

Formalizing Defeasible Argumentation using a Labeled Deductive System

Carlos Iván Chesñevar

Guillermo Ricardo Simari

Departamento de Ciencias de la Computación – Universidad Nacional del Sur
Av. Alem 1253 – B8000CPB Bahía Blanca – REPÚBLICA ARGENTINA
TEL/FAX: (+54) (291) 459 5135/5136 – Email: {cic,grs}@cs.uns.edu.ar

Keywords: argumentation frameworks, labeled deductive systems

Abstract. In the last years there has been an increasing demand of a variety of logical systems, prompted mostly by applications of logic in AI, logic programming and other related areas. Labeled Deductive Systems (LDS) were developed as a flexible methodology to formalize such a kind of complex logical systems.

In the last decade, *defeasible argumentation* has proven to be a confluence point for many approaches to formalizing commonsense reasoning. Different formalisms have been developed, many of them sharing common features.

This paper presents a formalization of an *LDS for defeasible argumentation*, in which the main issues concerning defeasible argumentation are captured within a unified logical framework. The proposed framework is defined in two stages. First, defeasible inference will be formalized by characterizing an *argumentative LDS*. That system will be then extended in order to capture conflict among arguments using a dialectical approach. We also present some logical properties emerging from the proposed framework, discussing also its semantical characterization.

1 Introduction and motivations

A well-known problem in knowledge representation (KR) is that the available information is usually incomplete and potentially inconsistent. In almost every domain of human knowledge, beliefs are subject to change, and this implies that the conclusions to be obtained from them are generally contradictory. This motivates the study and development of KR techniques to cope with this problem.

As pointed out by [Carbogim et al., 2000], two approaches are commonly used to solve the problem of incomplete information. One of them consists in *restoring consistence* (such as in belief revision). The other approach involves performing reasoning by taking into account this potential inconsistency within a ‘rational’ frame. Defeasible argumentation is such an approach, allowing us to reason with incomplete information, building and evaluating relevant arguments which may lead to contradictory conclusions.

Argumentation bears, in many respects, a strong resemblance with the approach used in *truth-maintenance systems* (TMS) proposed originally by [Doyle, 1979] for inconsistency handling. However, there is an important difference: in TMS, the main goal is to keep a ‘trace’ of the reasons used to arrive to a certain conclusion, so that we are able to explain *how* we arrived to it in case a conflict appears. Should a piece of knowledge p be retracted to restore consistency, then the TMS can help us identify which conclusions are to be affected with the removal of p . Argumentation,

on the contrary, aims at clarifying which are the sources of conflict in a rational and methodic way.

Formal argumentation frameworks are characterized by representing certain features of informal argumentation using a formal language, along with an inference mechanism.¹ Although these frameworks differ in their aims and characterization, the notion of *argument* is more or less the same, with a strong resemblance to the notion of *proof* in logic.

The formalization of reasoning with incomplete information has been a major goal in AI since the '80s. Different approaches were developed to capture *non-monotonic reasoning*, such as circumscription [McCarthy, 1980] and default logic [Reiter, 1980], among many others. Argumentation differs from these approaches since it is motivated in *defeasible reasoning*, where tentative conclusions are obtained departing from uncertain or incomplete information. When new information is available, conclusions can no longer be valid. This motivated the development of various *systems for defeasible argumentation* [Loui, 1987], [Simari and Loui, 1992], [García, 1997]. Defeasible argumentation has recently undergone an important cross-breeding with logic programming and legal reasoning resulting in subareas with well-defined profiles and goals [Prakken and Vreeswijk, 1999].

In the last years, there has been a growing interest in the research community in studying defeasible argumentation under a formal setting (see for example [Parsons et al., 1998, Prakken, 2000]). We are concerned in studying these aspects within a logical system, using the *DeLP* framework for defeasible argumentation as a basis [Simari and Loui, 1992, Simari et al., 1994, García, 1997, García et al., 1998]. In our logical framework we want to capture the main issues involved in defeasible argumentation by specifying a suitable underlying logical language and its associated inference rules. In order to accomplish this goal we will make use of *labeled deductive systems* [Gabbay, 1996], or LDS. LDS offer an attractive approach to formalizing complex systems, since they allow to characterize the different components involved in a logical system by using *labels*. Among other advantages, labels allow the possibility of bringing meta-level features into the object level of a given logic (e.g. the use of labels to trace the proof of a goal). Our main motivation is the definition of a LDS to study and compare different formal aspects of defeasible argumentation.

The rest of this paper is structured as follows: section 2 introduces an outline of our approach, as well as a motivating example. Section 3 describes the formalism using for knowledge representation. Section 4 focuses on defining inference for building *arguments*, and section 5 extends the previous notion to handle *dialectical trees*. In section 6 we discuss logical properties of the proposed framework as well as semantical considerations. Section 7 summarizes some of the current research being carried out. Finally, section 8 concludes.

2 An outline of our approach

We want to capture both defeasible knowledge representation and argumentative inference within a *logical system* (Γ, \sim) , in which Γ represents the agent's knowl-

¹ See [Carbogim et al., 2000], [Prakken and Vreeswijk, 1999], [Chesñevar et al., 2000b] for details about different argumentation frameworks.

Argumentative LDS $(\Gamma, \vdash_{\mathcal{T}})$
Defeasible LDS (Γ, \vdash_{Arg})
Knowledge Base Γ

Fig. 1. Formalizing argumentation using LDS

edge base and \vdash stands for a consequence relation. Traditionally, a logical system (Γ, \vdash) allows the inference of new wffs from those available in Γ using the rules of inference that characterize the notion of logical consequence \vdash . In order to formalize defeasible argumentation within a logical system (which involves the well-known problems associated with non monotonic reasoning), we will make use of a LDS. Since the agent's knowledge base Γ will contain incomplete and potentially inconsistent information, our intelligent agent will be provided with a *defeasible LDS* (Γ, \vdash_{Arg}) which will allow him to arrive to tentative conclusions. Those conclusions will correspond to labeled formulas *label:wff*, where *label* will be associated with the notion of *generalized argument* (closely related to the definition given originally in [Simari and Loui, 1992]).

In other words, the consequence relation \vdash_{Arg} will allow our agent to derive labeled wffs having the form *generalized argument:conclusion*, where *generalized argument* will be a wff in a *labeling language* \mathcal{L}_{Labels} , and *conclusion* will be a ground literal in a *knowledge representation language* \mathcal{L}_{KR} . In this setting, a (generalized) argument will represent a tentative proof our intelligent agent can build in order to support p . However, our agent could also be able to build an argument supporting $\sim p$ from the knowledge available in Γ . This leads to a comparative, recursive analysis of arguments in which a given argument should be compared with all those *counter-arguments* which may *defeat* it. To model this process, our approach will consist in extending the consequence relationship \vdash_{Arg} , in order to obtain a new consequence relationship $\vdash_{\mathcal{T}}$. Those wffs derivable from Γ via $\vdash_{\mathcal{T}}$ will have the form *dialectical label:conclusion*, where *dialectical label* stands for the notion of *dialectical tree* [Simari et al., 1994] in an argumentation framework.

