $\mathfrak{lbas}(\mathfrak{genas}(\tau)) = \langle \mathbf{B}, \mathbf{T} \rangle$*; if* $\mathcal{L}^\kappa$ *is a* π-pANF *logic,* Σ *has no tautologies, the* GenAF δ *is closed under transposition, and* $\Omega \subseteq \{\omega_\pi^1, \omega_\pi^2\}$ *where* ω_π^1 *and* ω_π^2 *are atom constraint functions modeling* $\mathbf{AC}_1$ *and* $\mathbf{AC}_2$*, respectively, then* $\mathfrak{warrant}^{\mathbf{C}}_{[\tau,\mathfrak{s}]}(\Sigma) = \{\mathfrak{clset}(\mathcal{E}) | \mathcal{E} \in Ext_{\mathfrak{s}}(\mathfrak{lbas}(\mathfrak{genas}(\tau)))\}$.

Proof. From Definition 5.15 we know that $\mathfrak{warrant}^{\mathbf{C}}_{[\tau,\mathfrak{s}]}(\Sigma) = \{\{(\bigwedge \mathfrak{pr}(\mathcal{S})) \rightarrow \mathfrak{cl}(\mathcal{S}) | \mathcal{S} \in \mathbf{E}\} | \mathbf{E} \in \mathfrak{ext}^{\mathbf{C}}_{\mathfrak{s}}(\mathfrak{genas}(\tau))\}$. From Definition 7.7 we know that $\mathbf{R}$ is such that for every $(\mathcal{S}_1, \mathcal{S}_2) \in \mathbf{R}_\delta$ there is $(\mathcal{B}_1, \mathcal{B}_2) \in \mathbf{R}$ where $\mathcal{B}_1 = \mathfrak{lbarg}(\mathcal{S}_1)$ and $\mathcal{B}_2 = \mathfrak{lbarg}(\mathcal{S}_2)$. Which means that for every extension $\mathcal{E} \in Ext_{\mathfrak{s}}(\mathfrak{lbas}(\mathfrak{genas}(\tau)))$ there is an extension $\mathbf{E} \in \mathfrak{ext}_{\mathfrak{s}}(\mathfrak{genas}(\tau))$ such that for any $\mathcal{B} \in \mathcal{E}$ there is $\mathcal{S} \in \mathbf{E}$, and viceversa. Afterwards, for each $X \in \mathfrak{warrant}_{[\tau,\mathfrak{s}]}(\Sigma)$ we have $(\bigwedge \mathfrak{pr}(\mathcal{S})) \rightarrow \mathfrak{cl}(\mathcal{S}) \in X$ which therefore is equivalent to $\mathfrak{cl}(\mathcal{B}) \in \mathcal{E}$. Hence, $\mathfrak{warrant}^{\mathbf{C}}_{[\tau,\mathfrak{s}]}(\Sigma) = \{\mathfrak{clset}(\mathcal{E}) | \mathcal{E} \in Ext_{\mathfrak{s}}(\mathfrak{lbas}(\mathfrak{genas}(\tau)))\}$. □

Proposition 7.12. *Given a* GenAT $\tau = \langle \Sigma, \pi, \mathfrak{panf}_\pi, \Omega, \mathfrak{atom}_{[\pi,\Omega]} \rangle \in \mathbb{T}$, *the* GenAS $\mathfrak{genas}(\tau) = \langle \delta, \mathbf{S}_\delta, \mathbf{R}_\delta \rangle \in \mathbb{G}$, *and the translated logic-based AS* $\mathfrak{lbas}(\mathfrak{genas}(\tau)) = \langle Args(\Sigma), \mathbf{R} \rangle$, *the attack relation* $\mathbf{R}$ *verifies* R_2.

Proof. This demonstration is trivial since $(\mathcal{B}_3, \mathcal{B}_1)$ can be always a rebuttal and therefore, $(\mathcal{B}_3, \mathcal{B}_2)$ is an undercut. □

Proposition 7.13. *Given a* GenAT $\tau = \langle \Sigma, \pi, \mathfrak{panf}_\pi, \Omega, \mathfrak{atom}_{[\pi,\Omega]} \rangle \in \mathbb{T}$, *the* GenAS $\mathfrak{genas}(\tau) = \langle \delta, \mathbf{S}_\delta, \mathbf{R}_\delta \rangle \in \mathbb{G}$, *and the translated logic-based AS* $\mathfrak{lbas}(\mathfrak{genas}(\tau)) = \langle Args(\Sigma), \mathbf{R} \rangle$, *the attack relation* $\mathbf{R}$ *is conflict-dependent.*

