Inteligencia Artificial 16(52) (2013), 29-41

INTELIGENCIA ARTIFICIAL

http://journal.iberamia.org/

# A defeasible logic programming with extra meta-level information through labels

Maximiliano C. D. Budán[1,2,3], Mauro Gómez Lucero[2] and Guillermo R. Simari[2]

[1] Argentinean National Council of Scientific and Technical Research (CONICET), Argentina
[2] AI Research and Development Laboratory (LIDIA) – Universidad Nacional del Sur, Argentina
[3] Universidad Nacional de Santiago del Estero, Argentina
E-mail: {mcdb,mjg,grs} @cs.uns.edu.ar

**Abstract** Several argument-based formalisms have emerged with application in many areas, such as legal reasoning, intelligent web search, recommender systems, autonomous agents and multi-agent systems. In decision support systems, autonomous agents need to perform epistemic and practical reasoning; the first requiring reasoning about what to believe, and the latter, involving reasoning about what to do reaching decisions and, often, attaching more information to the pieces of knowledge involved.

We will introduce an approach in the framework of DeLP called *Argumentative Label Algebra* (ALA), incorporating labels as a medium to convey meta-level information; through these labels it will represent different features of interest in the reasoning process, such as strength and weight measures, time availability, degree of reliability, *etc.* The labels associated with arguments will thus be combined and propagated according to argument interactions. This information can be used for different purposes: to carry information for a specific purpose, to determine which argument defeats another, analyzing a feature that is relevant to the domain, and to define an acceptability threshold which will determine if the arguments are strong enough to be accepted.

The aim of this work is to improve the ability of representing real-world scenarios in argumentative systems by modeling different arguments attributes through labels.

**Resumen** Varios formalismos basados en argumentos han emergido, con aplicaciones en muchas áreas, tales como el razonamiento legal, la búsqueda inteligente en la web, sistemas de recomendación, agentes autónomos y sistemas multi-agente. En los sistemas de soporte a la decisión, los agentes autónomos necesitan realizar razonamiento epistémico y práctico, el primero requiere razonamiento sobre qué creer, y el último involucra razonamiento acerca de qué hacer, frecuentemente, agregando más información a las piezas de conocimiento involucradas.

Introduciremos una aproximación en el marco de DeLP denominada Álgebra de Etiqueta para Argumentos (AEA), incorporando etiquetas como un medio para transmitir información de meta-nivel. A través de estas etiquetas se puede representar diferentes rasgos de interés en el proceso de razonamiento, tales como las medidas de peso y fuerza, disponibilidad de tiempo, grados de confiabilidad, etc. Las etiquetas asociadas con los argumentos podrán así ser combinadas y propagadas de acuerdo a las interacciones de los argumentos. Esta información puede ser usada para diferentes propósitos: llevar información para un objetivo específico, determinar cuáles argumentos derrotan a otros, analizar un rasgo que es relevante a un dominio, y definir un umbral de aceptabilidad que determinará si un argumento es lo suficientemente fuerte como para ser aceptado.

El objetivo de este trabajo es mejorar la habilidad de representar escenarios del mundo real en sistemas argumentativos al modelar diferentes atributos de los argumentos a través de las etiquetas.

**Keywords**: Argumentation, Meta-level information, Argumentative Label Algebra, Defeasible Logic Programming.

**ISSN: 1988-3064(on-line)**

**Palabras Clave**: Argumentación, información de meta-nivel, Álgebra de Etiqueta para Argumentos, Programación en Lógica Rebatible

# 1 Introduction

Argumentation has contributed to the AI community with a human-like mechanism to the formalization of commonsense reasoning; concisely, is the process of defending a given affirmation by giving reasons for its acceptance [4, 16]. Both, the original claim and its support are subject to scrutiny since reasons supporting conflicting claims can be advanced. Several argument-based formalisms have been proposed dealing with applications in many areas such as legal reasoning, intelligent web search, recommender systems, autonomous agents and multi-agent systems. An agent may use argumentation to perform individual reasoning to resolve over contradictory evidence or to decide between conflicting goals [1, 2]; multiple agents may also use dialectical argumentation to identify and reconcile differences between themselves, through interactions such as negotiation, persuasion, and joint deliberation [12, 19, 17, 15].

Besides abstract argumentation approaches concrete argumentation systems exists dealing with discernable arguments, specifying a knowledge representation language, and how arguments are built. One of those systems is Defeasible Logic Programming (DeLP) [10], a formalism that combines elements of Logic Programming and Defeasible Argumentation. DeLP permits the representation of information in the form of weak rules in a declarative way, from which arguments supporting conclusions are constructed, providing also a defeasible argumentation inference mechanism for determining warranted conclusions. The defeasible argumentation nature of DeLP is appropriated to build applications dealing with incomplete and contradictory information in dynamic domains.

In real-world domains, argumentation may be required to explicitly handle special features such as degree of reliability, weight, strength, probabilities, among others. Generally, this type of information is not directly associated with arguments, but instead it is attached to the basic pieces of knowledge from which arguments are built. Adding meta-level information to the argumentation reasoning process in the form of labels attached to arguments will enhance the representational capabilities of the framework; a reason for this extension is that, besides the all-important property of soundness of an argument, as we mentioned, there might be other considerations to take into account, as each argument may have associated particular characteristics like strength [3], Possibilistic Weights varying over time [11], or reliability varying over time [5]. Labels can be defined by a set of characteristics that is important to associate with an argument and the interaction between arguments, such as support and conflict, can affect these labels.

We present a framework called *Argumentative Labels Algebra* (ALA), which integrates the handling of labels. Through this device, we will establish argument acceptability, where the final labels propagated to the accepted arguments provide additional acceptability information, such as degree of justification, restrictions on justification, explanation, etc. This general framework will be instantiated with a weight measure associated with the arguments that represent the force of an argument.

The central contribution of this paper is to increase the ability of real-world representation by modeling different attributes associated to the arguments, and using an algebra of labels for propagate this information. Below we will present an intuitive example to motivate and illustrate the goals of our work. The rest of the paper is organized as follows. In Section 3, a brief introduction to DeLP is given. In Section 4, we we will present the formalism ALA. Then, in Section 5, we will apply ALA to DeLP modeling the example presented in Section 2. Finally, in Section 6 and 7, related work and conclusions are introduced.

