(Preliminary Manuscript)

To appear in "Logica Universalis", Birkhäuser, Springer - Jean-Yves Beziau (editor), 2006

Modelling Inference in Argumentation through Labelled Deduction: Formalization and Logical Properties

Carlos Iván Chesñevar and Guillermo Ricardo Simari

Abstract. Artificial Intelligence (AI) has long dealt with the issue of finding a suitable formalization for commonsense reasoning. Defeasible argumentation has proven to be a successful approach in many respects, proving to be a confluence point for many alternative logical frameworks. Different formalisms have been developed, most of them sharing the common notions of *argument* and *warrant*. In defeasible argumentation, an *argument* is a tentative (defeasible) proof for reaching a conclusion. An argument is *warranted* when it ultimately prevails over other conflicting arguments. In this context, defeasible consequence relationships for modelling argument and warrant as well as their logical properties have gained particular attention.

This article analyzes two non-monotonic inference operators C_{arg} and C_{war} intended for modelling argument construction and dialectical analysis (warrant), respectively. As a basis for such analysis we will use the LDS_{ar} framework, a unifying approach to computational models of argument using Labelled Deductive Systems (LDS). In the context of this logical framework, we show how labels can be used to represent arguments as well as argument trees, facilitating the definition and study of non-monotonic inference operators, whose associated logical properties are studied and contrasted. We contend that this analysis provides useful comparison criteria that can be extended and applied to other argumentation frameworks.

Mathematics Subject Classification (2000). Primary 03B22; Secondary 03B42.

Keywords. Defeasible Argumentation, Knowledge Representation, Nonmonotonic Inference, Labelled Deduction.

This article is based on the paper *"Consequence operators for Defeasible Argumentation: characterization and properties"* (C. Chesñevar and G. Simari). Proc. of the VII Argentinean Conference in Computer Science, pp. 309-320 (ISBN 987-96 288-6-1). El Calafate, Argentina, October 2001. Other papers of the authors were used as a source to make this article self-contained. Thus, the definition of LDS_{ar} presented in Section 3 is based on [9, 7], and the discussion on the different applications of labelled systems for argumentation presented in Section 6 is based on [10].

1. Introduction and Motivations

Artificial Intelligence (AI) has long dealt with the issue of finding a suitable formalization for commonsense reasoning. Defeasible argumentation has proven to be a successful approach in many respects, proving to be a confluence point for many alternative logical frameworks. Different argument-based formalisms have been developed, most of them sharing the common notions of *argument* and *warrant*. In defeasible argumentation, an *argument* is a tentative (defeasible) proof for reaching a conclusion. An argument is *warranted* when it ultimately prevails over other conflicting arguments. In this context, defeasible *inference relationships* for modelling argument and warrant as well as their logical properties have gained particular attention.[1] As highlighted in [35, 8] several common elements can be identified in such frameworks: an underlying logical language, the concept of argument, etc. However, these elements appear along with a number of particular features which make it difficult to compare such frameworks with each other from a logical viewpoint. In this context, the search for a general framework for defeasible argumentation in which such common elements could be abstracted away led to the development of of LDS_{ar} [7], an argumentation formalism based on the *labelled deduction* methodology [15].

Labelled Deductive Systems (LDS) [15, 16] were developed as a rigorous but flexible methodology to formalize complex systems as logical frameworks with labelled deduction capabilities (*e.g.*, temporal logics, database query languages and defeasible reasoning systems). In labelled deduction, the usual notion of formula is replaced by the notion of *labelled formula*, expressed as *Label*:*f*, where *Label* represents a label associated with the wff *f*. A labelling language $\mathcal{L}_{Label}$ and knowledge-representation language $\mathcal{L}_{kr}$ can be combined to provide an enriched object language, in which labels convey additional information also encoded at object-language level.

The application of Labelled Deductive Systems to the formal analysis of Defeasible Argumentation establishes a bridge from the logical perspective on deduction to the procedure-oriented construction of arguments. In that manner, LDS constitutes a formalism within the set of tools characterizing the *Universal Logic* approach [5] which offers ways of studying the mechanisms by which arguments are constructed and their validity is studied. The precise definitions of the basic elements in Argumentation obtained inside the LDS framework helps in the comparison of the different argumentative systems created in the last three decades thus bringing the possibility of finding commonalities and differences in such developments. With that aim, this article analyzes two non-monotonic inference operators C_{arg} and C_{war} intended for modelling argument construction and dialectical analysis (warrant), respectively. As a basis for such analysis we will use the LDS_{ar} framework, a unifying approach to computational models of argument

[1]We are aware that the term *non-monotonic consequence relationship* could be also used in this context. However, as usually consequence relationships are associated with different (monotonic) variants of classical logics, we prefer to use the term *inference relationship* instead.

using Labelled Deductive Systems (LDS). In the context of this logical framework, we show how labels can be used to represent arguments as well as argument trees, facilitating the definition and study of consequence operators, whose associated logical properties are studied and contrasted with SLD-based Horn logic. We contend that this analysis provides useful comparison criteria that can be extended and applied to other argumentation frameworks.

The rest of the article is structured as follows: Section 2 introduces some fundamentals of Labelled Deductive Systems. Section 3 presents the main elements of the LDS_{ar} framework [9], along with a worked example. Section 4 presents an overview of the basic notions concerning consequence operators, non-monotonic inference and their properties. Section 5 introduces two non-monotonic inference operators C_{arg} and C_{war}, used for computing arguments and warranted conclusions in the context of the proposed framework. Logical properties that characterize the behavior of these operators are discussed and contrasted. Section 6 analyzes some relevant aspects of of defeasible argumentation that can be analyzed on the basis of the proposed framework. Section 7 discusses related work. Finally, Section 8 concludes and presents some research lines for future work.

2. Labelled Deductive Systems: Fundamentals

Logic has been traditionally perceived as the study of 'consequence relations' between sets of formulæ. The complexity of real-world problems and the associated formalizations in different application areas (such as Artificial Intelligence, Cognitive Science and Computer Science) have resulted in a plethora of new logical systems, created to give an account to the variety of reasonings required in such areas. This situation prompted the development of *Universal Logic*,[2] which is not a new logic but a general theory of logics, where logics are considered as mathematical structures [5]. General tools started being developed for a systematic study of this huge amount of new logics. In this context Dov Gabbay focused on the issue of how to formally define the differences existing among logics, (which involves characterizing a way of comparing them). According to Gabbay [15], the answers to these considerations are to be found in metalevel considerations, which can be better identified by analyzing those aspects which are *uniform* in most logics (*e.g.*, the structure of inference rules, rules for quantifiers, etc.).

To this end Gabbay developed *Labelled Deductive Systems* (LDS) [15], a unifying framework for the study of logics and of their interactions. LDS aim to characterize a logical system by 'abstracting away' the common aspects mentioned before. As a first approximation, a labelled deductive system is a 3-uple $(\mathbf{A}, \mathcal{L}, M)$, where $\mathcal{L}$ is a *logical language* (including connectives and wffs), $\mathbf{A}$ is an *algebra* on labels (with given operations), and M is a *discipline* which indicates how to label formulas in the logic, given the algebra $\mathbf{A}$ of labels [15]. Such a discipline will be formulated using deduction rules. In order to characterize an LDS, a labelled

[2] This term was coined in [4], as an analogy with the concept of *universal algebra* in mathematics.

language must be defined including wffs and labels. In such language labels can be seen as carriers of information which is not present in the wffs themselves.

Why are labels needed? In LDS labels will be used to store information of different sort of the one encoded in the predicate associated with it. There may exist different reasons for doing this: it can be the case that the information on the label is of a different nature or purpose than the one coded in the main predicate, and therefore it is more convenient to keep it as an annotation or label; or it may also be the case that the manipulation of this extra information is too complex, and so we want to keep it apart from the predicate associated with it. Instead of referring to a formula A, the name of *declarative unit* is generically used to refer to labelled formulæ $t : A$. As Gabbay remarks (*op. cit.*), there may be many uses for a label t in such a declarative unit $t : A$. The value t might correspond to a confidence value in fuzzy logic (*e.g.*, t could be a real number between 0 and 1), an indicator of the origin of the wff A (*e.g.*, in a very complex database), or an annotation of the proof of A (*e.g.*, t can include the set of assumptions that lead to believe A).

There are several additional constraints imposed on the way we use LDS. The main ones are the following: a) the only inference rules allowed are the traditional ones, modus ponens and some form of introduction rule (deduction theorem) for implication, for example; b) allowable modes of label propagation are fixed for all logics. They can be adjusted in agreed ways to obtain variations, but in general the format is the same. This also applies to quantifiers (the quantifier rules are the same for all logics); and c) Metalevel features are implemented via the labelling mechanism, which is the object language. Finally, whereas in traditional logical systems the consequence is defined using proof rules on formulas, in the LDS methodology the consequence is defined by using rules on *both* formulæ and labels. Thus the traditional notion of consequence between formulæ of the form $A_1, \ldots, A_n \vdash B$ is replaced by the notion of consequence between labelled formulæ $t_1{:}A_1;\ t_2{:}A_2;\ \ldots;\ t_n{:}A_n \mid\!\sim\ s{:}B$. Accordingly, we will have formal rules for manipulating labels and this will allow for more scope and detailed analysis when decomposing the various features of the consequence relation. The meta features can be reflected in the algebra or logic of the labels, and the object features in the rules of the formulas. An in-depth discussion on LDS is outside the scope of this paper, and for further details the reader is referred to [15, 16].

3. Modelling Argumentation with LDS: the LDS$_{ar}$ Framework

Argumentation frameworks[3] are characterized by representing certain features of informal argumentation using a formal language, along with an inference mechanism. Although these frameworks differ in their aims and characterization, the notion of *argument* is quite similar, having a strong resemblance to the notion of

[3]See [35, 8] for a detailed description of relevant logic-based approaches to argumentation.

proof in logic. In fact, the difference between arguments and logical proofs is more 'pragmatic' than 'syntactic' [26].

Prakken & Vreeswijk [35] have defined a conceptual framework in which most argumentation systems can be characterized. This conceptual framework involves five elements, namely:

a) an underlying logical language $\mathcal{L}$;
b) a concept of *argument*;
c) a concept of *conflict* among arguments;
d) a notion of *defeat* among arguments;
e) a notion of *acceptability* of arguments according to a well-defined criterion.

We contend that the above elements can be embedded as different parts of an LDS-based framework for argumentation called LDS_{ar} [9], whose salient features will be summarized in this section. In our approach the underlying logical language $\mathcal{L}$ will be a labelled language $\mathcal{L}_{\mathsf{Arg}} = (\mathcal{L}_{\mathsf{Labels}}, \mathcal{L}_{\mathsf{KR}})$, where $\mathcal{L}_{\mathsf{Labels}}$ is a labelling language (representing epistemic status of knowledge, as well as arguments and their interrelationships) and $\mathcal{L}_{\mathsf{KR}}$ represents object-level knowledge. Thus, the labelling language $\mathcal{L}_{\mathsf{Labels}}$ will encode different information features which correspond to the elements (a)–(e) in Prakken & Vreeswijk's conceptualization.

Usually $\mathcal{L}_{\mathsf{KR}}$ will be a distinguished subset of FOL (*e.g.*, the language of Horn clauses or the language of extended logic programming), together with the symbol "$\sim$" to denote *strict negation* [27]. For practical purposes, $\mathcal{L}_{\mathsf{KR}}$ will usually be restricted to *rules* and *facts*, in which the notion of *contradictory information* can be expressed in terms of complementary literals p and $\sim p$. We will also assume an underlying inference procedure $\vdash$ associated with $\mathcal{L}_{\mathsf{KR}}$ (*e.g.*, SLD derivation). Given a set $P \subseteq \mathsf{Wffs}(\mathcal{L}_{\mathsf{KR}})$, and $\phi \in \mathsf{Wffs}(\mathcal{L}_{\mathsf{KR}})$, we will write that $P \vdash \phi$ to denote that ϕ follows from P via $\vdash$. If two complementary literals can be derived from P via $\vdash$ we will say that P is *contradictory*, or just write $P \vdash \bot$. Following [15], labelled wffs in $\mathcal{L}_{\mathsf{Arg}}$ will be called *declarative units*, having the form *Label:wff*.

Definition 3.1 (Labelling Language $\mathcal{L}_{\mathsf{Labels}}$). The labelling language $\mathcal{L}_{\mathsf{Labels}}$ is a set of labels $\{\ L_1,\ L_2,\ \ldots L_k,\ \ldots\}$, such that every label $L \in \mathcal{L}_{\mathsf{Labels}}$ is:

1. The empty set $\emptyset$, or any $\phi \in \mathsf{Wffs}(\mathcal{L}_{\mathsf{KR}})$. These labels are called *epistemic labels*.
2. A set $\Phi \subseteq \mathsf{Wffs}(\mathcal{L}_{\mathsf{KR}})$. This is a label called *argument label*.
3. A functor $\mathbf{T}$ is a label called *dialectical label*, defined as follows:
   (a) If Φ is an argument label, then $\mathbf{T}^U(\Phi)$, $\mathbf{T}^D(\Phi)$ and $\mathbf{T}^*(\Phi)$ are *dialectical labels* in $\mathcal{L}_{\mathsf{Labels}}$.[4]
   (b) If $\mathbf{T}_1, \ldots, \mathbf{T}_k$ are dialectical labels, then $\mathbf{T}_n^U(\mathbf{T}_1, \ldots, \mathbf{T}_k)$, $\mathbf{T}_n^*(\mathbf{T}_1, \ldots, \mathbf{T}_k)$ and $\mathbf{T}_m^D(\mathbf{T}_1, \ldots, \mathbf{T}_k)$ will also be dialectical labels in $\mathcal{L}_{\mathsf{Labels}}$.
4. Nothing else is a label in $\mathcal{L}_{\mathsf{Labels}}$.