Proof. Since $\mathbf{R}$ is constructed from $\mathbf{C}_\delta$, any pair in $\mathbf{R}$ is a rebuttal or an undercut. Moreover, since every undercut is constructed from some underlying rebuttal, it is sufficient to study only rebuttals since its results would be trivially extended to undercuts. Hence, for any $(\mathcal{B}_1, \mathcal{B}_2) \in \mathbf{R}$ we have a pair $(\mathcal{S}_1, \mathcal{S}_2) \in \mathbf{C}_\delta$ such that $\mathcal{B}_1 = \mathfrak{lbarg}(\mathcal{S}_1)$ and $\mathcal{B}_2 = \mathfrak{lbarg}(\mathcal{S}_2)$. This means that $\mathfrak{cl}(\mathcal{S}_1) \cup \mathfrak{cl}(\mathcal{S}_2) \models \bot$. And since, both structures are full arguments, we know they have no free premises. This means that, $\mathfrak{cl}(\mathcal{S}_1) = \mathfrak{cl}(\mathcal{B}_1)$ and $\mathfrak{cl}(\mathcal{S}_2) = \mathfrak{cl}(\mathcal{B}_2)$. Thus, we know that $\mathfrak{cl}(\mathcal{B}_1) \cup \mathfrak{cl}(\mathcal{B}_2) \models \bot$. However, from Proposition 7.5 we know that $\mathfrak{bd}(\mathcal{B}_1) \models \mathfrak{cl}(\mathcal{B}_1)$ and also $\mathfrak{bd}(\mathcal{B}_2) \models \mathfrak{cl}(\mathcal{B}_2)$ which therefore is sufficient to verify $\mathfrak{bd}(\mathcal{B}_1) \cup \mathfrak{bd}(\mathcal{B}_2) \models \bot$. □

Proposition 7.15. *Given a* GenAT $\tau = \langle \Sigma, \pi, \mathfrak{panf}_\pi, \Omega, \mathfrak{atom}_{[\pi,\Omega]} \rangle \in \mathbb{T}$, *the* GenAS $\mathfrak{genas}(\tau) = \langle \delta, \mathbf{S}_\delta, \mathbf{R}_\delta \rangle \in \mathbb{G}$, *and the translated logic-based AS* $\mathfrak{lbas}(\mathfrak{genas}(\tau)) = \langle Args(\Sigma), \mathbf{R} \rangle$, *if* $\mathcal{L}^\kappa$ *is a* π-pANF *logic and* $\mathcal{L}_{cl}$ *is parallel-complementary then* $\mathbf{R}$ *is conflict-exhaustive.*

Proof. From *conflict-exhaustive* (see Definition 7.11), if $X \in \mathfrak{mi}(\Sigma)$ such that $|X| > 1$ then there is X_1, $X_2 \subset X$ such that $X = X_1 \cup X_2$ and $\mathcal{B}_1, \mathcal{B}_2 \in Args(\Sigma)$ such that $\mathfrak{bd}(\mathcal{B}_1) = X_1$ and $\mathfrak{bd}(\mathcal{B}_2) = X_2$, and either $(\mathcal{B}_1, \mathcal{B}_2) \in \mathbf{R}$ or $(\mathcal{B}_2, \mathcal{B}_1) \in \mathbf{R}$. Since $X \in \mathfrak{mi}(\Sigma)$ and $\Sigma = \mathfrak{panf}_\pi(\Sigma)$ we know that $X \in \mathfrak{mi}(\mathfrak{panf}_\pi(\Sigma))$. Also, $X \vDash \bot$, and since $\mathcal{L}_{cl}$ is parallel-complementary we know that $X \vDash \vartheta$ and $X \vDash \vartheta'$, where $\vartheta, \vartheta' \in \mathcal{L}_{cl}$ such that $\{\vartheta, \vartheta'\} \vDash \bot$. Besides, we know X is minimal, hence, there are necessarily two sets X_1, $X_2 \subset X$ such that $X = X_1 \cup X_2$ and $X_1 \vDash \vartheta$ and $X_2 \vDash \vartheta'$. Afterwards, we have that there are two full arguments $\mathcal{S}_1 \in \mathbf{S}_\delta$ for ϑ and $\mathcal{S}_2 \in \mathbf{S}_\delta$ for ϑ' such that $\mathfrak{base}(\{\mathcal{S}_1\}) = X_1$ and $\mathfrak{base}(\{\mathcal{S}_2\}) = X_2$. Also, since their claims are contradictory, we know they are rebuttals, hence $(\mathcal{S}_1, \mathcal{S}_2) \in \mathbf{C}^b_\delta$. Finally, from Definition 7.4, we know $\mathfrak{lbarg}(\mathcal{S}_1) = \mathcal{B}_1$ and $\mathfrak{lbarg}(\mathcal{S}_2) = \mathcal{B}_2$, and hence from Definition 7.7 we know $(\mathcal{B}_1, \mathcal{B}_2) \in \mathbf{R}$. □