# 2 Motivation Example

This work aims to contribute to the successful integration of argumentation in different artificial intelligence applications, such as Autonomous Agents in Decision Support Systems, Knowledge Management, and others of similar importance. Now, we will illustrate the usefulness of our formalization for a particular agent decision-making problem, where the reasons supporting decisions have different weights. Let's consider the following scenario:

*Brian is looking for an apartment to rent, and when he is considering one of the candidate apartments he analyzes different arguments in favor and against renting it:*

- $\mathcal{A}$ *The apartment is in a good location and quiet area; therefore, he should rent it.*
- $\mathcal{B}$ *The apartment is rather small; therefore, he should not rent it.*
- $\mathcal{C}$ *The apartment seems to have mold problems; therefore, he should not rent it.*
- $\mathcal{D}$ *The area will not be quiet anymore because of the increase on the number of students living there.*
- $\mathcal{E}$ *The area is quiet, because most of the neighbors are retirees, and peaceful people.*

This example illustrates how the knowledge used to reach decisions can be naturally structured as arguments. These arguments interact in various ways such as support (*e.g.*, $\mathcal{E}$ supports $\mathcal{A}$) and conflicts involving arguments that support contradictory conclusions (*e.g.*, $\mathcal{A}$ and $\mathcal{C}$). Each of the arguments described above can be considered as having different weights according to the agent preferences; for instance, the agent can see $\mathcal{A}$ as more significant than $\mathcal{C}$. Thus, the example illustrates the need of considering meta-information associated with arguments; each argument is associated with a weight, and this information must be propagated by the relationships of support and conflicts among arguments.

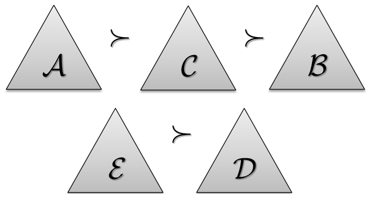


Figure 1: Preference between the arguments

In classical argumentation theory, when two arguments $\mathcal{X}$ and $\mathcal{Y}$ are in conflict, where $\mathcal{X}$ attacks $\mathcal{Y}$ and $\mathcal{Y}$ is stronger than $\mathcal{X}$, then $\mathcal{Y}$ is undefeated. In certain application domains, where a more complex treatment of conflict evaluation is needed, is necessary to capture that $\mathcal{X}$ is somewhat affected the strength of $\mathcal{Y}$; for instance, the weight of an argument should not be the same when it is free of counter-arguments than when it is controversial. That is, we need to model the weakening effect over the undefeated argument achieved by its counter-arguments. In our example, we regard the argument for renting the apartment ($\mathcal{A}$) as stronger than all the arguments for not renting it ($\mathcal{B}$, $\mathcal{C}$, and $\mathcal{D}$). Following the classical approach to the resolution of conflicts, the argument $\mathcal{A}$ remains undefeated. However, it is interesting to consider that the arguments $\mathcal{B}$, $\mathcal{C}$, and $\mathcal{D}$, even when they are not strong enough to defeat $\mathcal{A}$, at least should affect the strength of $\mathcal{A}$ which is weaken by the existence of its counter-arguments.

We propose a general framework allowing to represent meta-information associated with arguments through labels, and defining acceptability by combining and propagating labels according to the support and conflict interactions. We will finally instantiate the proposed formalization to model the example presented in this section.

# 3 Defeasible Logic Programming (DeLP)

There exist different argumentation systems, each specifying a knowledge representation language and how arguments are built; *Defeasible Logic Programming (DeLP)* [10] is one of them. The formalism combines elements of Logic Programming and Defeasible Argumentation allowing the representation of information as weak and strict rules in a declarative way, arguments supporting conclusions are constructed from them, and uses a defeasible argumentation inference mechanism to obtain *warranted* conclusions. The defeasible argumentation basis of DeLP allows to build applications dealing with incomplete and contradictory information in dynamic domains; thus, the resulting approach is suitable for representing agents' knowledge and for providing an argumentation based reasoning mechanism to these agents. Below we present the core definitions of *program* and *argument* in DeLP as introduced in [10].

**Definition 1 (DeLP Program)** *A DeLP program $\mathcal{P}$ is a pair* $(\Pi, \Delta)$ *where (1)* $\Delta$ *is a set of defeasible rules of the form* $L \prec P_1, \ldots, P_n$*, with* $n > 0$*, where* $L$ *and each* $P_i$ *are literals, and (2)* $\Pi$ *is a set of strict rules of the form* $L \longleftarrow P_1, \ldots, P_n$*, with* $n \geq 0$*, where* $L$ *and each* $P_i$ *are literals.* $L$ *is a ground atom* $A$ *or a negated ground atom* $\sim A$*, where '*$\sim$*' represents the strong negation.*

Pragmatically, strict rules can be used to represent strict (non defeasible) information, whereas defeasible rules are used to represent tentative or weak information. In particular, a strict rule $L \longleftarrow P_1, \ldots, P_n$ with $n = 0$ is called *fact* and will be denoted just as $L$, and a defeasible rule $L \prec P_1, \ldots, P_n$ with $n = 0$ is called *presumption* and will be denoted just as $L \prec$ . It is important to remark that the set $\Pi$ must be consistent as it represents strict (undisputed) information. In contrast, the set $\Delta$ will generally be inconsistent, since it represents tentative information.

We say that a set of DeLP rules is contradictory iff exists a defeasible derivation for a pair of complementary literals (w.r.t. strong negation) from this set.

**Definition 2 (Argument)** *Let* $L$ *be a literal and* $\mathcal{P} = (\Pi, \Delta)$ *be a DeLP program. An* argument *for* $L$ *is a pair* $\langle \mathcal{A}, L \rangle$*, where* $\mathcal{A}$ *is a set of defeasible rules of* $\Delta$*, such that:*

(1) *there is a defeasible derivation for* $L$ *from* $\Pi \cup \mathcal{A}$*.*

(2) $\Pi \cup \mathcal{A}$ *is non-contradictory, and*

(3) $\mathcal{A}$ *is a minimal, i.e., there exist no* $\mathcal{A}' \subset \mathcal{A}$ *satisfying conditions* (1) *and* (2)*.*

*We say that an argument* $\langle \mathcal{B}, Q \rangle$ *is a* sub-argument *of* $\langle \mathcal{A}, L \rangle$ *iff* $\mathcal{B} \subseteq \mathcal{A}$*.*

DeLP provides an argumentation based mechanism to determine *warranted* conclusions. This dialectical process involves the construction arguments from programs, the identification of conflicts or *attacks* between arguments, the evaluation of pairs of arguments in conflict to determine if the attack is successful becoming a *defeat*, and finally the analysis of the defeat interaction between all relevant arguments to determine warrant. Below we briefly present the formalization of the previously mentioned notions.