The elements in our ontology are summarized in figure 1. The lower level represents the knowledge base Γ , from which our agent will be able to build arguments using a defeasible LDS. In order to decide whether an *argument* is a *warrant* or not, a comparison among arguments is needed, which results in computing an acceptable dialectical tree. This will be captured by the consequence relation $\vdash_{\mathcal{T}}$ within an *argumentative* LDS. Figure 2 shows a sketch of the proposed framework.

2.1 An example

Next we will introduce an example which will be used for introducing different concepts along this paper.

Example 2.1. Consider an intelligent agent which has to control an engine whose performance is determined by three switches *sw1*, *sw2* and *sw3*.² The switches regulate different features of the engine's behavior, such as pumping system and work-

² For the sake of simplicity, we restrict ourselves to propositional language for this example.

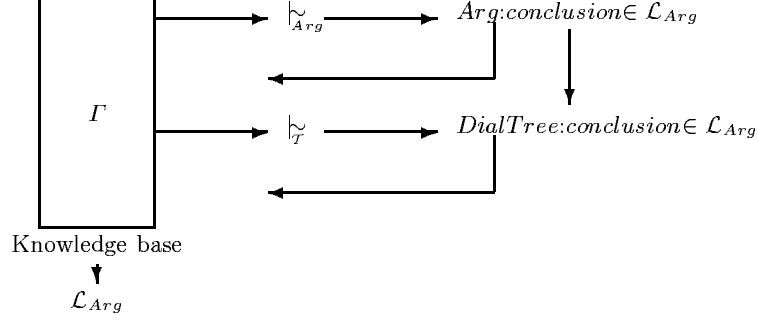


Fig. 2. Sketch of the proposed framework

fuel is ok \leftarrow sw1 is on	$f \leftarrow sw1$
oil is ok \leftarrow sw2 is on	$l \leftarrow sw2$
engine works ok \leftarrow fuel & oil	$ok \leftarrow f, l$
engine not ok \leftarrow fuel, oil & heat	$\sim ok \leftarrow f, l, h$
oil not ok \leftarrow heat	$\sim l \leftarrow h$
fuel not ok \leftarrow low speed & sw1 is on	$\sim f \leftarrow sw1, ls$
speed is low \leftarrow sw2 is on	$ls \leftarrow sw2$
speed is not low \leftarrow sw2 and sw3 are on	$\sim ls \leftarrow sw2, sw3$
sw1 is on	$sw1 \leftarrow$
sw2 is on	$sw2 \leftarrow$
sw3 is on	$sw3 \leftarrow$
there is heat	$h \leftarrow$

Fig. 3. Set Δ (example 2.1)

ing speed. We can model the engine behavior using a *dlp* program (Π, Δ) , where $\Pi = \{(sw1 \leftarrow), (sw2 \leftarrow), (sw3 \leftarrow), (heat \leftarrow)\}$ (specifying that the three switches are on, and there is heat) and Δ specifies defeasible information, modeling the possible behavior of the engine under different conditions (see figure 3).

Using an argumentative framework (such as DELP), the agent in the previous example can build *arguments* for and against the ground literal *engine_ok*. Our goal will be to formalize a logical system in which this process can be modeled. In the following section we will show how to formalize this knowledge base in terms of a labeled language. Then we will introduce a defeasible LDS which will allow the agent to derive different generalized arguments. A global analysis of these generalized arguments will be then performed, which will allow the agent to conclude whether there is (or is not) a *warranted argument* for believing that the engine is working ok.

3 Knowledge Representation

In this section we will introduce a *knowledge representation language* \mathcal{L}_{KR} for performing defeasible inference, together with a *labeling language* \mathcal{L}_{Labels} . These languages will be used to define the object language \mathcal{L}_{Arg} to be used in our defeasible LDS.

Following Gabbay's terminology [Gabbay, 1996], the basic information units in \mathcal{L}_{Arg} will be called *declarative units*, having the form $Label:wff$. In our approach we will restrict wffs in labeled formulas to ground literals. As we will see along this section, a ground literal can be understood as *conclusion* of an *argument*, which will be defined by the label.

A label in a formula $L:\alpha$ will provide three elements which are convenient to take into account when formalizing defeasible argumentation, namely:

1. For every declarative unit $L:\alpha$ the label L will distinguish whether that declarative unit corresponds to *defeasible* or *non-defeasible* information.
2. The label L will also provide an *unique name* to identify a wff in the knowledge base Γ .
3. When performing the inference of a declarative unit $L:\alpha$ from a set Γ of declarative units, the label L will provide a *trace* of the wffs needed in the derivation of $L:\alpha$ from Γ .

Wffs in our knowledge representation language \mathcal{L}_{KR} will be a subset of a classic propositional language \mathcal{L} , restricted to *implications* and *facts*. A modality (label) will be attached to both kinds of wffs: *defeasible* and *non-defeasible*. Formally:

Definition 3.1 (Language \mathcal{L}_{KR}). *The language \mathcal{L}_{KR} will be composed of*

1. *A countable set of propositional atoms, possibly subindicated. We will denote propositional atoms with lowercase letters. Example: $a, b, c, d, e, \dots, a_1, a_2, a_3$ are propositional atoms.*
2. *Logical connectives \wedge, \neg and \leftarrow .*

The set of all atoms in \mathcal{L}_{KR} will be denoted as $\text{Atoms}(\mathcal{L}_{KR})$.

Definition 3.2 (Wffs in \mathcal{L}_{KR}). *Wffs in \mathcal{L}_{KR} will be defined as follows:*

1. *If α is an atom in \mathcal{L}_{KR} , then α and $\sim\alpha$ are wffs called literals in \mathcal{L}_{KR} . We will denote as $\text{Lit}(\mathcal{L}_{KR})$ the set of all literals in \mathcal{L}_{KR} .*
2. *If $\alpha_1, \dots, \alpha_k, \beta$ are literals in \mathcal{L}_{KR} , then $\beta \leftarrow \alpha_1, \dots, \alpha_k$ is a wff in \mathcal{L}_{KR} .*

We will denote as $\text{Wffs}(\mathcal{L}_{KR})$ the set of all wffs in \mathcal{L}_{KR} .

Note that wffs in \mathcal{L}_{KR} can be understood in a logic programming setting: implications correspond to *rules*, and literals can be understood as *facts*.³ Labels add an additional feature to these rules, which can be defeasible or non-defeasible. More precisely, our formalization will follow the approach used in *defeasible logic programming* [García, 1997] for representing knowledge.

For the sake of simplicity, when referring to the language \mathcal{L}_{KR} the following conventions will be used: Greek lowercase letters α, β, γ will refer to any wff in \mathcal{L}_{KR} . Lowercase letters (such as h, q , etc.) will be used for referring to ground literals in \mathcal{L}_{KR} . Greek uppercase letters Υ, Φ, Γ will refer to a *set* of wffs in \mathcal{L}_{KR} . The conjunction $\alpha_1 \wedge \alpha_2 \wedge \dots \wedge \alpha_k$ will be simply written as $\alpha_1, \alpha_2, \dots, \alpha_k$. Separately we will define a labeling language \mathcal{L}_{Labels} , associated with wffs in \mathcal{L}_{KR} .

³ We will consider a negated literal $\sim a$ in \mathcal{L}_{KR} as standing for a new predicate name $\sim a$. As usual, we will assume the equivalence $\sim\sim a = a$.