Proposition 7.21. *Given a set* $\Theta \subseteq Args(\Sigma)$, Θ *is closed iff* $\Theta = Args(\mathfrak{base}(\Theta))$.

Proof. $\Rightarrow$) We need to prove that if Θ is closed then $\Theta = Args(\mathfrak{base}(\Theta))$. Besides, from Definition 7.20 we know that ($\mathcal{B} \in \Theta$ *iff* $\mathcal{B}' \in \Theta$, for all $\mathcal{B}' \trianglelefteq \mathcal{B}$). We need to show 1) $\Theta \subseteq Args(\mathfrak{base}(\Theta))$ and 2) $Args(\mathfrak{base}(\Theta)) \subseteq \Theta$.

For 1), by *reductio ad absurdum* we assume there is an argument $\mathcal{B}' \in \Theta$ such that $\mathcal{B}' \notin Args(\mathfrak{base}(\Theta))$. The definition of a base function for logic-based arguments (see page 30) is $\mathfrak{base}(\Theta) = \{\vartheta \in \mathfrak{bd}(\mathcal{B}) | \mathcal{B} \in \Theta\}$. From Definition 7.2, we know $Args(\Sigma) = \{\mathcal{B} \in \mathbb{B} | \mathfrak{bd}(\mathcal{B}) \subseteq \Sigma\}$, and by assuming $\Sigma = \mathfrak{base}(\Theta)$ we have that $Args(\mathfrak{base}(\Theta)) = \{\mathcal{B} \in \mathbb{B} | \mathfrak{bd}(\mathcal{B}) \subseteq \mathfrak{base}(\Theta)\}$. Then, since $\mathfrak{bd}(\mathcal{B}') \subseteq \mathfrak{base}(\Theta)$ we have that $\mathcal{B}' \in Args(\mathfrak{base}(\Theta))$, which is absurd.

For 2), by *reductio ad absurdum* we assume there is an argument $\mathcal{B} \in Args(\mathfrak{base}(\Theta))$ such that $\mathcal{B} \notin \Theta$. From Definition 7.2, we know that $\mathfrak{bd}(\mathcal{B}) \subseteq \mathfrak{base}(\Theta)$. We can assume that for each formula $\vartheta \in \mathfrak{bd}(\mathcal{B})$ there is some argument $\mathcal{B}' \in \Theta$ such that $\vartheta \in \mathfrak{bd}(\mathcal{B}')$. Afterwards, we can also assume that the primitive argument $\mathcal{B}'' = \langle \{\vartheta\}, \vartheta \rangle$ is such that $\mathcal{B}'' \trianglelefteq \mathcal{B}'$, and then $\mathcal{B}'' \in \Theta$ given that Θ is closed. Since this holds for every formula $\vartheta \in \mathfrak{bd}(\mathcal{B})$, we know that all the primitive subarguments of $\mathcal{B}$ are in Θ. Given that Θ is closed, we know that $\mathcal{B} \in \Theta$ which is absurd.

$\Leftarrow$) We need to prove that if $\Theta = Args(\mathfrak{base}(\Theta))$ then Θ is closed. This means that, Θ contains all the arguments constructible from $\mathfrak{base}(\Theta)$ (see Definition 7.2). We need to show: 1) if $\mathcal{B} \in \Theta$ then $\mathcal{B}' \in \Theta$, for all $\mathcal{B}' \trianglelefteq \mathcal{B}$ and 2) if $\mathcal{B}' \in \Theta$, for all $\mathcal{B}' \trianglelefteq \mathcal{B}$, then $\mathcal{B} \in \Theta$. For 1), we know $\mathfrak{bd}(\mathcal{B}) \subseteq \mathfrak{base}(\Theta)$. Then, for all $\mathcal{B}' \trianglelefteq \mathcal{B}$ we know that $\mathfrak{bd}(\mathcal{B}') \subseteq \mathfrak{bd}(\mathcal{B})$ and thus $\mathfrak{bd}(\mathcal{B}') \subseteq \mathfrak{base}(\Theta)$, which means that $\mathcal{B}' \in \Theta$. The proof for 2) is similar. □