---

[4]For the sake of simplicity, we will just write $\mathbf{T}_1$, $\mathbf{T}_2$, etc. to denote arbitrary dialectical labels.

Next we introduce the notion of *argumentative theory*, which will be a set of 'basic' declarative units (bdu's) in our labelled language $\mathcal{L}_{\mathsf{Arg}}$. Such bdu's will be used to encode defeasible and non-defeasible information.

Definition 3.2 (Argumentative Theory). A labelled formula $\phi{:}\alpha \in \mathsf{Wffs}(\mathcal{L})$ such that $\alpha \in \mathsf{Wffs}(\mathcal{L}_{\mathsf{KR}})$ and either (1) $\phi = \emptyset$ or (2) $\phi = \{\alpha\}$ will be called a *basic declarative unit* (bdu). Cases (1) and (2) correspond to representing non-defeasible and defeasible knowledge, resp. A finite set $\Gamma = \{\phi_1{:}\alpha_1, \ldots \phi_k{:}\alpha_k\}$ where every $\phi_i{:}\alpha_i$ is a bdu will be called an *argumentative theory.* We will assume that the set $Strict(\Gamma) = \{\emptyset{:}\alpha_i \mid \emptyset{:}\alpha_i \in \Gamma\}$ is non-contradictory wrt $\vdash$.

3.1. Argument Construction

Given an argumentative theory Γ and a wff $\phi \in \mathcal{L}_{\mathsf{KR}}$, we will provide a labelled inference relationship "$\mathrel{|\!\!\sim}_{Arg}$" to characterize the notion of argument. Our labelled inference relationship "$\mathrel{|\!\!\sim}_{Arg}$" will be characterized by a number of suitable deduction rules $\mathsf{Intro\text{-}NR}$, $\mathsf{Intro\text{-}RE}$, $\mathsf{Intro\text{-}}\wedge$ and $\mathsf{Elim\text{-}}\leftarrow$(Figure 1). Rules $\mathsf{Intro\text{-}NR}$ and $\mathsf{Intro\text{-}RE}$ allow the introduction of non-defeasible and defeasible information when constructing arguments. Rules $\mathsf{Intro\text{-}}\wedge$ and $\mathsf{Elim\text{-}}\leftarrow$ stand for introducing conjunction and applying modus ponens. Note that in the last three rules a 'consistency check' wrt $\vdash$ is performed, in order to ensure that the label associated with the inferred formula does not allow the derivation of complementary literals. Note also that the label $\mathcal{A}$ associated with a formula $\mathcal{A}{:}\alpha$ contains all *defeasible* information needed to conclude α from Γ.

Definition 3.3 (Argument). Let Γ be an argumentative theory, let α be a literal in $\mathcal{L}_{\mathsf{KR}}$ and let $\mathcal{A} \subseteq \mathsf{Wffs}(\mathcal{L}_{\mathsf{KR}})$ such that $\Gamma \mathrel{|\!\!\sim}_{Arg} \mathcal{A}{:}\alpha$. Then $\mathcal{A}{:}\alpha$ will be called an *argument* on the basis of Γ. An argument $\mathcal{A}{:}h$ is a *subargument* of another argument $\mathcal{B}{:}q$ if $\mathcal{A} \subset \mathcal{B}$. We will write $\mathsf{Args}(\Gamma)$ to denote the set of all possible arguments that can be obtained from Γ.

3.2. Attack among Arguments. Dialectical Analysis

Clearly, given an argumentative theory Γ there may exist conflicting arguments (*e.g.*, $\mathcal{A}{:}\alpha$ and $\mathcal{B}{:}\overline{\alpha}$) emerging from it. We will assume that *conflict* (also *counterargument* or *attack*) among arguments is captured using the notion of contradiction associated with the $\vdash$ inference relationship used for argument construction. Note that our notion of conflict is intentionally generic, as different, more concrete formalizations are possible.

Definition 3.4 (Counterargument). Let Γ be an argumentative theory, and let $\mathcal{A}{:}h$ and $\mathcal{B}{:}q$ be arguments based on Γ. Then $\mathcal{A}{:}h$ *counter-argues* $\mathcal{B}{:}q$ if there exists a subargument $\mathcal{B}'{:}s$ of $\mathcal{B}{:}q$ such that $Strict(\Gamma) \cup \{h, s\}$ is contradictory. The argument $\mathcal{B}'{:}s$ will be called *disagreement subargument.*

Defeat among arguments involves a *preference criterion* among conflicting arguments. If Γ is an argumentative theory, a *preference order* $\preceq$ is any partial order defined on the arguments in $\mathsf{Args}(\Gamma)$. Particular approaches in argumentation

$$\text{Intro-NR:} \quad \frac{}{\emptyset{:}\alpha} \quad \text{for any } \emptyset{:}\alpha$$

$$\text{Intro-RE:} \quad \frac{}{\Phi{:}\alpha} \quad \text{for any } \Phi{:}\alpha \text{ such that } \mathit{Strict}(\Gamma) \cup \Phi \not\vdash \bot$$

$$\text{Intro-}\wedge\text{:} \quad \frac{\Phi_1{:}\alpha_1 \qquad \Phi_2{:}\alpha_2 \qquad \ldots \Phi_k{:}\alpha_k}{\cup_{i=1\ldots k}\Phi_i{:}\alpha_1, \alpha_2, \ldots, \alpha_k} \quad \text{whenever } \mathit{Strict}(\Gamma) \cup \bigcup_{i=1\ldots k} \Phi_i \not\vdash \bot$$

$$\text{Elim-}\leftarrow\text{:} \quad \frac{\Phi_1{:}\beta{\leftarrow}\alpha_1, \ldots, \alpha_k \qquad \Phi_2{:}\alpha_1, \ldots, \alpha_k}{\Phi_1 \cup \Phi_2{:}\beta} \quad \text{whenever } \mathit{Strict}(\Gamma) \cup \Phi_1 \cup \ \Phi_2 \not\vdash \bot$$

FIGURE 1. Inference rules for argument construction

frameworks to characterizing defeat may differ: some argumentation frameworks will only consider attack relationships [13], others will distinguish between rebutting and undercutting attacks [34], etc.

Definition 3.5 (Defeat). Let Γ be an argumentative theory, such that $\Gamma \mid\!\sim_{Arg} \mathcal{A}{:}h$ and $\Gamma \mid\!\sim_{Arg} \mathcal{B}{:}q$. We will say that $\mathcal{A}{:}h$ *defeats* $\mathcal{B}{:}q$ (or equivalently $\mathcal{A}{:}h$ is a *defeater* for $\mathcal{B}{:}q$) if

1. $\mathcal{A}{:}h$ counterargues $\mathcal{B}{:}q$, with disagreement subargument $\mathcal{B}'{:}q'$.
2. Either it holds that $\mathcal{A}{:}h \succ \mathcal{B}'{:}q'$, or $\mathcal{A}{:}h$ and $\mathcal{B}'{:}q'$ are unrelated by the preference order "$\preceq$".

We will not delve into such differences here, but will rather focus on capturing the notion of dialectical analysis in terms of natural deduction rules. A usual approach involves computing (explicitly or implicitly) a so-called *dialectical tree*.[5] A dialectical tree is a dialogue tree between two parties, proponent and opponent. Branches of the tree correspond to *all* possible dialogues or exchanges of arguments between these two parties, starting from the initial argument at issue (root node). A dialectical tree can be marked as an AND-OR tree according to the following procedure: nodes with no defeaters (leaves) are marked as U-nodes (undefeated nodes). Inner nodes are marked as D-nodes (defeated nodes) iff they have at least one U-node as a child, and as U-nodes iff they have every child marked as a D-node (as shown in Figure 3). In the case that the root node of the tree is finally marked as U-node, then the associated argument is said to be *justified* or *warranted*.[6]

[5] Also called "argument tree", "dialogue tree" or "dispute tree" in the literature (see e.g. [34, 40, 3]).

[6] Both terms are commonly used in the literature [35, 8].

Intro-1D: $$\frac{\mathcal{A}{:}\alpha}{\mathbf{T}^*(\mathcal{A}){:}\alpha}$$
whenever $\mathcal{A}$ is minimal wrt set inclusion

Intro-ND: $$\frac{\mathbf{T}^*(\mathcal{A}){:}\alpha \quad \mathbf{T}_1^*(\mathcal{B}_1,\ldots){:}\beta_1 \quad \mathbf{T}_k^*(\mathcal{B}_k,\ldots){:}\beta_k}{\mathbf{T}^*(\mathcal{A},\mathbf{T}_1^*,\ldots,\mathbf{T}_k^*){:}\alpha}$$
whenever $\mathsf{VSTree}(\mathcal{A},\, \mathbf{T}_i^*)$ holds, $i = 1\ldots k$.

Mark-Atom: $$\frac{\mathbf{T}^*(\mathcal{A}){:}\alpha}{\mathbf{T}^U(\mathcal{A}){:}\alpha}$$

Mark-1D: $$\frac{\begin{array}{c}\mathbf{T}^*(\mathcal{A},\mathbf{T}_1^*,\ldots,\mathbf{T}_i^*,\ldots,\mathbf{T}_k){:}\alpha \\ \mathbf{T}_i^U(\mathcal{B}_i\ldots){:}\beta_i\end{array}}{\mathbf{T}^D(\mathcal{A},\mathbf{T}_1^*,\ldots,\mathbf{T}_{i-1}^*,\mathbf{T}_i^U,\mathbf{T}_{i+1}^*,\ldots,\mathbf{T}_k^*){:}\alpha}$$
whenever $\mathsf{VSTree}(\mathcal{A}{,}\mathbf{T}_j^U)$ holds, for some $i \in \{1,\ldots,k\}$

Mark-ND: $$\frac{\begin{array}{c}\mathbf{T}^*(\mathcal{A},\mathbf{T}_1^*,\ldots,\mathbf{T}_i^*,\ldots,\mathbf{T}_k^*){:}\alpha \\ \mathbf{T}_1^D(\mathcal{B}_1,\ldots){:}\beta_1 \ \ldots\ \mathbf{T}_k^D(\mathcal{B}_k,\ldots){:}\beta_k\end{array}}{\mathbf{T}^U(\mathcal{A},\mathbf{T}_1^D,\ldots,\mathbf{T}_i^D,\ldots,\mathbf{T}_k^D){:}\alpha}$$

whenever $\mathsf{VSTree}(\mathcal{A}{,}\mathbf{T}_i^D)$, $\forall i \in \{1,\ldots,k\}$

FIGURE 2. Rules for dialectical analysis characterizing the inference relation $\mathrel{|\!\!\!\sim}_{\mathcal{T}}$

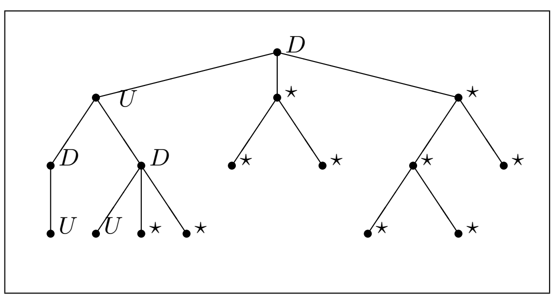

FIGURE 3. Labelling a dialectical tree with $\alpha - \beta$ pruning

In the context of LDS_{ar}, the construction and marking of dialectical trees is captured in terms of *dialectical labels* (Def. 3.1). Special marks (*, U, D) are associated with a label $\mathbf{T}(\mathcal{A},\ldots)$ in order to determine whether $\mathcal{A}$ corresponds to an argument which has been (a) *not analyzed yet* $(*)$ in the dialectical context given by the label; or (b) *defeated* (D) (resp. *undefeated* (U)) in such context. In LDS_{ar}, the construction of dialectical trees is formalized in terms of an inference relationship $\mathrel{|\!\!\sim_{\mathcal{T}}}$ given by the natural deduction rules shown in Figure 2. Rule Intro-1D allows to generate a tree with a single argument.[7] Rule Intro-ND allows to expand a given tree $\mathbf{T}^*$ by introducing new subtrees $\mathbf{T}_1^*(\mathcal{B}_1,\ldots){:}q_1 \quad \mathbf{T}_k^*(\mathcal{B}_k,\ldots){:}q_k$.