**Definition 3 (Disagreement)** *Let* $\mathcal{P} = (\Pi, \Delta)$ *be a DeLP program. Two literals* $L$ *and* $Q$ *are in* disagreement *iff the set* $\Pi \cup \{L, Q\}$ *is contradictory.*

**Definition 4 (Attack)** *Let* $\mathcal{P} = (\Pi, \Delta)$ *be a DeLP program. Let* $\langle \mathcal{A}, L \rangle$ *and* $\langle \mathcal{B}, Q \rangle$ *be two arguments in* $\mathcal{P}$*. We say that* $\langle \mathcal{A}, L \rangle$ counterargues, rebuts, *or* attacks $\langle \mathcal{B}, Q \rangle$ *at the literal* $R$ *iff there is a sub-argument* $\langle \mathcal{C}, R \rangle$ *of* $\langle \mathcal{B}, Q \rangle$ *such that* $R$ *and* $L$ *are in* disagreement*. The argument* $\langle \mathcal{C}, R \rangle$ *is called* disagreement sub-argument*, and the literal* $R$ *will be the counter-argument point.*

To decide if an attack is successful becoming a defeat, a comparison criterion must be used, establishing the relative strength of the arguments involved in the attack. Here, we will use the default criterion of *generalized specificity* adopted in DeLP [18], which favors arguments based on more information or supporting their conclusions more directly.

**Definition 5 (Defeat)** *Let* $\mathcal{P} = (\Pi, \Delta)$ *be a DeLP program. Let* $\langle \mathcal{A}, L \rangle$ *and* $\langle \mathcal{B}, Q \rangle$ *be two arguments in* $\mathcal{P}$*. We say that* $\langle \mathcal{B}, Q \rangle$ defeats $\langle \mathcal{A}, L \rangle$ *iff exist a sub-argument* $\langle \mathcal{C}, R \rangle$ *of* $\langle \mathcal{A}, L \rangle$ *such that* $\langle \mathcal{B}, Q \rangle$ *counterargues* $\langle \mathcal{A}, L \rangle$ *at literal* $R$ *and it holds that:*

– $\langle \mathcal{B}, Q \rangle$ *is strictly more specific that* $\langle \mathcal{C}, R \rangle$ *(proper defeater), or*

– $\langle \mathcal{B}, Q \rangle$ *is unrelated by the specifity relation to* $\langle \mathcal{C}, R \rangle$ *(blocking defeater)*

In DeLP a literal $L$ will be warranted if there exists a non-defeated argument structure $\langle \mathcal{A}, L \rangle$. In order to establish whether $\langle \mathcal{A}, L \rangle$ is non-defeated, the set of defeaters for $\mathcal{A}$ will be considered. Thus, a complete dialectical analysis is required to determine which arguments are ultimately accepted. Such analysis results in a tree structure called *dialectical tree*, in which arguments are nodes labeled as undefeated (`U-nodes`) or defeated (`D-nodes`) according to a marking procedure.

**Definition 6** Dialectical tree *The* dialectical tree *for an argument* $\langle \mathcal{A}, L \rangle$*, denoted* $\mathcal{T}_{\langle \mathcal{A}, L \rangle}$*, is recursively defined as follows:* (1) *A single node labeled with an argument* $\langle \mathcal{A}, L \rangle$ *with no defeaters (proper or blocking) is by itself the dialectical tree for* $\langle \mathcal{A}, L \rangle$*;* (2) *Let* $\langle \mathcal{A}_1, L_1 \rangle, \langle \mathcal{A}_2, L_2 \rangle, \ldots, \langle \mathcal{A}_n, L_n \rangle$ *be all the defeaters (proper or blocking) for* $\langle \mathcal{A}, L \rangle$*. The dialectical tree for* $\langle \mathcal{A}, L \rangle$*,* $\mathcal{T}_{\langle \mathcal{A}, L \rangle}$*, is obtained by labeling the root node with* $\langle \mathcal{A}, L \rangle$*, and making this node the parent of the root nodes of the dialectical trees for* $\langle \mathcal{A}_1, L_1 \rangle, \langle \mathcal{A}_2, L_2 \rangle, \ldots, \langle \mathcal{A}_n, L_n \rangle$*.*

The marking procedure starts labeling the leaves as `U-nodes`. Then, for any inner node $\langle \mathcal{A}_2, Q_2 \rangle$, it will be marked as `U-node` iff every child of $\langle \mathcal{A}_2, Q_2 \rangle$ is marked as a `D-node`. If $\langle \mathcal{A}_2, Q_2 \rangle$ has at least one child marked as `U-node` then it will be marked as a `D-node`. This marking characterizes the set of literals obtained from a DeLP program, called *warranted literals.* A literal $L$ is *warranted* iff there exists an argument structure $\langle \mathcal{A}, L \rangle$ for $L$, such that the root of its marked dialectical tree $\mathcal{T}^{\star}_{\langle \mathcal{A}, L \rangle}$ is a `U-node`.

# 4 An Argumentative Label Algebra

In certain application domains of argumentation, arguments are naturally associated with meta-information that reflects their weight, strength, reliability, or another important feature for the domain. For instance, in the implementation of an agent, it would be beneficial to establish the degree of success obtained by reaching a given objective. We will use the meta-information for the following purposes:

(1) to carry information for the specific purpose, *e.g.*, to carry conclusive weight,

(2) to determine which argument is defeated or weaken by another taking into account a particular feature that is relevant in the domain, *e.g.*, degree of reliability, weight, or strength, and

(3) to define an threshold for acceptability which will determine when an argument can be accepted, *e.g.*, a lower limit for strength or reliability.

For the first purpose, when through the argumentative process we arrive at a particular conclusion, we also produce a processed label for that conclusion, *e.g.*, the reliability of a recommendation in a recommender system. As for the second requirement, labels associated to the arguments will be used in the resolution of conflicts and in the comparison between them, *e.g.*, in the example 2 the arguments $\mathcal{A}$ and $\mathcal{B}$ are in conflict, and have different weights or degrees of importance to the agent. We have redefined the resolution of conflicts so that when an argument $\mathcal{X}$ is not strong enough to defeat an other argument $\mathcal{Y}$, the argument $\mathcal{X}$ could still have a weakening effect over the argument $\mathcal{Y}$. Finally, for the last requisite, for those application domains in which decisions are critical or high-risk, we should accept the arguments that remain undefeated with a level that is above the threshold; that is, no argument will be accepted with a level below the threshold. In the example 2, the agent that decides whether or not renting an apartment will take into account only those arguments that have a level above the established threshold.

We will now present a formalization of the representation of meta-level information through labels attached to arguments; also, we will introduce a collection of operators used to combine and propagate those labels according to the different arguments. This meta-information will complement the process of establishing acceptability of arguments. Next, we will present the definition of *Argumentative Label Algebra (ALA)*, which provides the elements required for the purpose described above.

**Definition 7** *An Argumentative Label Algebra (or simply* ALA*) is a 4-tupla* $\langle \Gamma, \circledS, \ominus, N_{\circledast} \rangle$ *where:*

- $\Gamma$ *is a set of labels called* domain of labels.
- $\circledS : \Gamma \times \Gamma \to \Gamma$ *is called the* support operator.
- $\ominus : \Gamma \times \Gamma \to \Gamma$ *is called the* conflict operator.
- $N_{\circledast}$ *is an* identity element *for the operator* $\ominus$.