Lemma 7.25. *Given* $AS(\Sigma) = \langle Args(\Sigma), \mathbf{R} \rangle$, *if* $\mathfrak{s}$ *is some logic-based semantics then* $AS(\Sigma)$ *guarantees the postulates of (closure), (closure under consequence), and (exhaustiveness).*

Proof. Since $\mathfrak{s}$ is some logic-based semantics, any $\mathfrak{s}$ is based upon logic-based admissibility. This means that for every extension $\mathcal{E} \in Ext_{\mathfrak{s}}(AS(\Sigma))$ it holds that $\mathbf{E}$ is a logic-based admissible set, which implies $\mathbf{E}$ is closed, self-acceptable, and conflict-free (see Definition 7.22). Thus, (closure) is guaranteed.

We need to show (closure under consequence) for all $\mathcal{E} \in Ext_{\mathfrak{s}}(AS(\Sigma))$, if $\mathfrak{clset}(\mathcal{E}) \models \vartheta$ then $\vartheta \in \mathfrak{clset}(\mathcal{E})$. By *reductio ad absurdum*, we assume $\vartheta \notin \mathfrak{clset}(\mathcal{E})$. This means that there is some argument $\mathcal{B} \in Args(\Sigma)$ such that $\mathfrak{cl}(\mathcal{B}) = \vartheta$ but $\mathcal{B} \notin \mathcal{E}$. This is so, given that ϑ is inferred from the claims of arguments in $\mathcal{E}$, and thus from their bodies, and also given that $\vartheta \in \mathcal{L}^{\kappa}$. Thus, $\mathfrak{bd}(\mathcal{B}) \subseteq \mathcal{E}$. Since $\mathcal{E}$ is closed, from Proposition 7.21 we know that $\mathcal{E} = Args(\mathfrak{base}(\mathcal{E}))$, and therefore, $\mathcal{B} \in \mathcal{E}$ which is absurd.

We need to show (exhaustiveness) for all $\mathcal{E} \in Ext_{\mathfrak{s}}(AS(\Sigma))$ and all $\mathcal{B} \in Args(\Sigma)$, if $\mathfrak{bd}(\mathcal{B}) \cup \mathfrak{cl}(\mathcal{B}) \subseteq \mathfrak{clset}(\mathcal{E})$ then $\mathcal{B} \in \mathcal{E}$. From Proposition 7.21 we know that $\mathcal{E} = Args(\mathfrak{base}(\mathcal{E}))$, hence $\mathcal{B} \in \mathcal{E}$. □

Lemma 7.26. *Given* $AS(\Sigma) = \langle Args(\Sigma), \mathbf{R} \rangle$, *if* $\mathfrak{s}$ *is some logic-based semantics and* $\mathbf{R}$ *is conflict-exhaustive then* $AS(\Sigma)$ *guarantees the postulate of (consistency).*

Proof. Since $\mathcal{E}$ is a logic-based admissible set, we know that $\mathcal{E}$ is closed, hence from Proposition 7.21, we know $\mathcal{E} = Args(\mathfrak{base}(\mathcal{E}))$. Afterwards, from Proposition 7.19, (consistency) is guaranteed. □

Theorem 2.27. *Given a* GenAT $\tau = \langle \Sigma, \pi, \mathfrak{panf}_{\pi}, \Omega, \mathfrak{atom}_{[\pi,\Omega]} \rangle \in \mathbb{T}$ *and the* GenAS $\mathfrak{genas}(\tau) = \langle \delta, \mathbf{S}_{\delta}, \mathbf{R}_{\delta} \rangle \in \mathbb{G}$, *if* $\mathcal{L}^{\kappa}$ *is a* π-pANF *logic,* $\mathcal{L}_{\mathrm{cl}}$ *is parallel-complementary logic,* Σ *has no tautologies, the* GenAF δ *is closed under transposition,* $\Omega \subseteq \{\omega_{\pi}^{1}, \omega_{\pi}^{2}\}$ *where* ω_{π}^{1} *and* ω_{π}^{2} *are atom constraint functions modeling* $\mathbf{AC}_1$ *and* $\mathbf{AC}_2$*, respectively, and* $\mathfrak{s}$ *is a logic-based semantics then* $\mathfrak{lbas}(\mathfrak{genas}(\tau)) = \langle \mathbf{B}, \mathbf{T} \rangle$ *is a logic-based AS guaranteeing the argumentation postulates of (closure), (closure under consequence), (consistency), and (exhaustiveness).*