It must be remarked that a special condition $\mathsf{VSTree}(\mathcal{A},\ \mathbf{T}_i^*)$, $i = 1\ldots k$ checks that such subtrees are valid. Such checking involves several considerations, such as determining that the root of every $\mathbf{T}_i^*$ is a defeater for the root of $\mathbf{T}^*$, and no *fallacious argumentation* is present by appending any $\mathbf{T}_i^*$ as a subtree rooted in $\mathcal{A}$ [39]. An in-depth discussion of such fallacies is outside the scope of this paper, and details can be found elsewhere [9]. Rules Mark-Atom, Mark-1D and Mark-ND allow to 'mark' the nodes (arguments) in a dialectical tree as defeated or undefeated. Note that the rules propagate marking from the bottom of the tree up to the root node, according to the marking criterion discussed before. A labelled formula $\mathbf{T}_i^U(\mathcal{A},...){:}\alpha$ stands for an argument $\mathcal{A}{:}\alpha$ which has been marked as U-node according to the arguments introduced in the subtrees associated with the dialectical label $\mathbf{T}_i$. If the dialectical label $\mathbf{T}_i$ cannot be further "expanded" (see discussion below), then it corresponds to an exhaustive analysis for the argument associated with the root of that tree. This allows to formalize the notion of warrant within the LDS_{ar} framework as follows.

Definition 3.6 (Warrant – Version 1). Let $C^k(\Gamma)$ be the set of all formulas that can be obtained from Γ via $\mathrel{|\!\!\sim_{Arg}}$ and $\mathrel{|\!\!\sim_{\mathcal{T}}}$ in at most k steps. A literal α is said to be *warranted* iff $\mathbf{T}_i^U(\mathcal{A},...){:}\alpha \in C^k(\Gamma)$, and there is no $k' > k$, such that $\mathbf{T}_j^D(\mathcal{A},...){:}\alpha \in (C^{k'}(\Gamma) \setminus C^k(\Gamma))$.

In fact, this approach to compute warrant resembles Pollock's original ideas of "ultimately justified belief" [33]. Note that Def. 3.6 forces the computation of the deductive closure under "$\mathrel{|\!\!\sim_{\mathcal{T}}}$" in order to determine whether a literal is warranted or not. Fortunately this is not necessarily the case, since warrants can be captured in terms of a *precedence relation* "$\sqsubset$ " between dialectical labels. Informally, we will write $\mathbf{T} \sqsubset \mathbf{T}$' whenever $\mathbf{T}$ reflects a state in a dialogue which is *previous* to $\mathbf{T}$' (in other words, $\mathbf{T}$' stands for a dialogue which evolves from $\mathbf{T}$ by incorporating new arguments). A *final label* is a dialectical label that cannot be extended any further.

Definition 3.7 (Warrant – Version 2). [8] Let Γ be an argumentative theory, such that $\Gamma \mathrel{|\!\!\sim_{\mathcal{T}}} \mathbf{T}_i^U(\mathcal{A},\ldots){:}\alpha$ and $\mathbf{T}_i^U$ is a final label (*i.e.*, it is not the case that $\Gamma \mathrel{|\!\!\sim_{\mathcal{T}}} \mathbf{T}_j^D(\mathcal{A},\ldots){:}\alpha$ and $\mathbf{T}_i^U \sqsubset \mathbf{T}_j^D$). Then α is a warranted literal wrt Γ.

In the next Subsection we will present a worked example which illustrates the main aspects of the LDS_{ar} framework for representing knowledge and performing argumentative inference.

---

[7]We require arguments to be minimal wrt set inclusion as it is a common requirement in several argument frameworks, starting with [40].

[8]It can be proven that Def. 3.7 and 3.6 are equivalent [9, 7].

$$
\begin{array}{rl}
\emptyset: & \sim f \leftarrow pc \\
\emptyset: & sw1 \leftarrow \\
\emptyset: & sw2 \leftarrow \\
\emptyset: & sw3 \leftarrow \\
\emptyset: & h \leftarrow \\
\{ pf \leftarrow sw1 \}: & pf \leftarrow sw1 \\
\{ f \leftarrow pf \}: & f \leftarrow pf \\
\{ po \leftarrow sw2 \}: & po \leftarrow sw2 \\
\{ o \leftarrow po \}: & o \leftarrow po \\
\{ e \leftarrow f, o \}: & e \leftarrow f, o \\
\{ \sim e \leftarrow f, o, h \}: & \sim e \leftarrow f, o, h \\
\{ \sim o \leftarrow h \}: & \sim o \leftarrow h \\
\{ pc \leftarrow pf, l \}: & pc \leftarrow pf, l \\
\{ l \leftarrow sw2 \}: & l \leftarrow sw2 \\
\{ \sim l \leftarrow sw2, sw3 \}: & \sim l \leftarrow sw2, sw3 \\
\{ f \leftarrow sw3 \}: & f \leftarrow sw3
\end{array}
$$

FIGURE 4. Argumentative theory Γ_{engine}

3.3. A Worked Example[9]

Consider an intelligent agent involved in controlling an engine with three switches $sw1$, $sw2$, and $sw3$. These switches regulate different features of the engine, such as the pumping system, speed, etc. Suppose we have defeasible information about how this engine works.

- If the pump is clogged (pc), then the engine gets no fuel ($\sim f$).
- When $sw1$ is on, fuel is normally pumped properly (pf).
- When fuel is pumped properly (pf), fuel is usually ok (f).
- When $sw2$ is on, oil is usually pumped (po).
- When oil is pumped (po), it usually works ok (o).
- When there is oil and fuel $(o) \wedge (f)$, usually the engine works ok (e).
- When there is fuel, oil, and heat $(o) \wedge (f) \wedge (h)$ then the engine is usually not ok ($\sim e$).
- When there is heat (h), normally there are oil problems ($\sim o$).
- When fuel is pumped (pf) and speed is low (l), then there are reasons to believe that the pump is clogged (pc).
- When $sw2$ is on, usually speed is low (l).
- When $sw3$ is on, usually fuel is ok (f).

Suppose we also know some particular facts: $sw1$, $sw2$, and $sw3$ are on, and there is heat (h). The knowledge of such an agent can be modeled by the argumentative theory Γ_{engine} shown in Figure 4. From the theory Γ_{engine}, the argument $\mathcal{A}$:e, with

$$\mathcal{A} = \{ (pf \leftarrow sw1), (po \leftarrow sw2), (f \leftarrow pf), (o \leftarrow po), (e \leftarrow f, o) \}$$

[9]This example has been adapted from [11]

can be inferred via $\mathrel{|\!\!\sim}_{Arg}$ by applying the inference rules Intro-NR twice (inferring $sw1$ and $sw2$), then Intro-RE twice (inferring $pf \leftarrow sw1$ and $po \leftarrow sw2$), then Intro-RE twice again to infer $f \leftarrow pf$ and $o \leftarrow po$, and finally Intro-RE once again to infer $e \leftarrow f, o$. In a similar way, arguments $\mathcal{B}{:}{\sim}f$, $\mathcal{C}{:}{\sim}l$, $\mathcal{D}{:}f$ and $\mathcal{E}{:}{\sim}e$ can be derived via $\mathrel{|\!\!\sim}_{Arg}$, with[10]

$$
\begin{array}{lcl}
\mathcal{B} & = & \{\ (pf \leftarrow sw1),\ (l \leftarrow sw2),\ (pc \leftarrow pf, l)\ \} \\
\mathcal{C} & = & \{\ ({\sim}l \leftarrow sw2, sw3)\ \} \\
\mathcal{D} & = & \{\ ({\sim}l \leftarrow sw2, sw3)\ \} \\
\mathcal{E} & = & \{\ (pf \leftarrow sw1),\ (po \leftarrow sw2),\ (f \leftarrow pf),\ (o \leftarrow po),\ ({\sim}e \leftarrow f, o, h)\ \}
\end{array}
$$

Note that the arguments $\mathcal{B}{:}{\sim}f$, and $\mathcal{E}{:}{\sim}e$, are *counterarguments* for the original argument $\mathcal{A}{:}e$, whereas $\mathcal{C}{:}{\sim}l$ and $\mathcal{D}{:}f$ are counterarguments for $\mathcal{B}{:}{\sim}f$. In each of these cases, these counterarguments are also defeaters according to the specificity preference criterion [40]. Assuming such defeat relationship among arguments, the following formulæ can be inferred via $\mathrel{|\!\!\sim}_{\mathcal{T}}$:

$$
\begin{array}{ll}
1)\mathbf{T}_1^*(\mathcal{A}){:}e & \textsf{Intro-1D} \\
2)\mathbf{T}_2^*(\mathcal{B}){:}{\sim}f & \textsf{Intro-1D} \\
3)\mathbf{T}_3^*(\mathcal{C}){:}{\sim}l & \textsf{Intro-1D} \\
4)\mathbf{T}_4^*(\mathcal{D}){:}f & \textsf{Intro-1D} \\
5)\mathbf{T}_5^*(\mathcal{E}){:}{\sim}e & \textsf{Intro-1D} \\
6)\mathbf{T}_2^*(\mathcal{B}, \mathbf{T}_3^*(\mathcal{C}), \mathbf{T}_4^*(\mathcal{D})){:}{\sim}f & \textsf{Intro-ND}, 3), 4) \\
7)\mathbf{T}_1^*(\mathcal{A}, \mathbf{T}_2^*(\mathcal{B}, \mathbf{T}_3^*(\mathcal{C}), \mathbf{T}_4^*(\mathcal{D})), \mathbf{T}_5^*(\mathcal{E})){:}e & \textsf{Intro-ND}, 6) \\
8)\mathbf{T}_5^U(\mathcal{E}){:}{\sim}e & \textsf{Mark-Atom} \\
9)\mathbf{T}_1^D(\mathcal{A}, \mathbf{T}_2^*(\mathcal{B}, \mathbf{T}_3^*(\mathcal{C}), \mathbf{T}_4^*(\mathcal{D})), \mathbf{T}_5^U(\mathcal{E})){:}e & \textsf{Mark-1D}, 8)
\end{array}
$$

Note that the formula obtained in step (7) has a final label associated with it, since it cannot be 'expanded' from previous formulæ. Hence, following Def. 3.7, we can conclude that e is not warranted.

4. Non-monotonic Inference Relationships: Fundamentals[11]

In classical logic, inference rules allow us to determine whether a given wff γ follows via "$\vdash$" from a set Γ of wffs. In classical logic the "$\vdash$" relationship is a *consequence relationship* (satisfying idempotence, *cut* and monotonicity). As non-monotonic and defeasible logics evolved into a valid alternative to formalize commonsense reasoning, a similar concept was needed to capture consequence without demanding some of these requirements (*e.g.* monotonicity). This led to the definition of a more generic notion of inference, namely *inference relationships.* New properties were defined and gained interest in this setting. In this section we will introduce the definition of (non-monotonic) inference relationship, as well the definitions some distinguished properties that characterize them.

[10]For the sake of clarity, we use parentheses to separate different elements (formulas) present in an argument.

[11]This section is based on the excellent overview on consequence and inference relationships given in [2].

4.1. Inference Relationship. Pure Logical Properties

Definition 4.1 (Inference Relationship $\mid\!\sim$. Inference Operator $C(\Gamma)$). Let Γ be a set of wffs in a language $\mathcal{L}$ and let γ be a wff in Γ. We will write $\Gamma \mid\!\sim \gamma$ if γ is a (non-monotonic) consequence of Γ. We define $C(\Gamma) = \{\gamma \mid \Gamma \mid\!\sim \gamma\}$.

Given an inference relationship "$\mid\!\sim$ " and a set Γ of sentences, the following are called *basic* (or *pure*) properties associated with *any* inference operator $C(\Gamma)$:

1. **Inclusion**: $\Gamma \subseteq C(\Gamma)$
2. **Idempotence**: $C(\Gamma) = C(C(\Gamma))$
3. **Cut**: $\Gamma \subseteq \Phi \subseteq C(\Gamma)$ implies $C(\Phi) \subseteq C(\Gamma)$
4. **Cautious Monotonicity**: $\Gamma \subseteq \Phi \subseteq C(\Gamma)$ implies $C(\Gamma) \subseteq C(\Phi)$.
5. **Cummulativity**: $\Gamma \subseteq \Phi \subseteq C(\Gamma)$ implies $C(\Gamma) = C(\Phi)$.
6. **Monotonicity**: $\Gamma \subseteq \Phi$ implies $C(\Gamma) \subseteq C(\Phi)$

The intuitive meaning of inclusion and idempotence should be clear without further comments. The *cut* rule states that expanding the information in Γ by adding new propositions from $C(\Gamma)$ does not result in new conclusions being obtained. Cautious monotonicity constitutes somehow the 'inverse' of *cut*: adding new lemmas does not decrease inference power, *i.e.* the set of conclusions that can be obtained from a given theory. Combining *cut* and cautious monotonicity we get *cummulativity*, which states that intermediate proofs (lemmas) can be used as part of other (more complex) proofs without affecting the soundness of their conclusions.

These properties are called *pure*, since they can be applied to any language $\mathcal{L}$, and are abstractly defined for an arbitrary inference relationship "$\mid\!\sim$ ". Nevertheless, other properties which link classical inference with an arbitrary inference relationship can be stated. These properties will be discussed next. In what follows we will assume that Th stands for an operator that characterizes classical inference, whereas C corresponds to some (non-monotonic) inference relationship "$\mid\!\sim$ ".