The carrier set of this algebra is a set of labels to be associated with argumentative claims, the *support* operator will be used to obtain the meta-information associated with the conclusion of an inference, from the meta-information (label) associated with the premises, and the *conflict* operator defines the meta-information (label) associated with a conclusion after considering the possible conflicts with other claims.

An instantiation of ALA suitable for the example in Section 2 will be presented below. This ALA is able to represent weights associated with the arguments.

**Example 1** *Let* $\Phi = \langle \Gamma, \circledS, \ominus, N_{\circledast} \rangle$ *be an* ALA*, instantiated to represent and manipulate argument weights in the following way:*

- *The labels domain* $\Gamma$ *is the real interval* $[0,1]$*; the identity element* $N_{\circledast}$ *is* $0$.
- *Let* $\alpha, \beta \in \Gamma$ *be two labels, the operators of support and conflict over labels are specified as follows:*

  $\alpha \circledS \beta = min(\alpha, \beta)$*, i.e., the support operator reflects that an argument is as strong as its weakest support.*

  $\alpha \ominus \beta = max(\alpha - \beta, 0)$*, i.e., the conflict operator models that the weight of a conclusion is weakened by the weight of its attacker.*

Below we will apply ALA to DeLP; then, we will analyze the effect of the postulates mentioned at the beginning of this section.

# 5 ALA applied to DeLP: Example and Analysis

Here, we will expand DeLP with ALA capabilities; thus, through this extension DeLP incorporates the ability to represent a particular feature associated with rules used in arguments. The labels are then propagated to the level of arguments, and they will be used to introduce the additional features of an argument. The association of this information to DeLP rules is formalized in the definition of *ℓ-program*.

**Definition 8 (*ℓ-program*)** *Let* $\mathcal{P}$ *be a* ℓ-program*. We say that* $\mathcal{P}$ *is a set of clauses of the form* $(\gamma, \alpha)$*, called* ℓ-clause*s, where:* (1) $\gamma$ *is a DeLP clause,* (2) $\alpha$ *is a label associated with the clause* $\gamma$.

We will say that $(\gamma, \alpha)$ is a strict (defeasible) *ℓ-clause* iff $\gamma$ is a strict (defeasible) DeLP clause; then, given a *ℓ-program* $\mathcal{P}$ we will distinguish the subset $\Pi$ of strict *ℓ-clause*s, and the subset $\Delta$ of defeasible *ℓ-clause*s. We introduce now the adapted version of the concept of argument, called *ℓ-argument*, for this extension of DeLP.

**Definition 9 (*ℓ-argument*)** *Let* $L$ *be a literal, and* $\mathcal{P}$ *be a* ℓ-program*. We say that* $\langle \mathcal{A}, L, E_{\mathcal{A}} \rangle$ *is an* ℓ-argument *for a literal* $L$ *from* $\mathcal{P}$*, where* $\mathcal{A} \subseteq \Delta$ *and* $Clauses(S) = \{\gamma \mid (\gamma, \alpha) \in \mathcal{A}\}$*, then:*

(1) $Clauses(\Pi \cup \mathcal{A}) \mathrel{|\!\!\!\sim} L$*, i.e., there exists a defeasible derivation of* $L$ *from* $Clauses(\Pi \cup \mathcal{A})$*,*

(2) $Clauses(\Pi \cup \mathcal{A})$ *is non contradictory,*

(3) $Clauses(\mathcal{A})$ *is such that there is no* $\mathcal{A}_1 \subsetneq \mathcal{A}$ *such that* $\mathcal{A}_1$ *satisfies* (1) *and* (2) *above, and*

(4) *Let* $\{\langle \mathcal{A}_1, L_1, E_{\mathcal{A}_1} \rangle, \ldots, \langle \mathcal{A}_n, L_n, E_{\mathcal{A}_n} \rangle\}$ *be the set of all proper ℓ-subarguments of* $\langle \mathcal{A}, L, E_{\mathcal{A}} \rangle$*, and* $E_R$ *is the label associated to the top rule with head* $L$ *and body* $L_1, L_2, ..., L_n$ *then the label* $E_{\mathcal{A}}$ *for* $\langle \mathcal{A}, L, E_{\mathcal{A}} \rangle$ *is*

$$E_{\mathcal{A}} = E_{\mathcal{A}_1} \circledS \ ... \circledS E_{\mathcal{A}_n} \circledS E_R$$

*Where the notion of ℓ-subargument extends naturally the previous notion of subargument, i.e., we say that* $\langle \mathcal{B}, Q, E_{\mathcal{B}} \rangle$ *is a* ℓ-subargument *of* $\langle \mathcal{A}, L, E_{\mathcal{A}} \rangle$ *iff* $\mathcal{B} \subseteq \mathcal{A}$.

As in the original DeLP we provide an argumentation based mechanism to determine *warranted* conclusions; for this, we preserve the definitions of disagreement (Definition 3) and attack (Definition 4). Nevertheless, the labels associated with the arguments define the relative influence of the arguments involved in the attack; in contrast, DeLP uses by default the notion of specificity to compare arguments. Thus, we redefine the concept of *defeat* to consider the influence of the arguments in the resolution of conflicts that arise between them.

**Definition 10 (Defeat)** *Let $\mathcal{P}$ be a $\ell$-*program. *Let $\langle \mathcal{A}, L, E_{\mathcal{A}} \rangle$ and $\langle \mathcal{B}, Q, E_{\mathcal{B}} \rangle$ be two arguments in $\mathcal{P}$. We say that $\langle \mathcal{B}, Q, E_{\mathcal{B}} \rangle$* defeats *$\langle \mathcal{A}, L, E_{\mathcal{A}} \rangle$, iff there exist a $\ell$-*subargument *$\langle \mathcal{A}_1, L_1, E_{\mathcal{A}_1} \rangle$ of $\langle \mathcal{A}, L, E_{\mathcal{A}} \rangle$ such that $\langle \mathcal{B}, Q, E_{\mathcal{B}} \rangle$ counterargues $\langle \mathcal{A}, L, E_{\mathcal{A}} \rangle$ at literal $L_1$, and $E_{\mathcal{A}_1} \ominus E_{\mathcal{B}} = N_{\circledast}$.*
*When $E_{\mathcal{A}_1} \ominus E_{\mathcal{B}} \neq N_{\circledast}$ we say that $\langle \mathcal{B}, Q, E_{\mathcal{B}} \rangle$* weakens *$\langle \mathcal{A}_1, L_1, E_{\mathcal{A}_1} \rangle$ resulting in $\langle \mathcal{A}_1, L_1, E'_{\mathcal{A}_1} \rangle$ with $E'_{\mathcal{A}_1} = E_{\mathcal{A}_1} \ominus E_{\mathcal{B}}$.*

In this extended version of DeLP, a literal $L$ will be warranted if there exists a non-defeated *$\ell$-argument* $\langle \mathcal{A}, L, E_{\mathcal{A}} \rangle$. To establish whether $\langle \mathcal{A}, L, E_{\mathcal{A}} \rangle$ is non-defeated, the set of defeaters for $\mathcal{A}$ will be considered, applying recursively the same process to each of these arguments. Thus, a complete dialectical analysis is required to determine which *$\ell$-arguments* are ultimately accepted. Such analysis results in a tree structure called *dialectical tree*, in which *$\ell$-arguments* are nodes marked as undefeated (`U-nodes`), undefeated weakened ($\texttt{U}_\texttt{w}$`-nodes`), or defeated (`D-nodes`). Note that $\texttt{U}_\texttt{w}$`-nodes` are also undefeated.