Proof. From Corollary 7.10 we know that $\mathfrak{lbas}(\mathfrak{genas}(\tau))$ is equivalent to $AS(\Sigma)$. Thus, from Lemma 7.25 we know $\mathfrak{lbas}(\mathfrak{genas}(\tau))$ guarantees the postulates of (closure), (closure under consequence), and (exhaustiveness). Afterwards, from Proposition 7.15 we know that $\mathbf{T}$ is conflict-exhaustive, and thus from Lemma 7.26 we know $\mathfrak{lbas}(\mathfrak{genas}(\tau))$ guarantees the postulate of (consistency). □

Theorem 2.28. *Given a* GenAT $\tau = \langle \Sigma, \pi, \mathfrak{panf}_{\pi}, \Omega, \mathfrak{atom}_{[\pi,\Omega]} \rangle \in \mathbb{T}$ *and the* GenAS $\mathfrak{genas}(\tau) = \langle \delta, \mathbf{S}_{\delta}, \mathbf{R}_{\delta} \rangle \in \mathbb{G}$, *if* $\mathcal{L}^{\kappa}$ *is a* π-pANF *logic,* $\mathcal{L}_{\mathrm{cl}}$ *is parallel-complementary logic,* Σ *has no tautologies, the* GenAF δ *is closed under transposition,* $\Omega \subseteq \{\omega_{\pi}^{1}, \omega_{\pi}^{2}\}$ *where* ω_{π}^{1} *and* ω_{π}^{2} *are atom constraint functions modeling* $\mathbf{AC}_1$ *and* $\mathbf{AC}_2$*, respectively,* $\mathfrak{s}$ *is a logic-based semantics, and* $Ext_{\mathfrak{s}}(AS(\Sigma)) \neq \emptyset$ *then* $\mathfrak{lbas}(\mathfrak{genas}(\tau)) = \langle \mathbf{B}, \mathbf{T} \rangle$ *is a logic-based AS guaranteeing the five argumentation postulates.*

Proof. From Theorem 7.27 we know $\mathfrak{lbas}(\mathfrak{genas}(\tau))$ guarantees the postulates of (closure), (closure under consequence), (consistency), and (exhaustiveness). From Proposition 7.13 we know $\mathbf{T}$ is conflict-dependent and from Proposition 7.17, we also know that $\mathfrak{lbas}(\mathfrak{genas}(\tau))$ guarantees (free precedence). Hence, the five postulates are guaranteed under the conditions stated by the hypothesis. □

References

Amgoud, L. (2014). Postulates for logic-based argumentation systems. *International Journal of Approximate Reasoning, 55*(9), 2028–2048.

Amgoud, L., & Cayrol, C. (2002). A reasoning model based on the production of acceptable arguments. *Annals of Mathematics and Artificial Intelligence, 34*(1–3), 197–215.

Amgoud, L., Cayrol, C., & Lagasquie-Schiex, M.-C. (2004). On the bipolarity in argumentation frameworks. In J. P. Delgrande, & T. Schaub (Eds.), *Nmr* (pp. 1–9).

Amgoud, L., Cayrol, C., Lagasquie-Schiex, M.-C., & Livet, P. (2008). On bipolarity in argumentation frameworks. *International Journal of Intelligent Systems, 23*(10), 1062–1093.

Amgoud, L., & Vesic, S. (2009). Repairing preference-based argumentation frameworks. In C. Boutilier (Ed.), *IJCAI 2009, Proceedings of the 21st international joint conference on artificial intelligence, Pasadena, California, USA, July 11–17, 2009* (pp. 665–670).

Baader, F. (1999). Logic-based knowledge representation. In *Artificial intelligence today: Recent trends and developments* (pp. 13–41)). Berlin, Heidelberg: Springer Berlin Heidelberg.

Baader, F. (2003). Terminological cycles in a description logic with existential restrictions. In G. Gottlob, & T. Walsh (Eds.), *Ijcai-03, proceedings of the eighteenth international joint conference on artificial intelligence, acapulco, mexico, august 9–15, 2003* (pp. 325–330). Morgan Kaufmann.