4.2. Horn and Non-Horn Logical Properties

A common name for cataloging non-pure properties is the distinction between *Horn properties* and *non-Horn properties.* Horn properties have the form *"from the <u>presence</u> of some particular inferences, the <u>presence</u> of some other inferences can be assured"*. Non-Horn properties, on the other hand, have the form *"from the <u>absence</u> of some particular inferences, the <u>absence</u> of some other inferences can be assured"*. Next we summarize the most important non-pure properties:[12]

Horn Properties.

1. **Supraclassicality**: $Th(A) \subseteq C(A)$
2. **Left Logical Equivalence**: $Th(A) = Th(B)$ implies $C(A) = C(B)$
3. **Right Weakening**: If $x \supset y \in Th(A)$ and $x \in C(A)$ then $y \in C(A)$.[13]

[12] An in-depth discussion of these properties can be found in [2].

[13] It should be noted that "$\supset$" stands for material implication, to be distinguished from the symbol " $\leftarrow$ " used in a logic programming setting.

4. **Conjunction of Conclusions**: If $x \in C(A)$ and $y \in C(A)$ then $x \wedge y \in C(A)$.
5. **Subclassical Cummulativity**: If $A \subseteq B \subseteq Th(A)$ then $C(A) = C(B)$.
6. **Left Absorption**: $Th(C(\Gamma)) = C(\Gamma)$.
7. **Right Absorption**: $C(Th(\Gamma)) = C(\Gamma)$.

Non-Horn Properties.

1. **Rationality of Negation**: if $A \mid\!\sim z$ then either $A \cup \{x\} \mid\!\sim z$ or $A \cup \{\neg x\} \mid\!\sim z$.
2. **Disjunctive Rationality**: if $A \cup \{x \vee y\} \mid\!\sim z$ then $A \cup \{x\} \mid\!\sim z$ or $A \cup \{y\} \mid\!\sim z$.
3. **Rational Monotonicity**: if $A \mid\!\sim z$ then either $A \cup \{x\} \mid\!\sim z$ or $A \mid\!\sim \neg x$.

5. Capturing Argument Construction and Warrant in LDS$_{ar}$

In this Section we will present two non-monotonic inference operators for LDS_{ar} associated with computing arguments and warranted literals, respectively. As discussed in Section 3, these concepts are based on the two inference relationships $\mid\!\sim_{Arg}$ and $\mid\!\sim_{\mathcal{T}}$, characterized by the natural deduction rules shown in Figures 1 and 2, respectively. Given an argumentative theory Γ, $\mid\!\sim_{Arg}$ is associated with labelled formulas corresponding to those arguments which can be inferred from Γ, whereas $\mid\!\sim_{\mathcal{T}}$ allows to obtain labelled formulas $\mathbf{T}(\mathcal{A}, \ldots)$:$\alpha$, where the label $\mathbf{T}(\mathcal{A}, \ldots)$ depicts the dialectial analysis carried out to determine the epistemic status of the literal α. In the case of $\mid\!\sim_{\mathcal{T}}$, we are particularly interested in those labelled formulas corresponding to ultimately accepted (or warranted) literals, as characterized in Def. 3.6.

How can we formalize the notion of *theorem* in the context of LDS_{ar}? In this respect, there is an important aspect to take into account: LDS_{ar} is based on an *extension* of logic programming, where literals can be preceded by strong negation, and some pieces of information can be distinguished (labeled) as 'defeasible'. Consequently, the notion of theorem in such a logic programming setting will be more restricted than the one used in classical logic, as only literals can be obtained as conclusions derivable from a given logic program. This leads to consider a *specialized* consequence operator for our framework, oriented towards a logic programming setting [27], where typically SLD resolution is used to model which literals follow from a given logic program. Formally:

Definition 5.1 (Consequence Operator Th$_{sld}(\Gamma)$). Given an argumentative theory Γ, we define $Th_{sld}(\Gamma)$ as the set of all possible empty arguments that can be obtained form Γ. Formally:

$$Th_{sld}(\Gamma) = \{\emptyset{:}h \ \mid \Gamma \mid\!\sim_{Arg} \emptyset{:}h\}, \text{for some } h \in \mathcal{L}_{\mathsf{KR}}$$

According to definition 5.1, 'classical' consequences from an argumentative theory Γ will be arguments whose argument label is the empty set (*i.e.*, arguments that do not rely on any defeasible information). We want to compare this notion of derivation with respect to inference relationships $\mid\!\sim_{Arg}$ and $\mid\!\sim_{\mathcal{T}}$. We will define two suitable inference operators for capturing those arguments that can be derived from Γ, and those literals that can be warranted from it. Warranted conclusions will be represented in terms of of new facts in our labelled language $\mathcal{L}_{\mathsf{Labels}}$.

Definition 5.2 (Inference Operators $\mathbf{C}_{arg}(\Gamma)$ and $\mathbf{C}_{war}(\Gamma)$). Given an argumentative theory Γ, we will define two non-monotonic inference operators $C_{arg}(\Gamma)$ and $C_{war}(\Gamma)$ as follows:

$$\begin{aligned} C_{arg}(\Gamma) &= \{\ \mathcal{A}{:}\alpha \mid \Gamma \mathrel{|\!\!\sim_{Arg}} \mathcal{A}{:}\alpha, \text{ where } \alpha \text{ is a literal in } \mathsf{Wffs}(\mathcal{L}_{\mathsf{KR}})\ \} \\ C_{war}(\Gamma) &= \{\ \emptyset{:}\alpha \mid \Gamma \mathrel{|\!\!\sim_{\mathcal{T}}} \mathbf{T}(\mathcal{A},\ldots){:}\alpha, \\ &\qquad \text{where } \alpha \text{ is a warranted literal in } \mathsf{Wffs}(\mathcal{L}_{\mathsf{KR}})\ \} \end{aligned}$$

In what follows we will analyze the different logical properties discussed in Section 4 in the context of LDS_{ar}.

5.1. Logical Properties of $\mathbf{C}_{arg}$

Inclusion does not hold in general for C_{arg}, since not every piece of defeasible information can be used as part of an argument. However, it does hold for (non-defeasible) facts. Therefore we refer to it as *restricted* inclusion.

Proposition 5.3 (Restricted Inclusion). *The operator $C_{arg}(\Gamma)$ only satisfies inclusion wrt the non-defeasible information present in Γ, i.e. all those labelled formulas $\emptyset{:}\alpha \in \Gamma$*

Proof. Let Γ be an argumentative theory, and let $\emptyset{:}\alpha$ be a non-defeasible formula in Γ. Clearly, from rule $\mathsf{Intro\text{-}NR}$ we can derive $\emptyset{:}\alpha$ from Γ, *i.e.* $\Gamma \mathrel{|\!\!\sim_{Arg}} \emptyset{:}\alpha$. Consequently, if $\emptyset{:}\alpha \in \Gamma$ then $\emptyset{:}\alpha \in C_{arg}(\Gamma)$.
A counterexample suffices to show that inclusion does not hold for defeasible information. Consider $\Gamma = \{\ \{\sim\alpha\}{:}\sim\alpha,\ \emptyset{:}\alpha\ \}$. Then $\Gamma \mathrel{|\!\!\sim_{Arg}} \emptyset{:}\alpha$, but $\Gamma \not\mathrel{|\!\!\sim_{Arg}} \{\sim\alpha\}{:}\sim\alpha$. □

The logical property of idempotence holds, as once the clogical closure of Γ under $\mathrel{|\!\!\sim_{Arg}}$ has been computed, no new arguments can be obtained by a new application of $\mathrel{|\!\!\sim_{Arg}}$.

Proposition 5.4 (Idempotence). *The operator $C_{arg}(\Gamma)$ satisfies idempotence, i.e.* $C_{arg}(\Gamma) = C_{arg}(C_{arg}(\Gamma))$

Proof. ($\Rightarrow$) We must show that $C_{arg}(\Gamma) \subseteq C_{arg}(C_{arg}(\Gamma))$. Let Γ be a theory. Clearly, $C_{arg}(\Gamma)$ will correspond to a set of arguments

$$C_{arg}(\Gamma) = \{\mathcal{A}_1{:}\alpha_1, \ldots \mathcal{A}_k{:}\alpha_k,\}$$

where every α_i is a literal in $\mathsf{Wffs}(\mathcal{L}_{\mathsf{KR}})$, $i = 1 \ldots k$. Note that by definition of $\mathrel{|\!\!\sim_{Arg}}$, there are only two inference rules that can be applied on the basis of $C_{arg}(\Gamma)$, namely $\mathsf{Intro\text{-}NR}$ and $\mathsf{Intro\text{-}RE}$ (as rule Intro-$\wedge$ does not derive an argument, and Elim-$\leftarrow$ requires having a formula $\Phi{:}\beta \leftarrow \alpha \in C_{arg}(\Gamma)$, which cannot be the case as $\beta \leftarrow \alpha$ is not a literal. Let us analzye the application of $\mathsf{Intro\text{-}NR}$ and $\mathsf{Intro\text{-}RE}$, which correspond to axioms in the characterization of $\mathrel{|\!\!\sim_{Arg}}$:

1. Intro-NR applications can only result in new empty arguments being introduced. Hence, $\emptyset{:}\alpha_1 \in C_{arg}(C_{arg}(\Gamma))$ whenever $\emptyset{:}\alpha_1 \in C_{arg}(\Gamma)$
2. Intro-RE applications can result in non-empty arguments $\Phi{:}\alpha$ being introduced whenever $Strict(\Gamma) \cup \Phi \not\vdash \bot$. Clearly, every argument $\mathcal{A}{:}\alpha$ in $C_{arg}(\Gamma)$ satisfies this restriction. Consequently, those arguments $\mathcal{A}{:}\alpha$ in $C_{arg}(\Gamma)$ will be also members in $C_{arg}(C_{arg}(\Gamma))$.

The proof in the other direction ($\Leftarrow$) follows an analogous way of reasoning. $\square$

As the inference operator $\mid\!\sim_{Arg}$ models the construction of defeasible arguments, monotonicity does not hold, as expected.

Proposition 5.5 (Monotonicity). *The operator $C_{arg}(\Gamma)$ does not satisfy monotonicity.*

Proof. A counterexample suffices. Consider the argumentative theory $\Gamma = \{\emptyset{:}q, \{p \leftarrow q\}{:}p \leftarrow q\}$. Clearly $\Gamma \mid\!\sim_{Arg} \mathcal{A}{:}p$, with $\mathcal{A} = \{p \leftarrow q\}$. But $\Gamma' = \Gamma \cup \{\emptyset{:}{\sim}p\}$ is such that $\Gamma \not\mid\!\sim_{Arg} \mathcal{A}{:}p$. $\square$

Semi-monotonicity is an interesting property suggested by Makinson & Schlechta [30] for analyzing non-monotonic consequence relationships. It is satisfied if all defeasible consequences from a given theory are preserved when the theory is augmented with new *defeasible* information. Next we show that semi-monotonicity holds for $\mid\!\sim_{Arg}$.

Proposition 5.6 (Semi-monotonicity). *The operator $C_{arg}(\Gamma)$ satisfy semi-monotonicity, i.e. $C_{arg}(\Gamma) \subseteq C_{arg}(\Gamma \cup \Gamma')$, where Γ' is a theory involving only defeasible information.*

Proof. Proof is direct from the structure of the inference rules. Assume $\Gamma \mid\!\sim_{Arg} \mathcal{A}{:}\alpha$, and consider $\Gamma \cup \Gamma'$ as stated in the proof of Prop. 5.5. Clearly, the sequence of steps in the proof $\Gamma \cup \Gamma' \mid\!\sim_{Arg} \mathcal{A}{:}\alpha$ is still valid, since all preconditions in inference rules are defined wrt $Strict(\Gamma) = Strict(\Gamma \cup \Gamma')$. Hence $\Gamma \cup \Gamma' \mid\!\sim_{Arg} \mathcal{A}{:}\alpha$. $\square$

Cummulativity holds for argument construction. The importance of this property will be discussed later in section 8.

Lemma 5.7 (Cummulativity). [14] *Let Γ be an argumentative theory, and let α_1 and α_2 be literals in $\mathsf{Wffs}(\mathcal{L}_{\mathsf{KR}})$. Then $\Gamma \mid\!\sim_{Arg} \mathcal{A}_1{:}\alpha_1$ implies that*

$$\Gamma \cup \{\mathcal{A}_1{:}\alpha_1\} \mid\!\sim_{Arg} \mathcal{A}_2{:}\alpha_2 \quad \textit{iff} \quad \Gamma \mid\!\sim_{Arg} \mathcal{A}_2{:}\alpha_2$$

The property of supraclassicality needs also a particular characterization in LDS_{ar}, as theorems are restricted to literals supported by empty arguments (as indicated in Def. 5.1). Therefore we refer to the restricted version of this property as *Horn Supraclassicality*.

Proposition 5.8 (Horn Supraclassicality). *The operator $C_{arg}(\Gamma)$ satisfies Horn supraclassicality wrt Th_{sld}, i.e. $Th_{sld}(\Gamma) \subseteq C_{arg}(\Gamma)$.*

[14] Proof not included for space reasons.