Definition 3.3 (Labeling constants). A set $Labels = \{ n_1, n_2, \dots, d_1, d_2, \dots \}$ of labeling constants will include constant names having the form n_i or d_i . A set of labeling constants will be denoted as L_1, L_2, \dots, L_k .

Definition 3.4 (Labeling language \mathcal{L}_{Labels}). A label L in our labeling language \mathcal{L}_{Labels} can be either an argument label or a dialectical label, defined as follows:

1. An argument label will be a tuple $\langle L_i, \Phi \rangle$ where $L_i \subseteq Labels$, $\Phi \subseteq \wp(Wffs(\mathcal{L}_{KR}))$. The set of all argument labels that can be defined from $Labels$ and \mathcal{L}_{KR} will be denoted as $ArgumL(Labels, \mathcal{L}_{KR})$.
2. If $\langle L_i, \Phi \rangle$ is an argument label, then $\mathbf{T}_j^U(\langle L_i, \Phi \rangle)$, with $j \in \mathbf{Nat}$ and $\mathbf{T}_k^D(\langle L_i, \Phi \rangle)$, with $k \in \mathbf{Nat}$ are dialectical labels in \mathcal{L}_{Labels} . For the sake of simplicity, we will write \mathbf{T}_k^D to denote a generic dialectical label $\mathbf{T}_k^D(\langle L_i, \Phi \rangle)$, for a given argument label $\langle L_i, \Phi \rangle$. We will also write \mathbf{T}_k to denote either the functor \mathbf{T}_k^D or the functor \mathbf{T}_k^U .
3. If $\mathbf{T}_1, \dots, \mathbf{T}_k$ are dialectical labels, then $\mathbf{T}_n^U(\mathbf{T}_1, \dots, \mathbf{T}_k)$, with $j \in \mathbf{Nat}$, $n \notin \{1 \dots k\}$, and $\mathbf{T}_m^D(\mathbf{T}_1, \dots, \mathbf{T}_k)$, with $k \in \mathbf{Nat}$ $m \notin \{1 \dots k\}$ will also be dialectical labels in \mathcal{L}_{Labels} . The set of all dialectical labels that can be defined from $Labels$ and \mathcal{L}_{KR} will be denoted as $DialectL(Labels, \mathcal{L}_{KR})$.

It should be noted that in order to characterize a *defeasible* LDS argument labels will suffice; dialectical labels will be used in characterizing an *argumentative* LDS, as discussed in section 5.

Definition 3.5 (Defeasible Labeled Language \mathcal{L}_{Arg}). If \mathcal{L}_{Labels} is a labeling language, and \mathcal{L}_{KR} is a knowledge representation language, then the defeasible labeled language, denoted \mathcal{L}_{Arg} , is defined as $\mathcal{L}_{Arg} = (\mathcal{L}_{Labels}, \mathcal{L}_{KR})$

Definition 3.6 (Declarative Unit). Given a language \mathcal{L}_{Arg} , a declarative unit will be a pair $Label:\alpha$, where $Label$ is a label written in the language \mathcal{L}_{Labels} , and α is a wff in \mathcal{L}_{KR} .

If $Label = \langle L_i, \Phi \rangle$ is an argument label, then $Label:\alpha$ will be called an *argumentative declarative unit*. If L_i is a singleton, then $Label:\alpha$ will be called an *atomic declarative unit*.

From now on we will refer to a declarative unit $Label:\alpha$ by the abbreviated form *du*. Greek uppercase letters Γ, Π, Υ will be used to refer to sets of *dus*, when no ambiguity arises.

3.1 Argumentative Theories

Intuitively, a *theory* $\Gamma \subset Wffs(\mathcal{L}_{Arg})$ will constitute the knowledge base from which an intelligent agent will perform its inference process. We will assume that the agent is provided with a logic programming inference mechanism as a basis. Given a set $P \subseteq Wffs(\mathcal{L}_{KR})$, we will write $P \vdash q$ to denote that q follows from P by using SLDNF resolution [Lloyd, 1987]. Furthermore, we will write $P \vdash \perp$ to denote that complementary literals p and $\sim p$ can be derived via a defeasible derivation from P .

A theory Γ will be defined in terms of argumentative *dus*, distinguishing:

- *Non-defeasible* information, given by argumentative *du*s of the form $\langle \{n_i\}, \emptyset \rangle : \alpha$
- *Defeasible* information, given by argumentative *du*s of the form $\langle \{d_i\}, \Phi \rangle : \alpha$

Labeling constants n_i and d_i will denote unique names for declarative units. In this ontology, a declarative unit $\langle \{d_i\}, \{\beta \leftarrow \alpha\} \rangle : \beta \leftarrow \alpha$ will stand for the defeasible rule $\beta \leftarrow \alpha$ in defeasible logic programming [García, 1997].

Example 3.7. Let $Labels = \{n_1, n_2, n_3, n_4, d_1, d_2, d_3, d_4, d_5\}$, and let \mathcal{L}_{KR} be a suitable KR language. The knowledge base involved in example 2.1 can be characterized in terms of the following declarative units:

- **Facts:** $\langle \{n_1\}, \emptyset \rangle : sw1$, $\langle \{n_2\}, \emptyset \rangle : sw2$, $\langle \{n_3\}, \emptyset \rangle : sw3$ and $\langle \{n_4\}, \emptyset \rangle : h$.
- **Rules:**
 - $\langle \{d_1\}, \{f \leftarrow sw1\} \rangle : f \leftarrow sw1$,
 - $\langle \{d_2\}, \{l \leftarrow sw2\} \rangle : l \leftarrow sw2$,
 - $\langle \{d_3\}, \{ok \leftarrow f, l\} \rangle : ok \leftarrow f, l$,
 - $\langle \{d_4\}, \{\sim ok \leftarrow f, l, h\} \rangle : \sim ok \leftarrow f, l, h$,
 - $\langle \{d_5\}, \{\sim l \leftarrow h\} \rangle : \sim l \leftarrow h$,
 - $\langle \{d_6\}, \{\sim f \leftarrow sw1, ls\} \rangle : \sim f \leftarrow sw1, ls$,
 - $\langle \{d_7\}, \{ls \leftarrow sw2\} \rangle : ls \leftarrow sw2$,
 - $\langle \{d_8\}, \{\sim ls \leftarrow sw2, sw3\} \rangle : \sim ls \leftarrow sw2, sw3$

Next we will focus on sets of declarative units (theories) that respect certain requirements to constitute an acceptable knowledge base for an intelligent agent. This situation is formalized through the following definition:

Definition 3.8 (Argumentative theory Γ). Let $\Gamma = \{\gamma_1, \gamma_2, \dots, \gamma_k\}$ be a finite set of declarative units in \mathcal{L}_{Arg} . Let $\Pi(\Gamma)$ be the set of all wffs in \mathcal{L}_{KR} associated with non-defeasible declarative units in Γ , i.e., $\Pi(\Gamma) = \{\alpha \mid \langle \{n_i\}, \emptyset \rangle : \alpha \in \Gamma\}$. We will say that Γ is an argumentative theory if γ_i is an atomic declarative unit, $i = 1 \dots k$, and $\Pi(\Gamma) \not\perp$.

Example 3.9. Consider the set Γ of facts and rules in example 3.7. From definition 3.8, it follows that Γ is an argumentative theory.