Baader, F., Brandt, S., & Lutz, C. (2005). Pushing the EL envelope. In L. P. Kaelbling, & A. Saffiotti (Eds.), *Ijcai-05, proceedings of the nineteenth international joint conference on artificial intelligence, edinburgh, scotland, uk, july 30-august 5, 2005* (pp. 364–369). Professional Book Center.

Baader, F., Calvanese, D., McGuinness, D. L., Nardi, D., & Patel-Schneider, P. F. (Eds.). (2003). *The description logic handbook: Theory, implementation, and applications*. Cambridge University Press.

Baader, F., & Peñaloza, R. (2011). GCIs make reasoning in fuzzy DL with the product T-norm undecidable. In R. Rosati, S. Rudolph, & M. Zakharyaschev (Eds.), *Proceedings of the 24th international workshop on description logics (DL 2011), barcelona, spain, july 13–16, 2011*. In *CEUR Workshop Proceedings: 745*. CEUR-WS.org.

Barker-Plummer, D., Barwise, J., & Etchemendy, J. (2011). *Language, proof and logic*. CSLI (Center for Logic and Information) University of Chicago Press, 2nd Edition.

Baroni, P., & Giacomin, M. (2007). On principle-based evaluation of extension-based argumentation semantics. *Artificial Intelligence, 171*(10–15), 675–700.

Besnard, P., & Hunter, A. (2005). Practical first-order argumentation. In M. M. Veloso, & S. Kambhampati (Eds.), *Proceedings, the twentieth national conference on artificial intelligence and the seventeenth innovative applications of artificial intelligence conference, july 9–13, 2005, pittsburgh, pennsylvania, USA* (pp. 590–595). AAAI Press / The MIT Press.

Besnard, P., & Hunter, A. (2006). Knowledgebase compilation for efficient logical argumentation. In P. Doherty, J. Mylopoulos, & C. A. Welty (Eds.), *Proceedings, Tenth International Conference on Principles of Knowledge Representation and Reasoning, Lake District of the United Kingdom, June 2–5, 2006*. AAAI Press.

Besnard, P., & Hunter, A. (2008). *Elements of argumentation*. The MIT Press.

Bex, F., Snaith, M., Lawrence, J., & Reed, C. (2014). Argublogging: An application for the argument web. *Web Semantics: Science, Services and Agents on the World Wide Web, 25*, 9–15.

Bienvenu, M. (2008). Prime implicate normal form for ALC concepts. In D. Fox, & C. P. Gomes (Eds.), *Proceedings of the twenty-third AAAI conference on artificial intelligence, AAAI 2008, chicago, illinois, usa, july 13–17, 2008* (pp. 412–417). AAAI Press.

Bienvenu, M. (2009). Prime implicates and prime implicants: From propositional to modal logic. *Journal of Artificial Intelligence Research (JAIR), 36*, 71–128.

Bochman, A. (2003). Collective argumentation and disjunctive logic programming. *Journal of Logic and Computation, 13*(3), 405–428.

Bondarenko, A., Dung, P. M., Kowalski, R. A., & Toni, F. (1997). An abstract, argumentation-theoretic approach to default reasoning. *Artificial Intelligence, 93*, 63–101.

Bondarenko, A., Toni, F., & Kowalski, R. A. (1993). An assumption-based framework for non-monotonic reasoning. In *LPNMR* (pp. 171–189).

Borgida, A. (1996). On the relative expressiveness of description logics and predicate logics. *Artificial Intelligence, 82*(1–2), 353–367.

Brandt, S. (2004). Polynomial time reasoning in a description logic with existential restrictions, GCI axioms, and- what else?. In R. L. de Mántaras, & L. Saitta (Eds.), *Proceedings of the 16th eureopean conference on artificial intelligence, ecai'2004, including prestigious applicants of intelligent systems, PAIS 2004, valencia, spain, august 22–27, 2004* (pp. 298–302). IOS Press.

Brewka, G., Strass, H., Ellmauthaler, S., Wallner, J. P., & Woltran, S. (2013). Abstract dialectical frameworks revisited. In F. Rossi (Ed.), *IJCAI 2013, proceedings of the 23rd international joint conference on artificial intelligence, beijing, china* (pp. 803–809). IJCAI/AAAI.

Brewka, G., & Woltran, S. (2010). Abstract dialectical frameworks. In F. Lin, U. Sattler, & M. Truszczynski (Eds.), *Principles of knowledge representation and reasoning: Proceedings of the twelfth international conference, KR 2010* (pp. 102–111). Toronto, Ontario, Canada: AAAI Press.