Proof. Proof is direct from Definition 5.1. Every member in $Th_{sld}(\Gamma)$ is an empty argument, and as such it is a member in $C_{arg}(\Gamma)$. Therefore $Th_{sld}(\Gamma) \subseteq C_{arg}(\Gamma)$. □

Proposition 5.9 (Left-Logical Equivalence). *The operator $C_{arg}(\Gamma)$ satisfies left-logical equivalence (i.e., given the formulas α, β and γ such that $\alpha \longleftrightarrow \beta$ (both are logically equivalent) if $\gamma \in C_{arg}(\Gamma \cup \{\alpha\})$ then $\gamma \in C_{arg}(\Gamma \cup \{\beta\})$.*

Proof. Note that logical equivalence in LDS_{ar} is restricted to rules and facts. From the definition 3.2 therefore $\alpha \longleftrightarrow \beta$ iff both α and β correspond to the same fact (i.e. $\alpha = \beta = \emptyset{:}\phi$, or $\alpha = \beta = \phi{:}\phi$, where ϕ is a fact) or both α and β correspond to logically equivalent rules. In a logic programming setting two rules $P \leftarrow Q_1, Q_2, \ldots, Q_k$ and $P' \leftarrow Q'_1, Q'_2, \ldots, Q'_k$ will be equivalent iff P and P' are the same fact ($P \equiv P'$) and the set $\{Q_1, Q_2, \ldots, Q_k\}$ is equal to the set $\{Q'_1, Q'_2, \ldots, Q'_k\}$ (i.e., rules differ just in a permutation of the literals present in the antecedent). If $\gamma \in C_{arg}(\Gamma \cup \{\alpha\})$ (hypothesis), then there exists an argument γ which follows from $C_{arg}(\Gamma \cup \{\alpha\})$ via a sequence of applications of natural deduction rules from $\mathrel{|\!\sim}_{Arg}$. From the definition of $\mathsf{Intro}\text{-}\wedge$ and $\mathsf{Elim}\text{-}\leftarrow$ it is straightforward to see that if such a proof could be obtained from $C_{arg}(\Gamma \cup \{\alpha\})$ and β is logically equivalent to α, then the same proof can be obtained from $C_{arg}(\Gamma \cup \{\beta\})$. □

Note that the property of right weakening cannot be considered (in a strict sense) in LDS_{ar}, since the deductive system associated with Th_{sld} does not allow the application of the deduction theorem [6]. Therefore, wffs of the form "$x \leftarrow y$" cannot be derived via $\vdash_{\text{SLD}}$. On the other hand, conjunction of conclusions does not hold for argument construction (as shown in Proposition 5.11). Therefore we will provide an alternative, restricted version of Right weakening considering only clauses with only one literal in the antecedent. We call such a property *Horn Right Weakening.*

Proposition 5.10 (Horn Right Weakening). *The operator $C_{arg}(\Gamma)$ satisfies Horn right weakening, i.e. if $\mathcal{A}{:}y \in C_{arg}(\Gamma)$ and $\emptyset{:}x \leftarrow y \in \Gamma$ (where y is a literal in $\mathsf{Wffs}(\mathcal{L}_{\mathsf{KR}})$) then $\mathcal{A}'{:}x \in C_{arg}(\Gamma)$.*

Proof. Suppose $\mathcal{A}{:}y \in C_{arg}(\Gamma)$. Clearly, $Strict(\Gamma) \not\vdash_{\text{SLD}} \bot$ (otherwise it would not have been possible to infer $\mathcal{A}{:}y$). However, $x \leftarrow y \in Strict(\Gamma)$. Then applying rule $\mathsf{Elim}\text{-}\leftarrow$ it holds that $\mathcal{A}{:}x$ can be derived from Γ, or equivalently $\mathcal{A}{:}x \in C_{arg}(\Gamma)$. □

Proposition 5.11 (Conjunction of Conclusions). [15] *The operator $C_{arg}(\Gamma)$ does not satisfy conjunction of conclusions, i.e. if $x \in C_{arg}(\Gamma)$ and $y \in C_{arg}(\Gamma)$, then it does not hold that $x \wedge y \in C_{arg}(\Gamma)$.*

Proof. A counterexample suffices. Consider the following theory

$$\begin{array}{ll} \Gamma = \{ & (\{p \leftarrow q\}{:}p \leftarrow q), (\{r \leftarrow z\}{:}r \leftarrow z) \\ & (\emptyset{:}q), (\emptyset{:}z), (\emptyset{:}w), (\emptyset{:}{\sim}w \leftarrow p, r) \ \} \end{array}$$

[15]Conjunction of conclusions in an argument is not possible as conclusions of arguments are restricted to literals (see Def. 3.3). However, we consider the general case as it is allowed by the inference rules in $\mathrel{|\!\sim}_{Arg}$.

Then there exists an argument $\mathcal{A}_1$:p with $\mathcal{A}_1$=$\{\ p\ \leftarrow\ q\ \}$, and an argument $\mathcal{A}_2$:r with $\mathcal{A}_2$=$\{\ r\ \leftarrow\ z\ \}$. However, the formula $\mathcal{A}_3$:p, r cannot be derived (nor any other with conclusion p, r) since $Strict(\Gamma) \cup \{p, r\} \vdash_{\text{SLD}} \perp$. □

Proposition 5.12 (Subclassical Cummulativity). *The operator $C_{arg}(\Gamma)$ satisfies subclassical cummulativity, i.e. $\Gamma \subseteq \Gamma' \subseteq Th_{sld}(\Gamma)$ implies $C_{arg}(\Gamma) = C_{arg}(\Gamma')$*

Proof. Let Γ be an argumentative theory, and assume $\Gamma \subseteq Th_{sld}(\Gamma)$. Note that $Th_{sld}(\Gamma)$ involves only labelled formulas with an empty label. Consequently, it holds that for any Φ:$\alpha \in \Gamma$ it is the case that $\Phi = \emptyset$, and α is a literal. But if this is the case, the only applicable inference rule on Γ is Intro-NR, which correspond to an axiom schema. Necessarily it follows that $\Gamma = Th_{sld}(\Gamma)$, and consequently $\Gamma' = \Gamma$, so that $C_{arg}(\Gamma) = C_{arg}(\Gamma')$, as we wanted to prove. □

Proposition 5.13 (Left Absorption). *The operator $C_{arg}(\Gamma)$ does not satisfy left absorption, i.e. $Th_{sld}(C_{arg}(\Gamma)) \neq C_{arg}(\Gamma)$.*

Proof. $Th_{sld}(C_{arg}(\Gamma))$ involves only empty arguments that can be derived from $C_{arg}(\Gamma)$. But there can exists non-empty arguments in $C_{arg}(\Gamma)$. Therefore $Th_{sld}(C_{arg}(\Gamma)) \neq C_{arg}(\Gamma)$. □

Proposition 5.14 (Right Absorption). *The operator $C_{arg}(\Gamma)$ does not satisfy right absorption, i.e. $C_{arg}(Th_{sld}(\Gamma)) \neq C_{arg}(\Gamma)$.*

Proof. As shown in prop 5.9, when computing $Th_{sld}(\Gamma)$ all defeasible information that might originally be present in Γ is lost. However, this information could appear in arguments in $C_{arg}(\Gamma)$. Then it holds that $C_{arg}(Th_{sld}(\Gamma)) \neq C_{arg}(\Gamma)$. □

Proposition 5.15 (Rational Negation). *The operator $C(\Gamma)$ does not satisfy rational negation.*

Proof. A counterexample suffices. Consider the following theory Γ=$\{\emptyset$:$\sim p \leftarrow x$, $\emptyset$:$\sim p \leftarrow \sim x$, $\emptyset$:r, $\{z \leftarrow p\}$:$z \leftarrow p$, $\{p \leftarrow r\}$:$p \leftarrow r$ $\}$. Then $\Gamma \mid\!\sim_{Arg} \mathcal{A}$:$z$, with $\mathcal{A} = \{z \leftarrow p, p \leftarrow r\}$. However, $\Gamma \cup \{\emptyset$:$x\} \not\mid\!\sim_{Arg} \mathcal{A}$:$z$, and $\Gamma \cup \{\emptyset$:$\sim x\} \not\mid\!\sim_{Arg} \mathcal{A}$:$z$ (since in both cases the use of $\{p \leftarrow r\}$:$p \leftarrow r$ is not valid because of consistency constraints). □

Proposition 5.16 (Rational Monotonicity). *The operator $C(\Gamma)$ does not satisfy rational monotonicity.*

Proof. Consider the example given in the proof of Proposition 5.15, where $\Gamma \mid\!\sim_{Arg} \mathcal{A}$:$z$, with $\mathcal{A} = \{z \leftarrow p, p \leftarrow r\}$. If we consider now $\Gamma \cup \{\emptyset$:$x\}$ it holds that $\Gamma \cup \{\emptyset$:$x\} \not\mid\!\sim_{Arg} \mathcal{A}$:$z$, and $\Gamma \not\mid\!\sim_{Arg} \sim x$. Therefore rational monotonicity is not satisfied. □

Clearly, the operator C_{arg} does not satisfy disjunctive rationality either, since disjunctions cannot be expressed as formulas in $\mathcal{L}_{\mathsf{KR}}$.

5.2. Logical Properties of $\mathbf{C}_{war}$

Next we will analyze some relevant logical properties of C_{war}, the inference operator for deriving warranted conclusions. Clearly, inclusion cannot be defined for considering arbitrary labelled formulas, as only formulas of the form $\emptyset{:}\alpha$ will be part of C_{war}.

Proposition 5.17 (Restricted Inclusion). *The operator $C_{war}(\Gamma)$ only satisfies inclusion wrt (non-defeasible) facts in Γ.*

Proof. Let Γ be an argumentative theory, and let $\emptyset{:}\alpha$ be a labelled formula in Γ. Clearly, from rule Intro-NR it follows that $\Gamma \mid\!\sim_{Arg} \emptyset{:}\alpha$. The formula $\emptyset{:}\alpha$ provides an argument for α. By rule Intro-1D, the formula $\mathbf{T}^*(\emptyset){:}\alpha$ can be derived. Clearly, rule Intro-1D cannot be applied, as no defeaters can be found for the empty set. Therefore $\Gamma \mid\!\sim_{\mathcal{T}} \mathbf{T}^U_*(\emptyset){:}\alpha$, and α is a warranted literal, or equivalently $\emptyset{:}\alpha \in C_{war}(\Gamma)$.
□

Proposition 5.18 (Idempotence). *The operator $C_{war}(\Gamma)$ satisfies idempotence, i.e. $C_{war}(\Gamma) = C_{war}(C_{war}(\Gamma))$*

Proof. (Sketch) If we consider $C_{war}(\Gamma)$ corresponds to a set of labelled formulas of the form $\emptyset{:}\alpha$, it is clear that all they are warranted wrt $C_{war}(\Gamma)$ and non-conflicting. But from Proposition 5.17 it follows that every of such formulas in $C_{war}(\Gamma)$ is warranted, that is $C_{war}(\Gamma) \subseteq C_{war}(C_{war}(\Gamma))$. Besides, applying the C_{war} operator on $C_{war}(\Gamma)$ will return only those labelled formulas of the form $\emptyset{:}\alpha$ in $C_{war}(\Gamma)$ which are warranted, that is $C_{war}(C_{war}(\Gamma)) \subseteq C_{war}(\Gamma)$. Therefore $C_{war}(C_{war}(\Gamma)) = C_{war}(\Gamma)$. □

Proposition 5.19 (Monotonicity). *The operator $C_{war}(\Gamma)$ does not satisfy monotonicity.*

Proof. A counterexample suffices. Consider the example given in Prop 5.5. In that case, $\Gamma \mid\!\sim_{\mathcal{T}} \mathbf{T}^U(\{p \leftarrow q\}){:}p$ and p is warranted. However, in $\Gamma \cup \{\emptyset{:}{\sim}p\}$ there is no argument with conclusion p, and consequently p is not warranted. □

In contrast with $\mid\!\sim_{Arg}$, semi-monotonicity does not hold for $\mid\!\sim_{\mathcal{T}}$. The reason is the following: adding new defeasible information cannot invalidate existing arguments, but it can enable building new arguments that were not derivable before. Hence, dialectical relationships among arguments are different. Arguments that were warranted may therefore no longer keep that status. Formally:

Proposition 5.20 (Semi-monotonicity). *The operator $C_{war}(\Gamma)$ does not satisfy semi-monotonicity, i.e. $C_{war}(\Gamma) \not\subseteq C_{war}(\Gamma \cup \Gamma')$, where Γ' is an argumentative theory which involves only defeasible information.*

Proof. Consider the following counterexample. Given the theory $\Gamma = \{\ (\emptyset{:}q),\ (\ \{p \leftarrow r\}{:}p \leftarrow r\ ),\ (\ \{r \leftarrow q\}{:}r \leftarrow q\ )\ \}$, it is clear that $\emptyset{:}p \in C_{war}(\Gamma)$, since there exists an argument $\mathcal{A}{:}p$ with $\mathcal{A}{=}\{p \leftarrow r,\ r \leftarrow q\ \}$, such that $\mathcal{A}{:}p$ has no defeaters (and hence p is warranted).