Each label associated with a *$\ell$-argument* will be refined by considering the attacks according to Definition 10 (Defeat). Nodes in the tree are considered bottom-up, starting with the leaves, and evaluating an inner node only after all its attackers were considered. Also, the attacks against each node are applied in sequence from those attacking at a deeper level to those attacking at a shallower level.

The labels in a dialectical tree are updated in the following way:

– The dialectical tree is built as usual (see Definition 6) with *$\ell$-arguments*.
– The labels are updated following Definition 7, and performing a breadth-first traversal of the dialectical tree in inverse order, *i.e.*, starting with the leaves.
  - The label contained in the arguments in the leaves remain unaltered.
  - The label of an internal node is calculated as specified in Definition 7 only when all its children nodes have their labels updated.

Now, the marking process proceeds on the resulting tree sequentially from deeper levels towards the root:

– Leaves are marked as `U-nodes`, since they have no defeaters.
– For every inner node $\langle \mathcal{A}, L, E_{\mathcal{A}} \rangle$, we label it as follows:
  1. The node $\langle \mathcal{A}, L, E_{\mathcal{A}} \rangle$ will be marked as `U-node` iff all its children are marked as `D-nodes`.
  2. Otherwise, if $\langle \mathcal{A}, L, E_{\mathcal{A}} \rangle$ has at least one child marked as `U-node` or $\texttt{U}_\texttt{w}$`-node`, then
     2.1 it is marked as `D-node` iff there exists a child marked as `U-node` or $\texttt{U}_\texttt{w}$`-node` that defeats it, and
     2.2 it is marked as $\texttt{U}_\texttt{w}$`-node`, otherwise.

This marking process allows us to characterize the set of literals obtained from a given *$\ell$-program*, these literals are called *warranted literals*. A literal $L$ is warranted iff there exist an *$\ell$-argument* $\langle \mathcal{A}, L, E_{\mathcal{A}} \rangle$ for $L$ such that the root of the associated marked dialectical tree $\mathcal{T}^{\star}_{\langle \mathcal{A}, L, E_{\mathcal{A}} \rangle}$ is a `U-node` or $\texttt{U}_\texttt{w}$`-node`.

Below we will present the *$\ell$-program*, the *$\ell$-arguments* and the relations between them, modeling the scenario described in Section 2. Finally, we will determine the acceptability of the arguments through the dialectical process described above.

**Example 2** *In the following $\ell$-*program *$\mathcal{P}$ each claim has a weight value representing how important is the claim for the agent, which depends on the application domain; in this case, the values attached to rules represent the weight of the connection between the antecedent and consequent of the rule. We use the constant 'a' to represent the 'department A'.*

$$\mathcal{P} = \left\{ \begin{array}{ll} r_1 : rent(X) \prec goodLocation(X), quiet(X)[1] & goodLocation(a)[0.9] \\ r_2 : {\sim}rent(X) \prec moldProblem(X)[0.4] & moreStudents(a)[0.7] \\ r_3 : {\sim}rent(X) \prec small(X)[0.3] & quiet(a)[0.9] \\ r_4 : {\sim}quiet(X) \prec moreStudents(X)[0.4] & small(a)[0.8] \\ r_5 : quiet(X) \prec peacefulPeople(X)[0.9] & peacefulPeolple(a)[1] \\ moldProblem(a)[0.6] & \end{array} \right\}$$

*From this* $\ell$-program $\mathcal{P}$ *the followings* $\ell$-argument*s can be produced, where we also show the weight measures (labels) calculated by applying the support operation defined in* ALA*:*

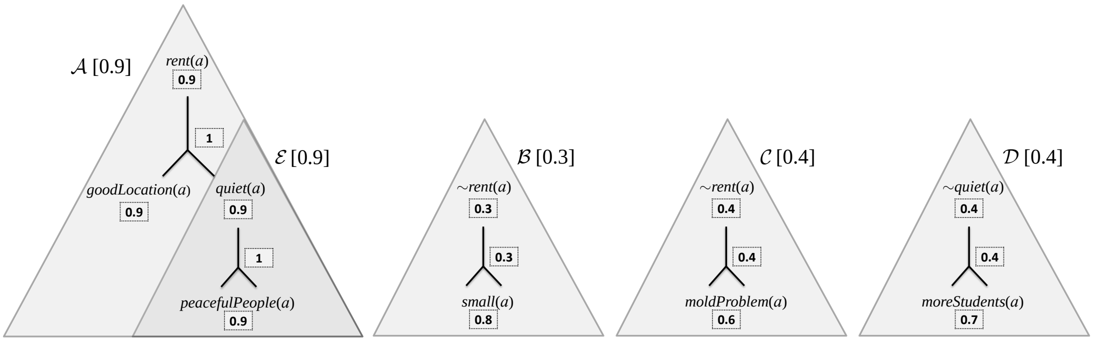


Figure 2: $\ell$-*argument*s

$$E_{\mathcal{B}} = E_{\mathcal{B}_1} \circledS E_{r_3} = min(0.8, 0.3) = 0.3$$
$$E_{\mathcal{C}} = E_{\mathcal{C}_1} \circledS E_{r_2} = min(0.6, 0.4) = 0.4$$
$$E_{\mathcal{D}} = E_{\mathcal{D}_1} \circledS E_{r_4} = min(0.7, 0.4) = 0.4$$
$$E_{\mathcal{E}} = E_{\mathcal{E}_1} \circledS E_{r_5} = min(0.9, 1) = 0.9$$
$$E_{\mathcal{A}} = E_{\mathcal{A}_1} \circledS E_{\mathcal{E}} \circledS E_{r_1} = min(min(0.9, 0.9), 1) = 0.9$$

*After obtaining the labels associated with the* $\ell$-arguments*, we proceed to build the dialectical tree (see Fig. 2) to determine the definitive status of acceptability and the weight of each argument.*

$$E'_{\mathcal{E}} = E_{\mathcal{E}} \ominus E_{\mathcal{D}} = max(0.9 - 0.4, 0) = 0.5$$

*Since the label associated with the* $\ell$-subargument $\mathcal{E}$ *of* $\mathcal{A}$ *was weakened, and the label of a* $\ell$-argument *depends on the labels of its* $\ell$-subarguments*, then we have to recalculate the label for* $\mathcal{A}$*.*

$$E'_{\mathcal{A}} = E_{\mathcal{A}_1} \circledS E'_{\mathcal{E}} \circledS E_{r_1} = min(min(0.9, 0.5), 1) = 0.5$$