The labeling system makes it easier to formalize a proof theory based on natural deduction. For every inference rule, labels propagate information from the premises to the conclusion. In this way, given a *du* $\langle L, \Phi \rangle : \alpha$ that has been inferred from a given theory Γ , the set L will provide a ‘history’ of the proof carried out to conclude α . If $\Phi = \emptyset$, then no defeasible information was needed in order to conclude α . Therefore wffs of the form $\langle L, \emptyset \rangle : \alpha$ will correspond to non-defeasible inferences. On the contrary, $\langle L, \Phi \rangle : \alpha$, $\Phi \neq \emptyset$, denotes a defeasible inference, and Φ is the set of facts, presumptions, defeasible rules and non-defeasible rules needed to conclude α . Intuitively, α will be ‘supported’ by the argument Φ . Some inference rules for labeling will incorporate additional *preconditions* which should be satisfied for an inference rule to be applied. This preconditions are mainly intended for ensuring that the defeasible wff $\langle L, \Phi \rangle : \alpha$ can be inferred only if certain consistency checks are satisfied.

Example 3.10. Consider the argumentative theory Γ according to examples 2.1 and 3.7. Then it should not be valid to infer a labeled wff such as

$$\langle \{n_2, n_3, d_7, d_8\}, \{sw2, sw3, ls \leftarrow sw2, \sim ls \leftarrow sw2, sw3\} \rangle : ls, \sim ls$$

since both ls and $\sim ls$ follow from $\Phi = \{sw2, sw3, ls \leftarrow sw2, \sim ls \leftarrow sw2, sw3\}$.

In the next section we will introduce an inference relation which will allow us to capture the notion of *consistent proof* involving defeasible information. The inference relation will ensure that only consistent proofs can be derived. These consistent proofs will be called *generalized arguments*.

4 Deriving arguments

Our goal will be to define a logical system (Γ, \vdash_{Arg}) , where Γ is a knowledge base as previously described, and \vdash_{Arg} is a consequence relation. The object language will be \mathcal{L}_{Arg} , and inference rules will be formulated in a natural deduction style.⁴

4.1 Natural deduction rules for \vdash_{Arg}

1. **Introducing non-defeasible information:** Any wff in Γ corresponding to non-defeasible information can be introduced in a proof.

$$\frac{}{\Gamma, \langle \{n_i\}, \emptyset \rangle : \alpha}$$

for any $\langle \{n_i\}, \emptyset \rangle : \alpha \in \Gamma$.

2. **Introducing defeasible information:** Any wff in Γ corresponding to defeasible information can be introduced in a proof if it is consistent wrt $\Pi(\Gamma)$.

$$\frac{\Pi(\Gamma) \cup \Phi_1 \not\vdash \perp}{\Gamma, \langle \{d_i\}, \Phi_1 \rangle : \alpha}$$

for any $\langle \{d_i\}, \Phi_1 \rangle : \alpha \in \Gamma$.

3. **Introducing conjunction:** If $\langle L_1, \Phi_1 \rangle : \alpha$ and $\langle L_2, \Phi_2 \rangle : \beta$ are *dus* such that $\Pi(\Gamma) \cup \Phi_1 \cup \Phi_2 \not\vdash p, \sim p$, then the conjunction α, β can be derived. Formally:

$$\frac{\Gamma, \langle L_1, \Phi_1 \rangle : \alpha \quad \langle L_2, \Phi_2 \rangle : \beta \quad \Pi(\Gamma) \cup \Phi_1 \cup \Phi_2 \not\vdash \perp}{\Gamma, \langle L_1 \cup L_2, \Phi_1 \cup \Phi_2 \rangle : \alpha, \beta}$$

4. **Eliminating implication:** As a precondition for applying *modus ponens*, a similar criterion as the one used in the previous rule will be applied.

$$\frac{\Gamma, \langle L_1, \Phi_1 \rangle : \beta \leftarrow \alpha \quad \langle L_2, \Phi_2 \rangle : \alpha \quad \Pi(\Gamma) \cup \Phi_1 \cup \Phi_2 \not\vdash \perp}{\Gamma, \langle L_1 \cup L_2, \Phi_1 \cup \Phi_2 \rangle : \beta}$$

Definition 4.1 (Defeasible LDS). *The pair (Γ, \vdash_{Arg}) defines a defeasible LDS, where $\Gamma \subseteq Wffs(\mathcal{L}_{Arg})$ is an argumentative theory and \vdash_{Arg} is the consequence relation characterized by the inference rules R1, R2, R3 and R4.*

⁴ Inference rules are numbered consecutively for the ease of reading.

Definition 4.2 (Generalized argument). Let Γ be an argumentative theory, and let h be a ground literal such that $\Gamma \sim_{Arg} \langle L, \Phi \rangle : h$. Then $\langle L, \Phi \rangle$ will be called a generalized argument for h .

For the sake of simplicity, and as long as no confusion arises, we will write \mathcal{A} for denoting the set Φ in an argumentative label $\langle L, \Phi \rangle$. Hence we will write an argumentative $\langle L, \Phi \rangle : h$ as $\mathcal{A} : h$. Alternatively, we will say that $\mathcal{A} : h$ is an *argument*.

Given two arguments $\langle L_1, \Phi_1 \rangle : q$ and $\langle L, \Phi \rangle : h$, we will say that $\langle L_1, \Phi_1 \rangle : q$ is a subargument of $\langle L, \Phi \rangle : h$ if $L_1 \subseteq L$ and $\Phi_1 \subseteq \Phi$.

Example 4.3. Consider the argumentative theory Γ as defined in examples 2.1 and 3.7. Then it holds that $\Gamma \sim_{Arg} \langle \{n_1, n_2, d_1, d_2, d_3\}, \mathcal{A}_1 \rangle : ok$, where $\mathcal{A}_1 = \{sw1, sw2, f \leftarrow sw1, l \leftarrow sw2, ok \leftarrow f, l\}$, by the successive application of the rules R1, R1, R4 R4, R3 and R4. Note that this labeled wff corresponds to a generalized argument. Some other generalized arguments that can be derived from Γ via \sim_{Arg} are

$\langle \{n_1, n_2, n_4, d_1, d_2, d_4\}, \mathcal{A}_2 \rangle : \sim ok$ with $\mathcal{A}_2 = \{\sim ok \leftarrow f, l, h, f \leftarrow sw1, l \leftarrow sw2, sw1, sw2, h\}$

$\langle \{n_2, n_3, d_8\}, \mathcal{A}_3 \rangle : \sim ls$ with $\mathcal{A}_3 = \{\sim ls \leftarrow sw2, sw3, sw2, sw3\}$,

$\langle \{n_1, n_2, d_7, d_6\}, \mathcal{A}_4 \rangle : \sim f$ with $\mathcal{A}_4 = \{\sim f \leftarrow sw1, ls, ls \leftarrow sw2, sw1, sw2\}$,

$\langle \{n_4, d_5\}, \mathcal{A}_5 \rangle : \sim l$ with $\mathcal{A}_5 = \{l \leftarrow h, h\}$

Hence we have the generalized arguments $\mathcal{A}_1 : ok$, $\mathcal{A}_2 : \sim f$, $\mathcal{A}_3 : \sim ls$, $\mathcal{A}_4 : \sim ok$ and $\mathcal{A}_5 : \sim l$. Note that $\mathcal{A}_1' : f$ and $\mathcal{A}_2' : ls$ are generalized subarguments of $\mathcal{A}_1 : ok$ and $\mathcal{A}_2 : \sim f$, with $\mathcal{A}_1' = \{f \leftarrow sw1\}$ and $\mathcal{A}_2' = \{ls \leftarrow sw2\}$, respectively.