Consider now $\Gamma'{=}\Gamma \cup\ \{{\sim}p \leftarrow q{:}{\sim}p \leftarrow q\}$. In this case, an argument $\mathcal{B}{:}{\sim}p$ could be obtained, with $\mathcal{B}{=}\{{\sim}p \leftarrow q\ \}$, such that $\mathcal{B}{:}{\sim}p$ defeats $\mathcal{A}{:}p$. Therefore $\emptyset{:}p \notin C_{war}(\Gamma')$.
□

Cummulativity does not hold for $\mid\!\sim_{\mathcal{T}}$,[16] as shown next.

Proposition 5.21 (Cummulativity). *The relationship $\mid\!\sim_{\mathcal{T}}$ does not satisfy cummulativity.*

Proof. A counterexample suffices. Consider the theory $\Gamma = \{ (\emptyset{:}{\sim}s \leftarrow q), (\emptyset{:}p), (q \leftarrow p{:}q \leftarrow p), (s \leftarrow p{:}s \leftarrow p), ({\sim}q \leftarrow s{:}{\sim}q \leftarrow s) \}$. Consider the arguments $\mathcal{A}{:}q$ with $\mathcal{A}{=}\{ q \leftarrow p \}$, and $\mathcal{B}{:}{\sim}q$ with $\mathcal{B}{=}\{ ({\sim}q \leftarrow s), (s \leftarrow p) \}$, based on Γ. Note that s is a warranted literal in Γ. Consider now $\Gamma' = \Gamma \cup \{\emptyset{:}s\}$. Note that ${\sim}q$ is warranted in Γ' but is not warranted wrt Γ (since $\mathcal{A}{:}q$ defeats $\mathcal{B}{:}{\sim}q$). $\square$

As in the case of $\mid\!\sim_{Arg}$, we will consider a restricted version of supraclassicality for analyzing $\mid\!\sim_{\mathcal{T}}$.

Proposition 5.22 (Horn Supraclassicality). *The operator $C_{war}(\Gamma)$ satisfies Horn supraclassicality wrt Th_{sld}, i.e. $Th_{sld}(\Gamma) \subseteq C_{war}(\Gamma)$.*

Proof. All literals in $Th_{sld}(\Gamma)$ have arguments with no defeaters. Therefore they are warranted, and hence they are members of $C_{war}(\Gamma)$. $\square$

Proposition 5.23 (Left-Logical Equivalence). *The operator $C_{war}(\Gamma)$ satisfies left-logical equivalence (i.e., given the formulas α, β and γ such that $\alpha \longleftrightarrow \beta$ (both are logically equivalent) if $\gamma \in C_{war}(\Gamma \cup \{\alpha\})$ then $\gamma \in C_{war}(\Gamma \cup \{\beta\})$.*

Proof. (Sketch) The reasoning is analogous as in Proposition 5.9. If γ is warranted on the basis $\Gamma \cup \{\alpha\}$, then there exists an argument $\mathcal{A}{:}\gamma$, such that $\Gamma \cup \{\alpha\} \mid\!\sim_{\mathcal{T}} \mathbf{T}^U(\mathcal{A}, \ldots){:}\gamma$. If α was used in the construction of some argument $\mathcal{B}{:}\phi$ as part of the dialectical analysis expressed in $\mathbf{T}^U(\mathcal{A}, \ldots){:}\gamma$, clearly every defeater for $\mathcal{B}{:}\phi$ will be also a defeater for $\mathcal{B}'{:}\phi$, where $\mathcal{B}'{:}\phi$ was obtained from $\Gamma \cup \{\beta\}$, as α and β are logically equivalent. Therefore the dialectical label in $\mathbf{T}^U(\mathcal{A}, \ldots){:}\gamma$ will involve the same defeaters as the dialectical label $\mathbf{T'}^U(\mathcal{A}', \ldots){:}\gamma$, where $\Gamma \cup \{\beta\} \mid\!\sim_{\mathcal{T}} \mathbf{T'}^U(\mathcal{A}', \ldots){:}\gamma$. But this accounts to saying that $\gamma \in C_{war}(\Gamma \cup \{\beta\})$, as we wanted to prove. $\square$

Note that right weakening cannot be considered wrt $\mid\!\sim_{\mathcal{T}}$, since there is no way of warranting formulas of the form "$x \leftarrow y$". However, as in the case of $\mid\!\sim_{Arg}$, an alternative version can be provided.

Proposition 5.24 (Horn Right Weakening). *The operator $C_{war}(\Gamma)$ satisfies* Horn right weakening, *i.e. if $\emptyset{:}y \in C_{war}(\Gamma)$ and it holds that $\emptyset{:}x \leftarrow y \in \Gamma$, where y is a literal in* $\mathsf{Wffs}(\Gamma)$*, then $\emptyset{:}x \in C_{war}(\Gamma)$.*

Proof. Suppose $\emptyset{:}y \in C_{war}(\Gamma)$. That means that there is an argument $\mathcal{A}{:}y$ that can be obtained from Γ such that y is warranted. But if $\mathcal{A}{:}y$ is an argument for y, and $\emptyset{:}x \leftarrow y \in \Gamma$, via $\mathsf{Elim}\text{-}\leftarrow$ it can be inferred that $\mathcal{A}{:}x$ is also an argument for x (note that the consistency constraint associated with the $\mathsf{Elim}\text{-}\leftarrow$ rule holds, as otherwise it would not have been possible to derive $\mathcal{A}{:}y$). Besides, the possible defeaters for argument $\mathcal{A}{:}y$ are the same as those for $\mathcal{A}{:}x$ (as the set $\mathcal{A}$ of defeasible information involved is the same in both cases). Therefore if $\mathcal{A}{:}y$ is warranted, then $\mathcal{A}{:}x$ is also warrantd, and consequently $\emptyset{:}x \in C_{war}(\Gamma)$, as we wanted to prove. $\square$

[16] This fact was originally suggested by Vreeswijk in the context of abstract argumentation systems [43], and further explored in [35]

Proposition 5.25 (Conjunction of Conclusions). *The operator* $C_{war}(\Gamma)$ *does not satisfy conjunction of conclusions, i.e. if* $x \in C_{war}(\Gamma)$ *and* $y \in C_{war}(\Gamma)$*, then it does not hold* $x \wedge y \in C_{war}(\Gamma)$*.*

Proof. Consider the counterexample shown in proposition 5.11. □

Proposition 5.26 (Subclassical Cummulativity). *The operator* $C_{war}(\Gamma)$ *satisfies subclassical cummulativity, i.e.* $\Gamma \subseteq \Gamma' \subseteq Th_{sld}(\Gamma)$ *implies* $C_{war}(\Gamma) = C_{war}(\Gamma')$

Proof. We can reason as in proposition 5.12. Let Γ be an argumentative theory, and assume $\Gamma \subseteq Th_{sld}(\Gamma)$. Note that $Th_{sld}(\Gamma)$ involves only labelled formulas with an empty label. Consequently, it holds that for any $\Phi{:}\alpha \in \Gamma$ it is the case that $\Phi = \emptyset$, and α is a literal. But if this is the case, the only applicable inference rule on Γ is Intro-NR, which correspond to an axiom schema. Necessarily it follows that $\Gamma = Th_{sld}(\Gamma)$, and consequently $\Gamma' = \Gamma$, so that $C_{war}(\Gamma) = C_{war}(\Gamma')$, as we wanted to prove. □

The C_{war} operator satisfies left absorption, but not right absorption. This follows from the epistemic status assigned to warranted literals: if they are incorporated as new facts into a given theory, clearly they will be also derivable via SLD inference as in logic programming. The converse is not true, since not every warranted literal is derivable via SLD.

Proposition 5.27 (Left Absorption). *The* $C_{war}(\Gamma)$ *operator satisfies left absorption, i.e.* $Th_{sld}(C_{war}(\Gamma)) = C_{war}(\Gamma)$*.*

Proof. ($\Longrightarrow$): Suppose h is warranted, and consequently $\emptyset{:}h \in C_{war}(\Gamma)$. Clearly $\Gamma \mid\!\sim_{Arg} \emptyset{:}h$, and in particular from Def. 5.1, it follows that $\emptyset{:}h \in Th_{sld}(C_{war}(\Gamma))$.
($\Longleftarrow$): Suppose $\emptyset{:}h \in Th_{sld}(C_{war}(\Gamma))$. Then from def. 5.1 there exists an empty argument $\emptyset{:}h$. Therefore h is warranted, or equivalently $\emptyset{:}h \in C_{war}(\Gamma)$. □

Proposition 5.28 (Right Absorption). *The* $C_{war}(\Gamma)$ *operator does not satisfy right absorption, i.e.* $C_{war}(Th_{sld}(\Gamma)) \neq C_{war}(\Gamma)$*.*

Proof. A counterexample suffices. Consider the theory $\Gamma = \{(p \leftarrow q{:}p \leftarrow q), (\emptyset{:}q)\}$. Clearly p and q are warranted literals. In particular $\emptyset{:}p \in C_{war}(\Gamma)$. Note that $Th_{sld}(\Gamma)=\{\emptyset{:}q\}$, and consequently $C_{war}(Th_{sld}(\Gamma)) = Th_{sld}(\Gamma)$. But $\emptyset{:}p \notin C_{war}(Th_{sld}(\Gamma))$. □

Proposition 5.29 (Rational Negation). *The* $C_{war}(\Gamma)$ *operator does not satisfy rational negation.*

Proof. Consider the example given in Proposition 5.1. In this case there is only one argument for z, namely $\Gamma \mid\!\sim_{Arg} \mathcal{A}{:}z$, and such an argument has no defeaters. Consequently z is warranted wrt Γ. However, $\Gamma \cup \{\emptyset{:}x\} \not\mid\!\sim_{Arg} \mathcal{A}{:}z$, and $\Gamma \cup \{\emptyset{:}{\sim}x\} \not\mid\!\sim_{Arg} \mathcal{A}{:}z$ (since in both cases the use of $\{z \leftarrow p\}{:}z \leftarrow p$ is not allowed by the consistency constraints associated with the deduction rule Elim-$\leftarrow$). Therefore z is not warranted in either of these cases. □

Proposition 5.30 (Rational Monotonicity). *The* $C_{war}(\Gamma)$ *operator does not satisfy rational monotonicity.*

Property	$\mathrel{|\!\sim}_{Arg}$	$\mathrel{|\!\sim}_{\mathcal{T}}$	C	Comments
Inclusion	yes	yes	P	Restricted to non-defeasible information. Propositions 5.3 and 5.17.
Idempotence	yes	yes	P	Propositions 5.4 and 5.18.
Cummulativity	yes	no	P	Lemma 5.7 and proposition 5.21.
Monotonicity	no	no	P	Propositions 5.5 and 5.19.
Horn Supraclassicality	yes	yes	H	Supraclassicality restricted to Horn-like formulas (Prop. 5.8 and 5.22)
Left-logical equivalence	yes	yes	H	Prop. 5.9 and 5.23
Horn Right Weakening	yes	yes	H	Weakening restricted to clauses with one literal in the antecedent (Propositions 5.10 and 5.24)
Conjunction of conclusions	no	no	H	Propositions 5.11 and 5.25.
Subclassical cummulativity	yes	yes	H	Propositions 5.12 and 5.26.
Left absorption	no	yes	H	Propositions 5.13 and 5.27.
Right absorption	no	no	H	Propositions 5.14 and 5.28.
Rational Negation	no	no	NH	Propositions 5.15 and 5.29.
Disjunctive Rationality	no	no	NH	Not considered due to object language constraints.
Rational monotonicity	no	no	NH	Propositions 5.16 and 5.30.

NOTE: Column C denotes the kind of property (P=pure; H=Horn; N=non-Horn)

FIGURE 5. Logical properties in LDS_{ar}: summary

Proof. Consider again the example given in the Proposition 5.1, where $\Gamma \mathrel{|\!\sim}_{Arg} \mathcal{A}{:}z$, with $\mathcal{A} = \{z \leftarrow p, p \leftarrow r\}$. If we consider now $\Gamma \cup \{\emptyset{:}x\}$ it holds that $\Gamma \cup \{\emptyset{:}x\} \not\mathrel{|\!\sim}_{Arg} \mathcal{A}{:}z$, and $\Gamma \not\mathrel{|\!\sim}_{Arg} \sim x$. Therefore rational monotonicity does not hold. $\square$

Finally, note that the C_{war} operator does not satisfy *disjunctive rationality*. The reasons are the same as those discussed for the C_{arg} operator.

5.3. Logical Properties of $\mathbf{C}_{arg}$ and $\mathbf{C}_{war}$: Discussion

As we have shown in this paper, LDS_{ar} provides a useful framework for analyzing different logical properties of defeasible argumentation, providing a better understanding of how argument construction and warrant behave. Figure 5 provides a summary of the logical properties discussed before.