Calì, A., Gottlob, G., & Kifer, M. (2013). Taming the infinite chase: Query answering under expressive relational constraints. *Journal of Artificial Intelligence Research (JAIR), 48*, 115–174.

Calvanese, D., De Giacomo, G., Lembo, D., Lenzerini, M., & Rosati, R. (2007). Tractable reasoning and efficient query answering in description logics: The *DL-Lite* family. *Journal of Automated Reasoning, 39*(3), 385–429.

ten Cate, B., Conradie, W., Marx, M., & Venema, Y. (2006). Definitorially complete description logics. In P. Doherty, J. Mylopoulos, & C. A. Welty (Eds.), *Proceedings, Tenth International Conference on Principles of Knowledge Representation and Reasoning, Lake District of the United Kingdom, June 2–5, 2006*. AAAI Press.

Cayrol, C., & Lagasquie-Schiex, M.-C. (2005). On the acceptability of aarguments in bipolar argumentation frameworks. In L. Godo (Ed.), *Ecsqaru*. In *Lecture Notes in Computer Science: 3571* (pp. 378–389). Springer.

Cayrol, C., & Lagasquie-Schiex, M.-C. (2010). Coalitions of arguments: A tool for handling bipolar argumentation frameworks. *International Journal of Intelligent Systems, 25*(1), 83–109.

Chomicki, J. (2006). Consistent query answering: Five easy pieces. In *Database theory – ICDT 2007: 11th international conference, Barcelona, Spain, January 10–12, 2007. Proceedings* (pp. 1–17)). Berlin, Heidelberg: Springer Berlin Heidelberg.

Darwiche, A., & Marquis, P. (2002). A Knowledge Compilation Map. *Journal of Artificial Intelligence Research (JAIR), 17*, 229–264.

Dimopoulos, Y., & Torres, A. (1996). Graph Theoretical Structures in Logic Programs and Default Theories. *Theoretical Computer Science, 170*(1), 209–244.

Dung, P. M. (1995). On the acceptability of arguments and its fundamental role in nonmonotonic reasoning and logic programming and *n*-person games. *Artificial Intelligence, 77*, 321–357.

Dung, P. M., & Thang, P. M. (2014). Closure and consistency in logic-associated argumentation. *Journal of Artificial Intelligence Research (JAIR), 49*, 79–109.

Flouris, G., Huang, Z., Pan, J. Z., Plexousakis, D., & Wache, H. (2006). Inconsistencies, Negations and Changes in Ontologies. In *Proceedings, the twenty-first national conference on artificial intelligence and the eighteenth innovative applications of artificial intelligence conference, july 16–20, 2006, boston, massachusetts, USA* (pp. 1295–1300). AAAI Press.

Furbach, U., Günther, H., & Obermaier, C. (2009). A Knowledge compilation technique for ALC Tboxes. In H. C. Lane, & H. W. Guesgen (Eds.), *Proceedings of the twenty-second international florida artificial intelligence research society conference, may 19–21, 2009, sanibel island, florida, USA*. AAAI Press.

Furbach, U., & Obermaier, C. (2007). Knowledge compilation for description logics. In J. Baumeister, & D. Seipel (Eds.), *Kese*. In *CEUR Workshop Proceedings: 282*. CEUR-WS.org.

García, A., & Simari, G. (2004). Defeasible logic programming: An argumentative approach. *TPLP, 4*(1–2), 95–138.

Goncalves, E., & Grädel, E. (2000). Decidability Issues for Action Guarded Logics. In *In proceedings of 2000 international workshop on description logics - dl2000* (pp. 123–132).

Gorogiannis, N., & Hunter, A. (2011). Instantiating Abstract Argumentation with Classical Logic Arguments: Postulates and Properties. *Artificial Intelligence, 175*(9–10), 1479–1497.

Lembo, D., Lenzerini, M., Rosati, R., Ruzzi, M., & Savo, D. F. (2010). Inconsistency-tolerant semantics for description logics. In *Web Reasoning and Rule Systems: Fourth International Conference, RR 2010, Bressanone/Brixen, Italy, September 22–24, 2010. Proceedings* (pp. 103–117)). Berlin, Heidelberg: Springer Berlin Heidelberg.

Lembo, D., & Ruzzi, M. (2007). Consistent query answering over description logic ontologies. In *Web Reasoning and Rule Systems: First International Conference, RR 2007, Innsbruck , Austria, June 7–8, 2007. Proceedings* (pp. 194–208)). Berlin, Heidelberg: Springer Berlin Heidelberg.