*Then, we will resolve the attacks over* $\mathcal{A}$ *given the new weight.*

$$E''_{\mathcal{A}} = E'_{\mathcal{A}} \ominus E_{\mathcal{B}} = max(0.5 - 0.3, 0) = 0.2$$
$$E'''_{\mathcal{A}} = E''_{\mathcal{A}} \ominus E_{\mathcal{C}} = max(0.2 - 0.4, 0) = 0$$

*As in DeLP, a literal L is* warranted *if there exist a* $\ell$-argument $\langle \mathcal{A}, L, E_{\mathcal{A}} \rangle$ *for L such that it is marked as* `U-node` *(or as* $\mathtt{U_w}$`-node`*). Thus, for the example we can conclude that* $rent(a)$ *is not warranted, since* $\langle \mathcal{A}, rent(a), E_{\mathcal{A}} \rangle$*, the unique* $\ell$-argument *supporting* $\mathcal{A}$*, is marked as* `D-node` *(notice that the final label associated to* $\mathcal{A}$ *is 0). For this reason, we arrive at the conclusion that it is not convenient to rent the apartment A, because the conditions for renting the apartments are not strong enough after considering the conditions for not renting it. Note that, in* DeLP*, the argument* $\mathcal{A}$ *would be accepted, as it is stronger than all their attackers and therefore they cannot defeat it; this is so, because* DeLP *considers the attacks against a given argument* $\mathcal{A}$ *individually, and with a binary result: complete defeat for* $\mathcal{A}$ *or, simply undefeated, without any effect over* $\mathcal{A}$*. In other words,* DeLP *ignores the effect of conflicting arguments that do not constitute defeaters by themselves. This policy can be too simple in most decision-making problems (like the one of the example), where the graded (rather than binary) nature of reasons supporting decisions requires a more graded way of evaluating the argumentative interactions to reach the final decision. Under this proposal, the argument* $\mathcal{A}$ *was weakened and, in this case, was defeated by its counter-arguments. In the case in which the argument is undefeated, at least its strength will be weakened if there exists any counter-argument. So, the information coming from the counter-arguments is reflected in the outcome of the argumentative process.*

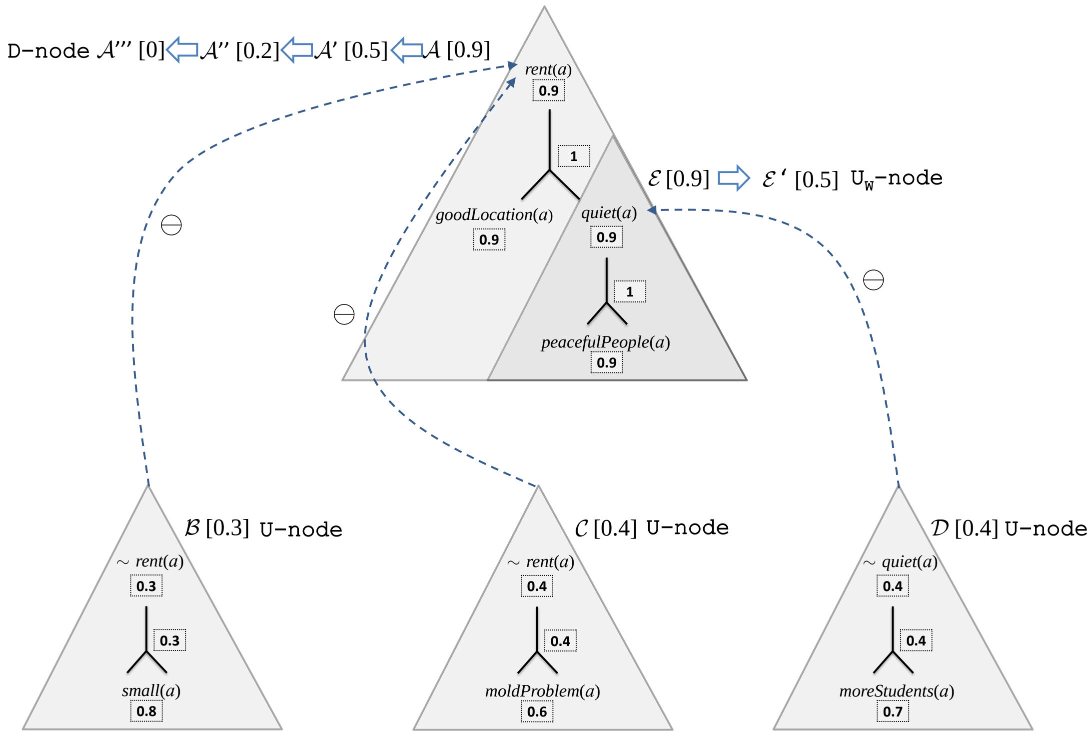


Figure 3: Dialectical tree and Acceptability status

As it was mentioned in the introduction, in certain real-world applications agents need to make decisions or follow recommendations to meet their goals; for example, it might be necessary to get recommendations matching at least certain degree of reliability or make decisions with a minimum degree of success. In other words, the recommendations or decisions must be supported with information meeting certain requirements that in turn are depending on the particular application domain. For that, we define an acceptability threshold $\tau \in \Gamma$ that determines the characteristics that the arguments must satisfy to form part of the justification for a recommendation or a decision. This threshold must be a totally ordered set. We say that a label $\alpha \in \Gamma$ *satisfies* a threshold $\tau$ iff $\alpha \ominus \tau \neq N_{\circledast}$.

– Leaves are marked as `U-nodes` iff the argument label satisfies the threshold $\tau$,

  otherwise it is marked as `D-node`.

– Then, for any inner node $\langle \mathcal{A}, L, E_{\mathcal{A}} \rangle$, we label it as follows:

  1) as `U-node` iff $E_{\mathcal{A}}$ satisfies $\tau$ and every child of $\langle \mathcal{A}, L, E_{\mathcal{A}} \rangle$ is marked as a `D-node`.
  2) as `D-node` if $E_{\mathcal{A}}$ does not satisfy $\tau$ or there exists a child marked as `U-nodes` or $\texttt{U}_\texttt{w}$`-nodes` that defeats it, and
  3) as $\texttt{U}_\texttt{w}$`-nodes`, otherwise.