5 An argumentative LDS

So far a formalization of defeasible inference has been introduced, in which the notion of *generalized argument* has been presented. However, given an argumentative theory Γ an intelligent agent can obtain different, conflicting arguments. Thus our agent could be able to find that $\Gamma \sim_{Arg} \mathcal{A} : h$ and $\Gamma \sim_{Arg} \mathcal{A} : \sim h$. In order to decide among conflicting arguments a preference criterion is needed, as well as a global analysis in which the attack relationships between those conflicting arguments can be captured. This analysis will be formalized in terms of an *acceptable dialectical tree* [Simari et al., 1994]. This will lead to defining a new logical system, an *argumentative* LDS (Γ, \sim_{τ}) .

Conflict between arguments will be captured as follows. An argument $\mathcal{A} : h$ will *counterargue* an argument $\mathcal{B} : q$ whenever there exists a subargument $\mathcal{B}' : q'$ of $\mathcal{B} : q$ such that $\Gamma \cup \{h, q'\} \vdash \perp$. If $\mathcal{A} : h$ is *better* than $\mathcal{B}' : q'$ according to some preference criterion (e.g. specificity, as in [Simari and Loui, 1992]), then $\mathcal{A} : h$ will *defeat* $\mathcal{B} : q$.

Example 5.1. Consider example 4.3. We will assume that the following relations between arguments hold: *Defeats*($\mathcal{A}_2, \mathcal{A}_1$), *Defeats*($\mathcal{A}_3, \mathcal{A}_2$), *Defeats*($\mathcal{A}_4, \mathcal{A}_1$) and *Defeats*($\mathcal{A}_5, \mathcal{A}_4$).

5.1 Building dialectical trees

When defining $\approx_{\mathcal{T}}$, inference rules for constructing *acceptable dialectical trees* are needed [Simari et al., 1994]. As already stated in the introduction, we will conceptualize these trees as dialectical labels. Generalized arguments in a dialectical setting will also be required to be *minimal*, following [Simari and Loui, 1992].

The possibility of building dialectical labels is restricted to certain constraints. On the one hand, the root of any immediate sublabel \mathbf{T}' associated with a dialectical formula $\mathbf{T}:h$ should be a *defeater* for the root of \mathbf{T} . On the other hand, dialectical sublabels should respect constraints known as *concordance* and *no circularity*, in order to avoid fallacious reasoning. Following Gabbay's approach, we capture these constraints by introducing a condition \mathbf{VSTree} , which basically stipulates what kind of labels we want to infer as valid. Next we will briefly sketch the main features of such a condition.

Definition 5.2 (Condition \mathbf{VSTree}).

Let $\mathbf{T}'(\mathcal{B}, \mathbf{T}_1(\mathcal{C}_1), \dots, \mathbf{T}_k(\mathcal{C}_k)):q$ and $\mathbf{T}(\mathcal{A}, \dots):h$ be dialectical formulas. Then \mathbf{T}' will be an immediate dialectical sublabel associated with \mathbf{T} , denoted $\mathbf{VSTree}(\mathbf{T}(\mathcal{A}, \dots):h, \mathbf{T}'(\mathcal{B}, \dots):q)$, iff the following conditions hold:

1. (**Progressive Defeat**)

The wff $\mathcal{B}:q$ should be a defeater for $\mathcal{A}:h$, that is, it should be the case that $\mathcal{B}:q$ defeats $\mathcal{A}:h$.⁵

2. (**Concordance**)

The set S of odd-level (even-level) arguments in every path in the resulting tree should be concordant, i.e., $\Pi(\Gamma) \cup S \not\vdash \perp$

3. (**Non-circularity**)

For every path in \mathbf{T} , it should hold that there is no circularity when building a new path including \mathcal{A} and \mathcal{B} .

Alternatively, if $\mathbf{VSTree}(\mathbf{T}(\mathcal{A}, \dots):h, \mathbf{T}'(\mathcal{B}, \dots):q)$ is satisfied, we will say that the dialectical formula $\mathbf{T}'(\mathcal{B}, \dots):q$ attacks $\mathbf{T}(\mathcal{A}, \dots):h$.

Next we will introduce the inference rules needed for characterizing dialectical trees.

5. **Introducing a dialectical tree:** Given a generalized argument $\mathcal{A}:h$ which is minimal, an *atomic* dialectical tree can be built, in which \mathcal{A} corresponds to the root of the tree.

$$\frac{\mathcal{A}:h \quad \nexists \mathcal{A}' \subseteq \mathcal{A}: \mathcal{A}':h}{\mathbf{T}(\mathcal{A}):h}$$

6. **N-level dialectical tree:** Given an atomic dialectical wff $\mathbf{T}^*(\mathcal{A}):h$, a more complex wff can be built by introducing *all* immediate dialectical sublabels $\mathbf{T}_1^* \dots \mathbf{T}_k^*$ which are valid wrt \mathbf{VSTree} .

⁵ In the final formalization two cases should be considered, blocking and proper defeat. A full analysis is beyond the scope of this paper.

$$\frac{\mathbf{T}^*(\mathcal{A}):h \quad \mathbf{T}_1^*(\mathcal{B}_1, \dots):q_1 \quad \mathbf{T}_k^*(\mathcal{B}_k, \dots):q_k \quad \text{VSTree}(\mathbf{T}^*(\mathcal{A}), \mathbf{T}_i^*)}{\mathbf{T}^*(\mathcal{A}, \mathbf{T}_1^*, \dots, \mathbf{T}_k^*):h}$$

Example 5.3. Consider example 4.3. Then some of the wffs that can be derived from Γ via $\underset{\tau}{\sim}$ are: $\mathcal{T}_1(\mathcal{A}_1):ok$, $\mathcal{T}_2(\mathcal{A}_2):\sim f$, $\mathcal{T}_3(\mathcal{A}_3):\sim ls$, $\mathcal{T}_4(\mathcal{A}_4):\sim ok$ and $\mathcal{T}_5(\mathcal{A}_5):\sim l$ (all of them by applying R5).

Rule R6 allows to build a more complex dialectical label from atomic ones. Thus it holds that $\Gamma \underset{\tau}{\sim} \mathcal{T}_1(\mathcal{A}_1, \mathcal{T}_2(\mathcal{A}_2), \mathcal{T}_4(\mathcal{A}_4)):ok$. From rule R6 it also holds that

$$\Gamma \underset{\tau}{\sim} \mathcal{T}_1(\mathcal{A}_1, \mathcal{T}_2(\mathcal{A}_2, \mathcal{T}_3(\mathcal{A}_3)), \mathcal{T}_4(\mathcal{A}_4)):ok.$$

However, $\Gamma \not\underset{\tau}{\sim} \mathcal{T}_1(\mathcal{A}_1, \mathcal{T}_2(\mathcal{A}_2, \mathcal{T}_3(\mathcal{A}_3)), \mathcal{T}_4(\mathcal{A}_4, \mathcal{T}_5(\mathcal{A}_5))):ok$, since \mathcal{A}_1 is not concordant wrt $\mathcal{T}_5(\mathcal{A}_5)$ (i.e., $\Pi(\Gamma) \cup \mathcal{A}_1 \cup \mathcal{A}_5 \vdash \perp$).