Let us next analyze the implications of some of the properties presented before in the context of $\mathrel{|\!\sim}_{Arg}$. When formalizing argument construction (operator C_{arg}), restricted inclusion ensures that non-defeasible facts (labelled formulas of the form $\emptyset{:}\alpha$, where α is a literal) can be ontologically understood as empty arguments. Cummulativity allows to keep any argument obtained from a theory Γ as an 'intermediate proof' (lemma) to be used in building more complex arguments. Horn supraclassicality indicates that every conclusion that follows via SLD can be considered as a special form of argument (namely, an empty argument), whereas

Horn right weakening tells us that strong rules in LDS_{ar} preserve the intuitive semantics of a Horn rule(the existence of a strong rule $\emptyset{:}y \leftarrow x$ makes every argument $\mathcal{A}$ for x be also an argument for y). Finally, subclassical cummulativity indicates that two argumentative theories Γ and Γ' whose information is a subset of those literals that can be derived via SLD from Γ (or Γ') are equivalent when considering the arguments that can be obtained from them.

Computing warrant, on the other hand, can also be better understood in the light of some logical properties of $C_{_{war}}$. Restricted inclusion ensures that any non-defeasible fact in a theory Γ can be considered as warranted. Idempotence indicates that successive applications of $C_{_{war}}$ on a the set S of warranted literals returns exactly the same set. This makes sense as the formalization of warrant is intended to capture the ultimate acceptance of a piece of information, given the defeasible information at hand. Consequently, once the process of computing warrant has been performed, no additional conclusions can be obtained by repeating applications of the inference rules provided by $\mathrel{|\!\!\sim}_{\mathcal{T}}$. From Horn supraclassicality it follows that every conclusion obtained via SLD is a particular case of warranted literal, whereas Horn right weakening indicates that non-defeasible rules behave as such in the meta-level (a strong rule $\emptyset{:}y \leftarrow x$ ensures that every warrant $\mathcal{A}$ for a literal x is also a warrant for y).

From subclassical cummulativity it follows that two theories Γ and Γ', whose information is a subset of the conclusions that can be obtained from Γ (or Γ') are equivalent when considering the set of literals that can be warranted from them. Finally, left absorption in $C_{_{war}}$ wrt $C_{_{arg}}$ indicates that once a set of warranted literals have been obtained, SLD derivation does not add any inferential power.[17]

In [28, 25, 29] five properties are considered as characterizing the core of so-called Cummulative Nonmonotonic Logics, namely Reflexivity, Left Logical Equivalence, Right Weakening, Cut and Cautious Monotonicity. As the authors put it in [25], these are "rockbottom properties without which a system should not be considered a logical system". In particular, in [25] several families of cummulative logics were defined and characterized, starting by the **C** system, the weakest system in this family of cummulative logics satisfying the above properties. We will focus on the analysis of three particular systems which deserve particular attention, namely **C** (for *cummulative*), **P** (for *preferential*), and **CM** (for *cummulative monotonic*). Clearly, a direct comparison with these families of cummulative logics is not straightforward, as we have seen that LDS_{ar} involves two separate aspects: argument construction (which satisfies cummulativity) and warrant computation (which does not satisfy cummulativity). In our framework, the above properties do hold for the construction of arguments, but not for modelling the computation of warranted conclusions (e.g. cummulativity is not satisfied). This emphasizes the fact that warranted literals have a particular epistemic status which cannot be analyzed at the same representation level as the context provided by the original

[17]This is due to the limitations of our formalism, in the sense that only literals can be warranted.

argumentative theory. The system **P**, which occupies a central position in the hierarchy of non-monotonic systems for preferential reasoning cannot be analyzed in our context, as this system assumes the existence of *disjunction* in the language of formulas (whereas our framework is restricted to rules and facts). Finally, the system **CM** (which is not comparable with the system **P**) is monotonic, differing thus from our approach for both the $\mid\!\sim_{Arg}$ and $\mid\!\sim_{\mathcal{T}}$ inference relationships (as both of them are non-monotonic).

As stated previously, we have restricted the conventional version of Right Weakening to clauses in which the antecedent Y of the rule $X \leftarrow Y$ is formed by only one literal. In the case that Y were formed by more than one literal (i.e. Y is a conjunction of literals $\phi_1, \phi_2, \ldots \phi_k$) and assuming that we consider Y to hold whenever every ϕ_i holds, Right Weakening does not apply (neither for argument construction nor for warrant). This situation can be seen the counterexample used in the proof of Proposition 5.11 where there are arguments $\mathcal{A}_1{:}p$ and $\mathcal{A}_2{:}r$ and a rule $\emptyset{:}{\sim}w \leftarrow p, r$. In that case there are arguments for r and for p (and both p and r are warranted), but there is no argument with conclusion ${\sim}w$ (and hence ${\sim}w$ is not warranted).

6. The Power of Labels for Modelling Argumentation

Labelled Deduction Systems offer a wide range of possibilities for formalizing different aspects of computational models of natural argument. Next we will focus on those which we consider to be particularly relevant (along with the ones presented in the previous sections), namely detecting fallacies, formalizing argumentation in social contexts, and weighing arguments in dialectical trees.

6.1. Detecting Fallacies

According to Hamblin, the classical definition of a fallacy is *"an argument that appears to be valid, but is not"* [20, p.12]. In more general terms, a fallacy is a general type of appeal (or category of argument) that resembles good reasoning, but some of their inference steps are not truth-preserving.[18] As pointed out in [41], while we may say that an argument is "fallacious", or "commits a fallacy", the term "fallacy" does not refer to an argument, but to an error of some identifiable kind. All of the arguments that are guilty of committing that error may be said to be *instances* of that fallacy, so fallacies are strictly and classically considered to be *types* of arguments.

Detecting logical fallacies plays an important role in computational models of argument. In this context the most basic fallacy involves *"circular reasoning"*, or repetition of arguments in a dialogue (as this leads to infinite branches in dialectical trees). Such situation is explicitly avoided in most formal approaches to

[18]Some authors (*e.g.*, Johnson [23]) suggest that a fallacy should occur *"with sufficient frequency in discourse to warrant being baptized."*. An in-depth treatment of fallacies is outside the scope of this paper.

defeasible argumentation (*e.g.*, [17, 22]) by imposing this as a constraint in the definition of argument trees. Other approaches (e.g., [17]) consider avoiding those dialogue lines whenever conflict arises among arguments advanced by the proponent (resp. opponent) in a given dialogue line. In that context, the advanced argument provoking such conflict is considered fallacious. In other cases (such as [24]), analogous situations are obtained as a by-product of the framework under certain constraints (*e.g.*, when characterizing well-founded semantics using an argument-based approach to logic programming, all proponent arguments in an argument tree turn out to be non-conflictive).

In our approach to argumentation using LDS, constraints upon formation of dialogue lines are given by the special condition $\mathsf{VSTree}(\mathcal{A}, \mathbf{T}_i^*)$, which takes into account if a given dialectical label $\mathbf{T}_i^*$ can be used as a sub-tree in a more complex dialectical label rooted in set $\mathcal{A}$ of wffs, corresponding to the main argument at issue. Cycle detection as well as ill-formed dialogue lines (as defined in [17]) are captured by this condition VSTree. Formal results concerning which dialectical trees are valid in a given argumentative framework can also be better analyzed by different characterizations of this condition.

6.2. Formalizing Arguments in Social Contexts

LDS also provide a sound framework for modelling multiagent societies. As Gabbay points out [15, p.311], a label could be the name of a person (source) who put some proposition forward, along with some indicator of the reliability of that person as a source of data. In this context, LDS play a role in formalizing *source-based arguments* [45], *i.e.* arguments whose evaluation depends not only on the structure of the inference used, but also on some assessment of the sources of the premises. Evaluation of source-based arguments is clearly important in the context of computational models of argumentation for multiagent systems. In a very interesting paper [46] Walton shows how LDS and multi-agent systems can be combined to evaluate argumentation that is source-based and depends on a credibility function. He also remarks that two of the most common forms of source-based arguments are appeal to expert opinion (or *ad verecundiam* argument) and personal attacks (or *ad hominem* argument). Although such types of argumentation have been acknowledged as informal fallacies, Walton states that both of them can be "quite reasonable in many cases", particularly in legal argumentation contexts. As Walton points out [46, p.66] *"LDS is a big step forward in the evaluation of* ad hominem *and* ad verecundiam *arguments, because it enables us to base our evaluation of such arguments on a label indicating a comparative assessment of the source of the propositions that were put forward".*

The LDS_{ar} framework can be naturally extended to formalize Walton's proposal, keeping at the same time the expressivity to capture the information involved in the dialectical analysis performed by a single agent. The labelled language $\mathcal{L}_{\mathsf{Arg}}$ in LDS_{ar} can in turn be labelled (*e.g.*, with a label (Ag_i, c_i) denoting an agent's name Ag_i and some associated credibility degree c_i), defining a new labelling language $\mathcal{L}_{\mathsf{Ag}}$. Thus a labelled formula in the new language $(\mathcal{L}_{\mathsf{Ag}}, \mathcal{L}_{\mathsf{Arg}})$ could be as follows:

$$(john, 0.7){:}(\mathbf{T}_i^U(\mathcal{A}, \ldots){:}\alpha)$$

denoting that agent *john* with a credibility degree of 0.7 has performed some dialectical analysis concluding that α is currently assumed as warranted belief, on the basis of a dialectical analysis stored in the label $\mathbf{T}_i^U(\mathcal{A}, \ldots)$. Suitable deduction rules could be defined in order to characterize conflicts among several agents in which their credibility could be a factor to consider in assessing the final outcome of a dialogue among them.

In [1] it was underscored the importance of having a formal model of inter-agent dialogues for argument exchange by providing a precisely defined protocol for interaction. In [36] it was also emphasized that an important challenge facing future research is the understanding of 'social' aspects of argument-based negotiation in agent societies, as *"there is still no generic formal theory that establishes a precise relationship between normative social behavior and the outcomes of communication processes."* Given its expressive power, we think that LDS could provide an adequate formal tool in the context of formalizing protocols and norm adoption, helping to achieve the above goals.

6.3. Pruning and Weighing Arguments in Dialectical Trees

Dialectical trees provide a way of exhaustively analyzing arguments and counter-arguments. A problem with this setting is that dialectical trees can often be "too big" [22] so that the use of some kind of pruning strategy is in order. There are several approaches to pruning the search space in dialectical trees. The most basic approach consists in applying $\alpha - \beta$ pruning, as illustrated in Figure 3. When analyzing a given argument, instead of computing all possible defeaters $(\star)$ only a part of the dialectical tree needs to be explored in order to determine whether the root node (main argument at issue) is defeated or not. It must be noted that the rules that characterize the "$\mathrel{|\!\!\sim}_{\mathcal{T}}$" relationship (Figure 2) are also based on this strategy, used when propagating marking in labels in a bottom-up fashion.

Recent research [22] has been focused on analyzing the *impact* of argumentation. Such an impact depends on what an agent regards as important, which allows to characterize the *resonance* and *cost* of producing arguments and argument trees. To measure resonance in argument trees, the sum of the resonance of the arguments in the tree is taken into account, scaled by a discount function which increases going down the tree, so that arguments at a greater depth have a reduced net effect on the resonance of the tree. The first ideas underlying this approach can be found in [3], where the notion of *categoriser* is introduced. A categoriser is a mapping from dialectical trees to numbers. The resulting number is intended to capture the relative strength of an argument taking into account its defeaters, the defeaters for those defeaters, and so on. An example of categoriser provided in [3] is the following:

$$h(N) = \frac{1}{1 + h(N_1) + \ldots + h(N_l)}$$

where $N_1, \ldots, N_l$ are the children nodes for l (if $l = 0$, $h(N_1) + \ldots + h(N_l) = 0$).

In the context of LDS_{ar}, assessing a weight to an argument on the basis of its defeaters can be performed in a natural way by suitably extending the criteria for labelling propagation. A function f could be defined to assign numbers to dialectical labels according to some particular criterion. Given a dialectical label, if $\mathcal{A}$ corresponds to an argument without defeaters it would be assigned a particular value $f(\mathcal{A})$. Otherwise, if $\mathcal{A}$ is the root node in a dialectical tree with label $\mathbf{T}(\mathcal{A}, \mathbf{T}_1, \ldots, \mathbf{T}_k)$, having as defeaters arguments $\mathcal{B}_1 \ldots, \mathcal{B}_k$, then f could be recursively defined as $f(\mathcal{A})$=$f(f(\mathbf{T}_1), ... f(\mathbf{T}_k))$ where $\mathbf{T}_1, \ldots, \mathbf{T}_k$ are the immediate subtrees (dialectical labels) associated with $\mathbf{T}$. In other words, numbers assigned to dialectical labels would be propagated bottom-up. Computing f can be thus defined in several ways (*e.g.*, as suggested in [3]). Such a setting allows to model a number of typical problems in defeasible argumentation, such as the the notion of *accrual of arguments* [44, 42], where arguments with many defeaters would be deemed weaker as those which have only one defeater.

7. Related Work

Research in logical properties for defeasible argumentation was started by G.Vreeswijk [43] and H.Prakken [35]. In particular, the work of S. Sardiña [38] focused on logical properties of the original Simari-Loui framework [40] and of *defeasible logic programming* [17]. This research is partly motivated by these results.