**Example 3** *Here we analyze the same situation of the example 2, using the dialectical process previously described and defining an acceptability threshold. According to the acceptability threshold $\tau = 0.4$, we start marking the arguments $\mathcal{C}$ and $\mathcal{D}$ as undefeated (*`U-node`*), and marking the argument $\mathcal{B}$ as defeated (*`D-node`*) because their strength does not satisfies $\tau$.*

$E_{\mathcal{B}} = E_{\mathcal{B}_1} \circledS E_{r_3} = min(0.8, 0.3) = 0.3 < \tau$ `Does not Satisfy`
$E_{\mathcal{C}} = E_{\mathcal{C}_1} \circledS E_{r_2} = min(0.6, 0.4) = 0.4 = \tau$ `Satisfies`
$E_{\mathcal{D}} = E_{\mathcal{D}_1} \circledS E_{r_4} = min(0.7, 0.4) = 0.4 = \tau$ `Satisfies`

*Then, as in the previous case the argument $\mathcal{D}$ weakens $\mathcal{E}$, which causes a weakening in the argument $\mathcal{A}$. Both arguments $\mathcal{E}$ and $\mathcal{A}$ are marked at the moment as weak arguments (`U_w-nodes`).*

$$E'_{\mathcal{E}} = E_{\mathcal{E}} \ominus \ E_{\mathcal{D}} = max(0.9 - 0.4, 0) = 0.5 > \tau \ \ \texttt{Satisfies}$$
$$E'_{\mathcal{A}} = E_{\mathcal{A}_1} \circledS \ E'_{\mathcal{E}} \circledS \ E_{r_1} = min(min(0.9, 0.5), 1) = 0.5 > \tau \ \ \texttt{Satisfies}$$

*Finally, after considering the attack of the argument $\mathcal{C}$ on $\mathcal{A}$ the force of the argument is weakened to the point it goes below the threshold. Finally, $\mathcal{A}$ is marked as defeated (`D-node`).*

$$E''_{\mathcal{A}} = E'_{\mathcal{A}} \ominus \ E_{\mathcal{B}} = max(0.5 - 0.4, 0) = 0.1 < \tau \ \ \texttt{Not Satisfy}$$

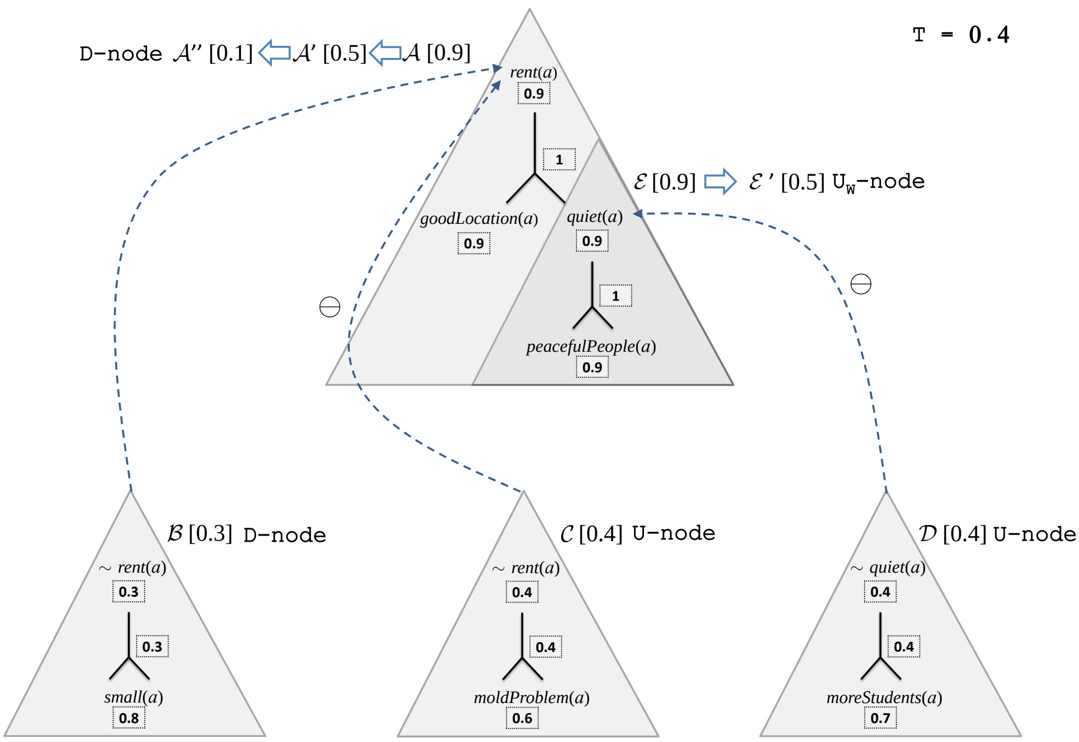


Figure 4: Argumentative tree, Acceptability status and Acceptability threshold

*This example is a variation of the previous example where we considered an acceptability threshold which determines whether the $\ell$-arguments are strong enough to be guaranteed or accepted. We obtain the conclusion that the $\ell$-argument $\langle \mathcal{A}, rent, E_{\mathcal{A}} \rangle$, where $E_{\mathcal{A}} = 0.1$, is not accepted even if its strength is not diminished completely because it does not satisfy the acceptability threshold $\tau = 0.4$.*

# 6 Related Work

Our main motivation can be found in Dov Gabbay's seminal work on Labeled Deductive Systems (LDS) [8, 9]. He introduced a rigorous yet flexible formalism to handle complex problems using logical frameworks extended with labeled deduction capabilities; this contribution has permitted to address research problems in areas such as temporal logics, database query languages, and defeasible reasoning systems. In labeled deduction, formulas are compound objets expressed as $L : \phi$, where $L$ represents a label associated with the logical formula $\phi$. These labels are used to carry additional information that give the representation language more capabilities. The intuitions labels represent may vary following the modeling needs of the system. The introduction of the idea of structuring labels as an algebra was present from the very inception of labeled systems [8].

Chesñevar et al. [6], applied the formalism of labeled deductive to argumentation systems; the proposal formally characterized different argument-based inference mechanisms putting them under the unifying structure of LDS. In this work, two inference operators are characterized and dialectical analysis as it is performed in argumentation systems is modeled: the building of arguments, and the construction of the tree-like dialectical structure for the analysis of warrant.

Although our proposal also involves the use of labels and an argumentative label algebra as in the works mentioned, our purpose is entirely different. We are not trying to unify and formally compare different logics, but to *extend* the representational capabilities of argumentation frameworks by allowing them to represent additional domain specific information. While it can be argued that, due to its extreme generality, Gabbay's framework could also be instantiated in some way to achieve this purpose, we have introduced a concrete way, where labels are propagated in the specific case of argument interactions, such as support and conflict.

T. J. M. Bench-Capon and J. L. Pollock have introduced systems that are very influential in argumentation community. In [3], Bench-Capon persuasively submits that in situations involving practical reasoning, it is impossible to demonstrate conclusively that either party is wrong; thus, in such cases the role of argument is to persuade rather than to prove, demonstrate or refute. In his own words: "The point is that in many contexts the soundness of an argument is not the only consideration: arguments also have a force which derives from the value they advance or protect." [3]. He also cites Perelman [13], and the work on jurisprudence as a source of telling examples where values become important. Pollock [14] puts forward the idea that most semantics for defeasible reasoning ignore the fact that some arguments are better than others, supporting their conclusions more strongly. But once we acknowledge that arguments can differ in strength and conclusions can differ in their degree of justification, things become more complicated. In particular, he introduces the notion of diminishes, which are defeaters that cannot completely defeat their target, but instead lower the degree of justification of that argument.