5.2 Marking of a dialectical tree

A wff $\mathbf{T}:h$ stands for an acceptable dialectical tree [Simari et al., 1994] whose root is an argument supporting a ground literal h . However, it is still necessary to determine whether a dialectical tree supports a *justified* ground literal. In order to do this, a *marking* procedure should be applied.

Next we will introduce inference rules for performing that marking procedure on dialectical labels. As a result, we will be able to write \mathbf{T}^U and \mathbf{T}^D , standing for *undefeated* and *defeated* dialectical trees, respectively.

7. **Marking an atomic dialectical wff: (M-Atom)** A dialectical wff is warranted iff there are no valid sublabeled associated with it. Formally:

$$\frac{\mathbf{T}^*(\mathcal{A}):h}{\mathbf{T}^U(\mathcal{A}):h}$$

8. **Marking a dialectical wff as defeated: (M-1D)** A dialectical wff can be marked as defeated if there exists *at least one* immediate sublabeled marked as undefeated.

$$\frac{\mathbf{T}^*(\mathcal{A}, \mathbf{T}_1^*, \dots, \mathbf{T}_i^*, \dots, \mathbf{T}_k^*):h \quad \mathbf{T}_i^U(\mathcal{B}_i, \dots):q_i : \text{VSTree}(\mathbf{T}^*(\mathcal{A}), \mathbf{T}_i^U)}{\mathbf{T}^D(\mathcal{A}, \mathbf{T}_1^*, \dots, \mathbf{T}_{i-1}^*, \mathbf{T}_i^U, \mathbf{T}_{i+1}^*, \dots, \mathbf{T}_k^*):h}$$

para algún \mathbf{T}_i^* , $i = 1 \dots k$

9. **Marking a dialectical tree as undefeated: (M-ND)** A dialectical wff can be marked as undefeated if *every* immediate sublabeled can be marked as defeated.

$$\frac{\mathbf{T}^*(\mathcal{A}, \mathbf{T}_1^*, \dots, \mathbf{T}_i^*, \dots, \mathbf{T}_k^*):h \quad \mathbf{T}_i^D(\mathcal{B}_i, \dots):q_i : \text{VSTree}(\mathbf{T}^*(\mathcal{A}), \mathbf{T}_i^D)}{\mathbf{T}^U(\mathcal{A}, \mathbf{T}_1^D, \dots, \mathbf{T}_i^D, \dots, \mathbf{T}_k^D):h}$$

$\forall \mathbf{T}_i^*$, $i = 1 \dots k$

Example 5.4. Consider the labeled wffs in example 5.3. Then it holds that

$$\Gamma \underset{\tau}{\sim} \mathcal{T}_1^D(\mathcal{A}_1, \mathcal{T}_2^D(\mathcal{A}_2, \mathcal{T}_3^U(\mathcal{A}_3)), \mathcal{T}_4^U(\mathcal{A}_4)):ok$$

In order to derive this labeled wff, the following steps were followed:

- a) $\Gamma \mid_{\sim}^{\tau} \mathcal{T}_3^U(\mathcal{A}_3) : \sim ls$ (by rule R7)
- b) $\Gamma \mid_{\sim}^{\tau} \mathcal{T}_4^U(\mathcal{A}_4) : \sim ok$ (by rule R7)
- c) $\Gamma \mid_{\sim}^{\tau} \mathcal{T}_2^D(\mathcal{A}_2, \mathcal{T}_3^U(\mathcal{A}_3)) : \sim f$ (from (a) by rule R8)
- d) $\Gamma \mid_{\sim}^{\tau} \mathcal{T}_1^D(\mathcal{A}_1, \mathcal{T}_2^D(\mathcal{A}_2, \mathcal{T}_3^U(\mathcal{A}_3)), \mathcal{T}_4^U(\mathcal{A}_4)) : ok$. (from (b) and (c) by rule R8)

The previous marking procedure allows us to define the notions of *justification* and *warrant*. Since reasoning is defeasible, a belief may be justified at one stage, unjustified at a later stage, and so on. Warrant is a less transitory notion of justification, being understood as "justification in the limit". A warranted proposition is one that eventually becomes justified and stays justified. As pointed out by John Pollock [Pollock, 1995], justification and warrant are concepts closely related to each other, but different. In our formalization we capture both concepts.

Definition 5.5 (Justification).

Let Γ be an argumentative theory, such that $\Gamma \mid_{\sim}^{\tau} \mathbf{T}^U : h$. Then the (ground) literal h will be justified, and the label $\mathbf{T}^U : \alpha$ will be called its justification.

Example 5.6. In example 5.4, we have that $\Gamma \mid_{\sim}^{\tau} \mathcal{T}_3^U(\mathcal{A}_3) : \sim ls$. Hence $\sim ls$ is justified.

Warrant will be characterized in terms of the deductive closure of the inference rules for computing justification.

DEFINICIÓN 5.1 (Warrant). Let $\Gamma \mid_{\sim}^{\tau}$ be the closure of Γ under \mid_{\sim}^{τ} . Let $S \subseteq \Gamma \mid_{\sim}^{\tau}$. A literal h is *warranted* iff $\mathbf{T}^U(\mathcal{A}, \dots) : h \in S$, and there is no $S' \supset S$ such that $\mathbf{T}^D(\mathcal{A}, \dots) : h$. \square

Example 5.7. In example 5.4, the labeled wff $\mathcal{T}_1^D(\mathcal{A}_1, \mathcal{T}_2^D(\mathcal{A}_2, \mathcal{T}_3^U(\mathcal{A}_3)), \mathcal{T}_4^U(\mathcal{A}_4)) : ok$ asserts that the argument \mathcal{A}_1 for believing that the engine was working ok is ultimately *defeated* (after considering all arguments involved). Therefore \mathcal{A}_1 is *not* a warrant for ok . From this example it also follows that both the argument \mathcal{A}_4 for $\sim ok$ and the argument \mathcal{A}_3 for $\sim ls$ are warrants (since they are not defeated by any other argument).

Definition 5.1 specifies declaratively how to compute warrant. Procedurally, warrant can be determined by means of a *precedence relation* between dialectical formulas. The basic idea is that new defeaters are added to a given atomic dialectical tree, building a a sequence of dialectical formulas $[\mathbf{T}_1 : h, \mathbf{T}_2 : h, \dots, \mathbf{T}_k : h]$ until any addition of a new argument would imply violating the condition **VSTree**. The root node of the last element of this sequence will be a warranted argument. This procedural definition of warrant can be shown to be equivalent to the one in Definition 5.1. For more details, the interested reader is referred to [Chesñevar, 2000].

6 Logical Properties in \mid_{\sim}^{Arg} . Semantics.⁶

Next we briefly mention some of the logical properties that have been studied for \mid_{\sim}^{Arg} . Let $C(\Gamma) = \{ q_1, \dots, q_k : \Gamma \mid_{\sim}^{Arg} \mathcal{A} : q_1, \dots, q_k \}$ (that is, the consequences of all generalized arguments that can be derived from Γ).