Early work which used some of the principles present in LDS (but not as formally) was Cohen's theory of endorsements [12]. Endorsements are symbolic representations of different items of evidence, the questions on which they bear, and the relations between them. Endorsements can operate on each other and hence lead to the retraction of conclusions previously reached. Research concerning aggregating arguments by incorporating numerical and symbolic features can be traced back to the work of Krause *et al.* [14, 26], where a uniform framework for reasoning with different kinds of *strength* in arguments is described. In particular, a characterization of defeasible reasoning using LDS is due to Hunter [21], who showed how different non-monotonic logics can be characterized in terms of labelling strategies, algebras for labels, proof rules, and preference criteria. In contrast with our approach, Hunter had as a key aim to analyze the notion of preference among different non-monotonic logics. His approach, however, is also argument-based. In [31] a credulous logical system based on LDS is defined, partially based on cummulative default logics. In some respects this approach is related to the one presented in this paper, although does not aim at modelling argumentative reasoning as in our proposal.

Multicontext systems [18] have also been proposed as an alternative framework to LDS to provide contextual reasoning for agents. A context is a triple $C = \langle L, \omega, \Delta \rangle$, where L is a language (*e.g.*, first order logic), ω is the set of axioms for the context and Δ is the set of inference rules associated with the context C. A multicontext system is a pair $\langle C, B \rangle$, where $C = \{c_1, \ldots, c_n\}$ is the set of all contexts, and B is the set of *bridge rules* which have the form

$c_1 : \phi_1 \ldots, c_k : \phi_k \rightarrow c_i : \phi_i$ standing for "if the wffs $\phi_1 \ldots, \phi_k$ are known to hold in contexts $c_1 \ldots, c_k$, then the wff ϕ_i will hold in context c_i. In [32] a multicontext approach to modelling argumentation among agents is presented. Different contexts represent different components in the agent architecture, and interactions between such components is specified by means of bridge rules between contexts. In [19] an interesting approach to modelling rhetorical argument is presented, in which mental states in an arguer are characterized as a multicontext system $\langle B, R \rangle$, where B is a set of attitude contexts and R is a set of bridge rules among them. We contend that similar approaches to the ones mentioned above can be achieved in terms of the logical machinery provided by LDS. It must be noted that contexts themselves can be recast as elements in the labelling language. Thus, the context notation $B_A : X$ suggested in [19] to denote "agent A believes X" can be seen as a labelled sentence, with B_A as associated label. Nesting beliefs is also possible in the LDS ontology, as suitable functors of the form $B_A(B_B)$ could be defined in the labelling language to denote situations like "agent A believes that agent B believes that...", which in [19] are defined by nesting contexts.

8. Conclusions and Future Work

LDS offer a powerful tool for formalizing different aspects of computational models of argument. In particular, as we have outlined in this paper, LDS_{ar} provides a sound formal framework for modelling argument-based dialectical reasoning. The underlying argumentative logic can be formally analyzed from the natural deduction rules that characterize it, providing a way of studying formal properties associated with such logics. We think that a formal analysis of defeasible consequence is mandatory in order to get an in-depth understanding of the behavior of argumentation frameworks. The logical properties discussed in this paper provide a natural tool for characterizing that behavior, as well as useful comparison criteria when developing new argumentation frameworks, or assessing their expressive power.

We also showed that such framework can be parametrized with respect to a number of features (knowledge representation language, preconditions in natural deduction rules, etc.) which are unified in a single logical system. We have also shown why labels are a good alternative for coping with several issues relevant in modelling argumentation: detecting fallacies, considering arguments in social contexts, analyzing dialectical trees, and formalizing consequence operators.

We contend that several other issues related to computational models of natural argument which have not been explored in this article (*e.g.*, argumentation protocols, resource-bounded reasoning, rhetorical capabilities, etc.) can be suitably modelled in terms of LDS by providing an appropriate ontology in which such notions can be 'abstracted away' as labels. We think that labels are also a good tool in the context of Semantic Web applications, as they can be naturally stored as pieces of structured XML code. On the other hand, the semantic annotation of web content may be stored as labels by means of an appropriate LDS. Different levels of granularity can also be better identified (*e.g.*, abstracting away particular sub-labels), which might be useful for identifying *argumentation schemes* [47] as well as for integrating LDS-based knowledge into salient software tools for argument analysis (such as Araucaria [37]). In our opinion, exploiting such integration can offer promising results that can help in solving several open problems in making formal models of argument computationally attractive. Research in this direction is currently being pursued.

Acknowledgments

The authors would like to thank anonymous reviewers for their comments which helped improve the original version of this article. This research was funded by CONICET (Argentina), by projects TIC2003-00950 and TIN2004-07933-C03-03 (MCyT, Spain), by Ramón y Cajal Program (MCyT, Spain) and by Agencia Nacional de Promoción Científica y Tecnológica (PICT 13096, PICT 15043, PAV 076).

References

[1] L. Amgoud, N. Maudet, and S. Parsons. Modeling dialogues using argumentation. In *ICMAS*, pages 31–38, 2000.

[2] G. Antoniou. *Nonmonotonic Reasoning*. The MIT Press, 1996.

[3] P. Besnard and A. Hunter. A logic-based theory of deductive arguments. *Artif. Intell.*, 128(1-2):203–235, 2001.

[4] J. Beziau. Universal logic. In *Proceedings of the 8th International Colloquium Logica '94*, pages 73–93. Czech Academy of Sciences, Prague, 2004.

[5] J. Beziau, editor. *Logica Universalis: Towards a General Theory of Logic*. Birkhäuser Verlag, 2005.

[6] D. Bonevac. *Deduction: Introductory Symbolic Logic (2nd Ed)*. Blackwell Publishers, 2002.

[7] C. Chesñevar. *Formalizing Defeasible Argumentation using Labelled Deductive Systems (in Spanish)*. PhD thesis, Departamento de Ciencias de la Computación. Universidad Nacional del Sur, Bahía Blanca, Argentina, January 2001.

[8] C. Chesñevar, A. Maguitman, and R. Loui. Logical Models of Argument. *ACM Computing Surveys*, 32(4):337–383, December 2000.

[9] C. Chesñevar and G. Simari. Formalizing Defeasible Argumentation using Labelled Deductive Systems. *Journal of Computer Science & Technology*, 1(4):18–33, 2000.

[10] C. Chesñevar and G. Simari. Towards computational models of natural argument using labelled deductive systems. In Chris Reed, editor, *Proc. of the 5th Intl.. Workshop on Computational Models of Natural Argument (CMNA 2005), 19th IJCAI Conf. Edimburgh, UK*, pages 32–39, July 2005.

[11] C. Chesñevar, G. Simari, T. Alsinet, and L. Godo. A Logic Programming Framework for Possibilistic Argumentation with Vague Knowledge. In *Proc. 20th UAI Conf. Banff, Canada*, pages 76–84, July 2004.

[12] P. Cohen. *Heuristic Reasoning about Uncertainty: An Artificial Intelligence Approach*. London, Pitman, 1985.

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

[14] M. Elvang-Gøransson, P. Krause, and J. Fox. Dialectic reasoning with inconsistent information. In *Proc. of the 9th UAI, Washington, USA*, pages 114–121, 1993.

[15] D. Gabbay. *Labelling Deductive Systems (vol.1)*. Oxford University Press (Volume 33 of Oxford Logic Guides), 1996.

[16] D. Gabbay, L. Lamb, and A. Russo. *Compiled Labelled Deductive Systems : A Uniform Presentation of Non-Classical Logics*. Inst of Physics Pub. Inc. Series: Studies in Logic and Computation, 2004.

[17] A. García and G. Simari. Defeasible Logic Programming: An Argumentative Approach. *Theory and Practice of Logic Prog.*, 4(1):95–138, 2004.

[18] F. Giunchiglia, L. Serafini, E. Giunchiglia, and M. Frixione. Non-omniscient belief as context-based reasoning. In *Proc. of 13th IJCAI Conf., Chambery, France*, pages 548–554, 1993.

[19] F. Grasso. A Mental Model for a Rhetorical Arguer. In R. Young F. Schmalhofer and G. Katz, editors, *Proc. of the European Cognitive Science Society Conf.*, June 2004.

[20] C. L. Hamblin. *Fallacies*. Methuen, London, 1970.

[21] A. Hunter. Defeasible Reasoning with structured information. In E Sandewall J Doyle and P Torasso, editors, *Proc. 4th KR*, pages 281–292. M.Kaufmann, 1994.

[22] A. Hunter. Towards Higher Impact Argumentation. In *Proc. 19th AAAI Conf.*, pages 275–280, 2004.

[23] R. Johnson. The blaze of her splendors. In H. Hansen and R. Pinto, editors, *Fallacies: Classical and Contemporary Readings*. Penn State University Press, 1995.

[24] A. Kakas and F. Toni. Computing argumentation in logic programming. *Journal of Logic Programming*, 9(4):515:562, 1999.

[25] S. Kraus, D. Lehmann, and M. Magidor. Nonmonotonic reasoning, preferential models and cumulative logics. *Artif. Intell.*, 44(1-2):167–207, 1990.

[26] P. Krause, S. Ambler, M. Elvang-Gøransson, and J. Fox. A logic of argumentation for reasoning under uncertainty. *Comput. Intell.*, 11:113–131, 1995.

[27] V. Lifschitz. Foundations of logic programs. In G. Brewka, editor, *Principles of Knowledge Representation*, pages 69–128. CSLI Pub., 1996.

[28] D. Makinson. General theory in cummulative inference. In Ginsberg Reinfrank de Kleer and Sandewall, editors, *Nonmonotonic Reasoning*, volume Lecture Notes in Artificial Intelligence Vol. 346, pages 35–110. Springer Verlag, 1989.

[29] D. Makinson. General patterns in nonmonotonic reasoning. In D.Gabbay, C.Hogger, and J.Robinson, editors, *Handbook of Logic in Art. Int. and Logic Prog.*, volume Nonmonotonic and Uncertain Reasoning, pages 35–110. Oxford Univ. Press, 1994.

[30] D. Makinson and K. Schlechta. Floating conclusions and zombie paths: two deep difficulties in the directly skeptical approach to defeasible inference nets. *Artificial Intelligence*, 48(2):199–209, 1991.

[31] S. Modgil. *A Labelled System for Practical Reasoning*. PhD thesis, Imperial College, London, 1999.

[32] S. Parsons, C. Sierrra, and N. Jennings. Agents that Reason and Negotiate by Arguing. *Journal of Logic and Computation*, 8:261–292, 1998.

[33] J. Pollock. A theory of defeasible reasoning. *Intl. Journal of Intelligent Systems*, 6:33–54, 1991.

[34] H. Prakken and G. Sartor. Argument-based extended logic programming with defeasible priorities. *Journal of Applied Non-classical Logics*, 7:25–75, 1997.

[35] H. Prakken and G. Vreeswijk. Logical Systems for Defeasible Argumentation. In D. Gabbay and F.Guenther, editors, *Handbook of Phil. Logic*, pages 219–318. Kluwer, 2002.

[36] I. Rahwan, S. Ramchurn, N. Jennings, P. McBurney, S. Parsons, and L. Sonenberg. Argumentation-based negotiation. *Knowl. Eng. Rev.*, 18(4):343–375, 2003.

[37] C. Reed and G. Rowe. Araucaria: Software for argument analysis, diagramming and representation. *Intl. J. of AI Tools*, 14(3-4):961–980, 2004.

[38] S. Sardiña and G. Simari. Propiedades del Operador de Consecuencia Argumentativo. In *Proc. of the IV Congreso Argentino de Ciencias de la Computación (Neuquén, Argentina)*, October 1998.

[39] G. Simari, C. Chesñevar, and A. García. The role of dialectics in defeasible argumentation. In *XIV Intl. Conf. of the Chilenean Computer Science Society*, November 1994.

[40] G. Simari and R. Loui. A Mathematical Treatment of Defeasible Reasoning and its Implementation. *Artificial Intelligence*, 53:125–157, 1992.

[41] B. Thompson. What is a fallacy. http://www.cuyamaca.net/bruce.thompson/, 2004.

[42] B. Verheij. *Rules, Reasons, Arguments: formal studies of argumentation and defeat.* PhD thesis, Maastricht University, Holland, December 1996.

[43] G. Vreeswijk. *Studies in Defeasible Argumentation.* PhD thesis, Vrije University, Amsterdam (Holland), 1993.

[44] G. Vreeswijk. Abstract argumentation systems. *Artif. Intell.*, 90(1-2):225–279, 1997.

[45] D. Walton. *Ad hominem Arguments.* Univ. of Alabama Press, Tuscaloosa, 1998.

[46] D. Walton. Applying labelled deductive systems and multi-agent systems to source-based argumentation. *J. of Logic and Computation*, 9(1):63–80, feb 1999.

[47] D. Walton and C. Reed. Argumentation schemes and defeasible inferences. In *Proc. ECAI'2002, CMNA Workshop*, pages 45–55, 2002.

Carlos Iván Chesñevar
Artificial Intelligence Research Group
Department of Computer Science – Universitat de Lleida
C/Jaume II, 69 – E-25001 Lleida, Spain, and
Consejo Nacional de Investigaciones Científicas y Técnicas (CONICET)–Argentina
e-mail: `cic@eps.udl.es`

Guillermo Ricardo Simari
Artificial Intelligence Research and Development Laboratory
Department of Computer Science and Engineering – Universidad Nacional del Sur
Av. Alem 1253 - (B8000CPB) Bahía Blanca - Argentina
e-mail: `grs@cs.uns.edu.ar`