Lutz, C., Sattler, U., & Wolter, F. (2001). Description logics and the two-variable fragment. In C. A. Goble, D. L. McGuinness, R. Möller, & P. F. Patel-Schneider (Eds.), *Working notes of the 2001 international description logics workshop (dl-2001), stanford, ca, usa, august 1–3, 2001*. In *CEUR Workshop Proceedings: 49*. CEUR-WS.org.

Martínez, D. C., García, A. J., & Simari, G. R. (2007). Modelling Well-Structured Argumentation Lines. In M. M. Veloso (Ed.), *IJCAI 2007, proceedings of the 20th international joint conference on artificial intelligence, hyderabad, india, january 6–12, 2007* (pp. 465–470).

Moguillansky, M. O., Rotstein, N. D., & Falappa, M. A. (2010a). Generalized abstract argumentation: A first-order machinery towards ontology debugging. *Inteligencia Artificial, Revista Iberoamericana de Inteligencia Artificial, 14*(46), 17–33.

Moguillansky, M. O., Rotstein, N. D., Falappa, M. A., & Simari, G. R. (2009). Generalized abstract argumentation: Handling arguments in FOL fragments. In C. Sossai, & G. Chemello (Eds.), *Ecsqaru*. In *Lecture Notes in Computer Science: 5590* (pp. 144–155). Springer.

Moguillansky, M. O., Wassermann, R., & Falappa, M. A. (2010b). An argumentation machinery to reason over inconsistent ontologies. In Á. F. K. Morales, & G. R. Simari (Eds.), *Iberamia*. In *Lecture Notes in Computer Science: 6433* (pp. 100–109). Springer.

Moguillansky, M. O., Wassermann, R., & Falappa, M. A. (2012). Inconsistent-tolerant base revision through argument theory change. *Logic Journal of the IGPL, 20*(1), 154–186.

Mortimer, M. (1975). On languages with two variables. *Zeitschr. f. math. Logik und Grundlagen d. Math., 21*, 135–140.

Nielsen, S. H., & Parsons, S. (2006). A generalization of Dung's abstract framework for argumentation: Arguing with sets of attacking arguments. In N. Maudet, S. Parsons, & I. Rahwan (Eds.), *Argmas*. In *Lecture Notes in Computer Science: 4766* (pp. 54–73). Springer.

Poole, D. (1988). A logical framework for default reasoning. *Artificial Intelligence, 36*(1), 27–47.

Prakken, H. (2010). An abstract framework for argumentation with structured arguments. *Argument and Computation, 1*(2), 93–124.

Prawitz, D. (1965). Natural deduction: A proof-theoretical study. *Acta Universitatis Stockholmiensis, Stockholm studies in philosophy 3*. Stockholm: Almqvist & Wicksell.

Rahwan, I., & Banihashemi, B. (2008). Arguments in OWL: A progress report. In P. Besnard, S. Doutre, & A. Hunter (Eds.), *Computational models of argument: Proceedings of COMMA 2008, toulouse, france, may 28–30, 2008.*. In *Frontiers in Artificial Intelligence and Applications: 172* (pp. 297–310). IOS Press.

Rosati, R. (2011). On the complexity of dealing with inconsistency in description logic ontologies. In T. Walsh (Ed.), *IJCAI 2011, proceedings of the 22nd international joint conference on artificial intelligence, barcelona, catalonia, spain, july 16–22, 2011* (pp. 1057–1062). IJCAI/AAAI.

Rotstein, N. D., Moguillansky, M. O., García, A. J., & Simari, G. R. (2010). A Dynamic argumentation framework. In P. Baroni, F. Cerutti, M. Giacomin, & G. R. Simari (Eds.), *Comma* (pp. 427–438). Amsterdam, The Netherlands, The Netherlands: IOS Press.

Schlobach, S., Huang, Z., Cornet, R., & van Harmelen, F. (2007). Debugging incoherent terminologies. *Journal of Automated Reasoning, 39*(3), 317–349.

Thimm, M., Villata, S., Cerutti, F., Oren, N., Strass, H., & Vallati, M. (2016). Summary report of the first international competition on computational models of argumentation. *AI Magazine*.

Vreeswijk, G. (1997). Abstract argumentation systems. *Artificial Intelligence, 90*(1–2), 225–279.

Walton, D. N. (2002). *Legal argumentation and evidence*. Pennsylvania State Univ. Pr.