⁶ Proofs of the propositions in this section as well as an in-depth discussion of semantical issues are not included for space limitations. The interested reader is referred to [Chesñevar, 2000].

The notion of generalized arguments satisfies superclassicality wrt provability in logic programs. In other words, all conclusions that can be derived in a logic program can also be derived wrt \sim_{Arg} .

Proposition 6.1. (*Superclassicality of \sim_{Arg} wrt \vdash*) *Let $Th(\Gamma)$ be the conclusions that follow from $\Pi(\Gamma)$. The operator $C(\Gamma)$ satisfies superclassicality, i.e., $Th(\Gamma) \subseteq C(\Gamma)$.*

Note that \sim_{Arg} does not satisfy monotony, since adding new information to Γ can invalidate old generalized arguments. The consequence relation \sim_{Arg} does not satisfy rational negation, nor rational monotonicity, defined as follows.

1. **Rational Negation:** if $A \sim z$ then either $A \cup \{x\} \sim z$ or $A \cup \{\neg x\} \sim z$.
2. **Rational Monotonicity:** If $A \sim z$ then $A \cup \{x\} \sim z$, or $A \sim \neg x$

Proposition 6.2. (*Failure of Rational Negation and Rational Monotonicity in \sim_{Arg}*) *The operator $C(\Gamma)$ does not satisfy rational negation, nor rational monotonicity.*

Generalized arguments can be used as lemmas for building more complex arguments. In other words, the property of cummulativity holds for the inference relation \sim_{Arg} .

Lemma 6.3 (Cummulativity in \sim_{Arg}). *Let Γ be an argumentative theory, and let α_1 and α_2 be wffs in \mathcal{L}_{KR} . Then $\Gamma \sim_{Arg} \mathcal{A}_1:\alpha_1$ implies that*

$$\Gamma \cup \{\mathcal{A}_1:\alpha_1\} \sim_{Arg} \mathcal{A}_2:\alpha_2 \text{ iff } \Gamma \sim_{Arg} \mathcal{A}_2:\alpha_2$$

Many researchers (e.g. [Gabbay, 1996], [Makinson, 1989]) argue that any non-monotonic consequence notion, although lacking the property of monotonicity, should still satisfy other criteria. Others (e.g. [Prakken and Vreeswijk, 1999]) do not follow that approach, in the sense that they think it is hard to find any criterion that should really hold for any argumentation system.

In this respect, we contend that a LDS-based formalization of defeasible argumentation can provide a useful tool for the study of emerging logical properties. As proven in lemma 6.3, the consequence relation \sim_{Arg} satisfies cummulativity. Under certain constraints, the same hold for warranted beliefs. A complete study of these properties is still subject of research work.

Semantical considerations. Transformation properties

When looking for a semantical characterization for an LDS, one natural alternative is to adopt Gabbay's viewpoint, which involves basically to stay within the 'realm of logical systems' (i.e., LDS with mechanisms), and to the extent that semantics is needed, it is brought into the syntax. This can transparently be done in modal logic, where the labels denote possible worlds, and the proof rules closely reflect semantical evaluation rules. Gabbay also suggest a generalization of this approach. The approach used by Modgil [Modgil, 1998] relies upon such a formalization, using possible world semantics.

However, an interesting option consists in determining *frames* (or classes of distinguished models) corresponding to the different proof-theoretic notions in our framework. A semantical account of justification and warrant has been provided using this setting. This work was initially started in [Simari and Loui, 1992]. Many new issues (such as dialectical constraints, which emerged in later research work) were not considered at that time, and are now integrated in the resulting formalization. Soundness and completeness results are provided, which relate the syntactic LDS formalization with a semantical counterpart [Chesñevar, 2000].

The complexity of determining warrant is associated with the number of formulas involved in an argumentative theory. Transformation properties provide a way to reduce a given argumentative theory Γ by repeated application of rewriting rules. The resulting residual theory is equivalent to Γ [Chesñevar et al., 2000a]. These transformation rules are based on similar work done in the area of logic programming, particularly concerning normal logic programs, where a complete simplification of a program can be achieved in order to read off its well-founded semantics. In our analysis, we show that a complete simplification of an argumentative theory cannot be achieved, although our results suggest that defeasible information cannot be easily compressed or re-written.

7 Ongoing work

Dialectics [Rescher, 1977, Hegselmann, 1985] is one of the known processes to arrive to rational beliefs. Dialectics refers to a form of disputation, in which a serializable resource is distributed among two or more parties (usually two parties, proponent and opponent). The serializable resource is typically search for arguments, or time for presentation of arguments [Loui and Norman, 1992]. An important issue in dialectical argumentation concerns the role of *protocols* of dispute within an argument-based framework. The question is whether there are rational principles governing the exchange of arguments and counterarguments in disputational dialogues, and if so, how to formalize them [Prakken, 2000].

The importance of protocols was first pointed out by Loui [Loui, 1998], and Prakken [Prakken, 2000] has also recently been concerned with them. Protocols characterize the way a dialectical proof can be carried out. Unlike proof theories, it does not always make sense to evaluate the outcome of a dispute with respect to all possible arguments; sometimes only those arguments that have actually been stated should be relevant (e.g. a legal procedure). Soundness and fairness can be defined with respect to protocols; sound protocols always prove warranted arguments, whereas fair protocols should allow an argument which is warranted to prevail over others arguments in a dispute.

A partial characterization of a protocol has been provided within our formalization. A partial order “ \sqsubseteq ” on labelled wffs corresponding to dialectical trees was defined. A sequence of dialectical formulas $[\mathbf{T}_1:h, \mathbf{T}_2:h, \dots, \mathbf{T}_k:h]$ such that $\mathbf{T}_i:h \sqsubseteq \mathbf{T}_{i+1}:h$, allows us to represent Prakken’s notion of dispute in our setting. A detailed analysis of this issue is outside the scope of this paper.

8 Conclusions

Labeled Deductive Systems offer a powerful tool for formalizing different logical frameworks. In this paper we have presented a formalization of an argumentative system in terms of LDS. In the proposed setting, the notion of label allows to capture the concept of *argument* as a set of wffs supporting a given proposition. On the other hand, the concept of *dialectical tree* can be also captured by a complex label, defined in terms of more simple ones. Such a complex label can be ‘marked’ in terms of its sublabels, following the procedure initially suggested in [Simari et al., 1994].

It should be remarked that LDS is not a single logic, but a family of logics. Thus, in our formalization, a condition such as *VSTree* can be used as a parameter for characterizing different, alternative logics for defeasible argumentation within the same logical framework. Having a formal system that models the process of defeasible argumentation allows us to analyze different aspects associated with characterizing argumentative frameworks (such as argumentation protocols and resource-bounded reasoning). Research in this direction is currently being pursued.