Another forerunner of our work can be found on P-DeLP [7], where the elements of the language are labeled with possibilistic values that are propagated to a final value for arguments constructed from these elements. In that work there was no attempt to further combine values between arguments, although our framework could handle that situation.

Based on the intuitions of these research lines, we formalize an argumentative label algebra applied to DeLP, in this system the labels are the way to represent the characteristics of the arguments, generalizing the notion of value. Clearly, the interaction between arguments can affect the labels they have associated, so that these changes can cause weakening (a form of diminishing) among arguments. Using this framework, we established argument acceptability, where the final labels propagated to the accepted arguments provide additional acceptability information, such as degree of justification, restrictions on justification, explanation, and others.

# 7 Conclusion and Future Work

Argumentation has contributed with a human-like mechanism to the formalization of commonsense reasoning. In the last decade, several argument-based formalisms have emerged, with application in many areas, such as legal reasoning, autonomous agents and multi-agent systems. For instance, in the implementation of an agent, it would be beneficial to establish the degree of success obtained by reaching a given objective.

Our work has focused on the development of the *Argumentative Label Algebra* (ALA), which gives us the possibility to increase the ability of representing real-world scenarios, modeling different attributes associated to the arguments such as uncertainty, reliability, time availability, degree of success, strength measure, or any other relevant feature. ALA allows to represent relations between arguments such as support and conflict. Each of these relationships has an operation associated in the algebra of labels defined within the formalism, which allows to propagate meta-information in the argumentation tree. Then, it is possible to determine the acceptability of arguments and the resulting meta-data associated with them. A peculiarity of the conflict relationship, is that through the operation defined for it in the algebra of labels it will allow the weakening of arguments. This weakening between arguments contributes to a better representation of the real world in some application domains.

We combined ALA and DeLP, introducing a rule-based argumentation framework considering different attributes represented by labels at the object language level. This information is used for: determine which argument defeats another, analyzing a feature that is relevant to the domain and define an acceptability threshold which will determine if the arguments are strong enough to be accepted which is a necessity in environments that require some weight measure in their answers.

As work in progress we are studying the formal properties of the algebra of labels operations that we have defined here. We also will study the effect of these notions on the acceptability relation. As future work we will develop an implementation of the application of ALA in the existing DeLP system [1] as a basis. The resulting implementation will be exercised in different domains requiring to model extra information associated to the arguments.

## References

[1] Leila Amgoud and Henri Prade. Using arguments for making and explaining decisions. *Artificial Intelligence*, 173(3-4):413–436, 2009.

[2] Christoph Beierle, Bernhard Freund, Gabriele Kern-Isberner, and Matthias Thimm. Using defeasible logic programming for argumentation-based decision support in private law. In *COMMA*, volume 216 of *Frontiers in Artificial Intelligence and Applications*, pages 87–98. IOS Press, 2010.

[3] Trevor J. M. Bench-Capon. Value-based argumentation frameworks. In Salem Benferhat and Enrico Giunchiglia, editors, *NMR*, pages 443–454, 2002.

[4] Philippe Besnard and Anthony Hunter. *Elements of Argumentation.* MIT Press, 2008.

[5] Maximiliano C. Budán, Mauro Javier Gómez Lucero, Carlos I. Chesñevar, and Guillermo R. Simari. Modelling time and reliability in structured argumentation frameworks. In Gerhard Brewka, Thomas Eiter, and Sheila A. McIlraith, editors, *KR*. AAAI Press, 2012.

[6] Carlos I. Chesñevar and Guillermo R. Simari. Modelling inference in argumentation through labelled deduction: Formalization and logical properties. *Logica Universalis*, 1(1):93–124, 2007.

[7] Carlos I. Chesñevar, Guillermo R. Simari, Teresa Alsinet, and Lluis Godo. A logic programming framework for possibilistic argumentation with vague knowledge. In David M. Chickering and Joseph Y. Halpern, editors, *UAI*, pages 76–84. AUAI Press, 2004.

[8] Dov Gabbay. Labelled deductive systems: a position paper. In J. Oikkonen and J. Vaananen, editors, *Proc. of Logic Colloquium '90*, volume 2 of *Lect. Notes in Logic*, pages 66–88. Springer-Verlag, 1993.

[9] Dov Gabbay. *Labelling Deductive Systems.* Oxford Univ. Press (Vol. 33, Oxford Logic Guides), 1996.

[10] Alejandro J. García and Guillermo R. Simari. Defeasible logic programming: An argumentative approach. *Theory Practice of Logic Programming*, 4(1):95–138, 2004.

[11] Lluis Godo, Enrico Marchioni, and Pere Pardo. Extending a temporal defeasible argumentation framework with possibilistic weights. In *JELIA*, pages 242–254, 2012.

[12] Philippe Pasquier, Ramon Hollands, Iyad Rahwan, Frank Dignum, and Liz Sonenberg. An empirical study of interest-based negotiation. *Autonomous Agents and Multi-Agent Systems*, 22(2):249–288, 2011.

[13] Chaim Perelman. *Justice, Law and Argument.* Synthese Library, Vol. 142. Reidel, Holland, 1980.

[14] John L. Pollock. Defeasible reasoning and degrees of justification. *Argument & Computation*, 1(1):7–22, 2010.

[1]See http://lidia.cs.uns.edu.ar/delp

[15] Iyad Rahwan, Sarvapali D. Ramchurn, Nicholas R. Jennings, Peter McBurney, Simon Parsons, and Liz Sonenberg. Argumentation-based negotiation. *Knowledge Eng. Review*, 18(4):343–375, 2003.

[16] Iyad Rahwan and Guillermo R. Simari. *Argumentation in Artificial Intelligence*. Springer Verlag, 2009.

[17] Iyad Rahwan, Liz Sonenberg, and Frank Dignum. Towards interest-based negotiation. In *AAMAS*, pages 773–780, 2003.

[18] F. Stolzenburg, Alejandro J. García, Carlos I. Chesñevar, and Guillermo R. Simari. Computing generalized specificity. *Journal of Applied Non-Classical Logics*, 13(1):87–113, 2003.

[19] Thomas L. van der Weide, Frank Dignum, John-Jules Ch. Meyer, Henry Prakken, and Gerard Vreeswijk. Multi-criteria argument selection in persuasion dialogues. In Peter McBurney, Simon Parsons, and Iyad Rahwan, editors, *ArgMAS*, volume 7543 of *LNCS*, pages 136–153. Springer, 2011.