References

- [Bondarenko et al., 1997] Bondarenko, A., Dung, P., Kowalski, R., and Toni, F. (1997). An abstract, argumentation-theoretic approach to default reasoning. *Artificial Intelligence*, 93(1-2):63–101.
- [Bondarenko et al., 1993] Bondarenko, A., Toni, F., and Kowalski, R. (1993). An assumption-based framework for non-monotonic reasoning. *Proc. 2nd. International Workshop on Logic Programming and Non-monotonic Reasoning*, pages 171–189.
- [Brewka, 1999] Brewka, G. (1999). Dynamic argument systems: a formal model of argumentation processes based on situation calculus. *Journal of Logic and Computation (submitted)*.
- [Carbogim et al., 2000] Carbogim, D., Robertson, D., and Lee, J. (2000). Argument-based applications to knowledge engineering. *The Knowledge Engineering Review (to appear)*.
- [Chau, 1991] Chau, H. F. (1991). Modalising resource logics in labelled deductive systems. Master’s thesis, Department of Computing, Imperial College, London (United Kingdom).
- [Chesñevar, 2000] Chesñevar, C. I. (2000). *Una formalización de la argumentación rebatible para su tratamiento computacional*. Universidad Nacional del Sur, Bahía Blanca, Argentina. PhD Thesis in preparation; draft available from cs.uns.edu.ar/~cic.
- [Chesñevar et al., 2000a] Chesñevar, C. I., Dix, J., Stolzenburg, F., and Simari, G. R. (2000a). Relating Transformation Properties in NLP and Defeasible Logic Programming. In *Proceedings of the VI Congreso Argentino de Ciencias de la Computación*, pages 371–382. Univ. Nacional de la Patagonia, Ushuaia, Argentina.
- [Chesñevar et al., 2000b] Chesñevar, C. I., Maguitman, A., and Loui, R. (2000b). Logical Models of Argument. *ACM Computing Surveys*, 32(4).
- [Chesñevar and Simari, 1998] Chesñevar, C. I. and Simari, G. R. (1998). Formalization of defeasible argumentation using labelled deductive systems. In *Proceedings of the IV Congreso Argentino en Ciencias de la Computación*, pages 1047–1059. Univ. Nac. del Comahue, Neuquén.
- [Doyle, 1979] Doyle, J. (1979). A Truth Maintenance System. *Artificial Intelligence*, 12(3):231–272.
- [Dung et al., 1996] Dung, P. M., Kowalski, R., and Toni, F. (1996). Argumentation-theoretic proof procedures for non-monotonic reasoning. In *Proc. 6th Logic Programming Synthesis and Transformation*.
- [Gabbay, 1987] Gabbay, D. (1987). Theoretical foundations of nonmonotonic logics. In *Proceedings of the Symposium on Logics in Computer Science*. Ithaca, NY.
- [Gabbay, 1996] Gabbay, D. (1996). *Labelling Deductive Systems (vol.1)*. Oxford University Press (Volume 33 of Oxford Logic Guides).
- [García et al., 1998] García, A., Simari, G. R., and Chesñevar, C. I. (1998). An argumentative framework for reasoning with inconsistent and incomplete information. In *Proceedings of the ECAI’98 Workshop on Practical Reasoning and Rationality*, pages 13–20. Brighton, UK.
- [García, 1997] García, A. J. (1997). Programación en lógica rebatible: su definición (msc thesis). Master’s thesis, Departamento de Cs. de la Computación, Universidad Nacional del Sur, Bahía Blanca, Argentina.

- [Hegselmann, 1985] Hegselmann, R. (1985). *Formale Dialektik: Ein Beitrag zu einer Theorie des rationalen Argumentierens*. Felix Meiner Verlag.
- [Kowalski and Toni, 1996] Kowalski, R. and Toni, F. (1996). Abstract Argumentation. *Artificial Intelligence and Law*, 4(3-4):275–296.
- [Kraus et al., 1990] Kraus, S., Lehmann, D., and Magidor, M. (1990). Nonmonotonic reasoning, preferential models and cumulative logics. *Artificial Intelligence*, 44:167–207.
- [Lloyd, 1987] Lloyd, J. W. (1987). *Foundations of Logic Programming*. Springer-Verlag, New York, second edition edition.
- [Loui, 1987] Loui, R. P. (1987). Defeat Among Arguments: A System of Defeasible Inference. *Computational Intelligence*, 3(3).
- [Loui, 1998] Loui, R. P. (1998). Process and policy: resource-bounded, non-demonstrative reasoning. *Computational Intelligence*, 2(5).
- [Loui and Norman, 1992] Loui, R. P. and Norman, J. (1992). Rationales and argument moves. Technical report, Washington University, Dept. of Computer Science, St.Louis, USA.
- [Makinson, 1989] Makinson, D. (1989). General theory of cumulative inference. *Proc. of the Second International Workshop on Non-Monotonic Reasoning. Lectures Notes in Artificial Intelligence*, 346:1–18.
- [McCarthy, 1980] McCarthy, J. (1980). Circumscription – A Form of Non-monotonic Reasoning. *Artificial Intelligence*, 13:27–39,171–172.
- [Modgil, 1998] Modgil, S. (1998). *A Labelled System for Practical Reasoning*. PhD thesis, Imperial College, London.
- [Parsons et al., 1998] Parsons, S., Sierra, C., and Jennings, N. (1998). Agents that Reason and Negotiate by Arguing. *Journal of Logic and Computation*, 8:261–292.
- [Pollock, 1987] Pollock, J. L. (1987). Defeasible Reasoning. *Cognitive Science*, 11:481–518.
- [Pollock, 1995] Pollock, J. L. (1995). *Cognitive Carpentry: A Blueprint for How to Build a Person*. Massachusetts Institute of Technology.
- [Prakken, 2000] Prakken, H. (2000). Relating protocols for dynamic dispute with logics for defeasible argumentation. *Synthese (to appear)*.
- [Prakken and Sartor, 1997] Prakken, H. and Sartor, G. (1997). Argument-based extended logic programming with defeasible priorities. *Journal of Applied Non-classical Logics*, 7:25–75.
- [Prakken and Vreeswijk, 1999] Prakken, H. and Vreeswijk, G. (1999). Logics for Defeasible Argumentation. In Gabbay, D., editor, *Handbook of Philosophical Logic*. Kluwer Academic Publisher.
- [Prawitz, 1965] Prawitz, D. (1965). *Natural Deduction - A proof-theoretical study*. Alquist and Wiksell.
- [Reiter, 1980] Reiter, R. (1980). A Logic for Default Reasoning. *Artificial Intelligence*, 13(1,2):81–132.
- [Rescher, 1977] Rescher, N. (1977). *Dialectics, a Controversy-Oriented Approach to the Theory of Knowledge*. State University of New York Press, Albany, USA.
- [Sardiña and Simari, 1998] Sardiña, S. and Simari, G. R. (1998). Propiedades del Operador de Consecuencia Argumentativo. In *Anales del IV Congreso Argentino de Ciencias de la Computación (Neuquén, Argentina)*.
- [Simari and Chesñevar, 1995] Simari, G. R. and Chesñevar, C. (1995). Arguments and contexts. In *CLEI'95, Canela, Brasil*, pages 779–790.
- [Simari et al., 1994] Simari, G. R., Chesñevar, C. I., and García, A. J. (1994). The role of dialectics in defeasible argumentation. In *Anales de la XIV Conferencia Internacional de la Sociedad Chilena para Ciencias de la Computación*. Universidad de Concepción, Concepción (Chile).
- [Simari and Loui, 1992] Simari, G. R. and Loui, R. P. (1992). A Mathematical Treatment of Defeasible Reasoning and its Implementation. *Artificial Intelligence*, 53:125–157.
- [Verheij, 1996] Verheij, B. (1996). *Rules, Reasons, Arguments: formal studies of argumentation and defeat*. PhD thesis, Maastricht University, Holland.