

An approach for an algebra applied to a Defeasible Logic Programming

Maximiliano C. D. Budán, Mauro J. Gómez Lucero,
Guillermo R. Simari

Consejo Nacional de Investigaciones Científicas y Técnicas (CONICET)
Laboratorio de Investigación y Desarrollo en Inteligencia Artificial (LIDIA)
Universidad Nacional del Sur
Universidad Nacional de Santiago del Estero
`{mcd, mjg, grs}@cs.uns.edu.ar`

Abstract. 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.

In this work we present an approach applied to any concrete argumentation systems, called *Argumentative Labels Algebra (ALA)*, in which propagate certain information through the internal structure of the argument and the existing relations between these. This information can be used for different proposed: (1) determine which argument defeat another, analyzing a feature that is relevant to the domain (eg. time availability, degree of reliability, particular characteristics, among others) and (2) define a acceptability threshold which will determine if the arguments are strong enough to be accepted.

For this way, we obtain an approach that strengthens the argumentation systems, increase the ability of real-world representation modeling different attributes associated to the arguments.

Keywords: Concrete Argumentation Systems, Argumentative Labels Algebra, Defeasible Logic Programming

1 Introduction

Argumentation has contributed with a human-like mechanism to the formalization of commonsense reasoning. Argumentation is the process of defending a given affirmation by giving reasons for its acceptance. Both the original claim and its support are subject to consideration, since reasons supporting conflicting claims can be proposed. Several argument-based formalisms have emerged, with application in many areas such as legal reasoning, autonomous agents and multi-agent systems [3–5].

Besides abstract argumentation approaches, different more concrete argumentation systems exists, specifying a knowledge representation language, and how arguments are built. One of those systems is Defeasible Logic Programming (DeLP) [1], a formalism that combines results of Logic Programming and

Defeasible Argumentation. DeLP allows representing information in the form of weak rules in a declarative way, from which arguments supporting conclusions are constructed, and provides a defeasible argumentation inference mechanism for determining warranted conclusions. The defeasible argumentation basis of DeLP allows to build applications that deal with incomplete and contradictory information in dynamic domains.

In real application domains of argumentation may be require the explicit treatment of special characteristics (eg. time availability, degree of reliability, particular characteristics, among others), this information is not in general directly associated with arguments, but instead it is attached to the basic pieces of knowledge (in general logical rules) from which arguments are built.

In this paper we introduce a novel approach, called *Argumentation Labels Algebra (ALA)*, which can be applied to any concrete argumentation systems for modeling the capability of propagate information trough the arguments. This information can be used to determine which argument defeat another, analyzing a feature that is relevant to the domain (eg. time availability, degree of reliability, particular characteristics, among others)¹ and define an acceptability threshold which will determine if the arguments are strong enough to be accepted.

The central contribution of this paper is increase the ability of real-world representation modeling different attributes associated to the arguments, using an algebra of labels for propagate this information.

2 Defeasible Logic Programming (DeLP)

Different argumentation systems exists, specifying a knowledge representation language, and how arguments are built. One of those systems is *Defeasible Logic Programming (DeLP)* [1], a formalism that combines results of *Logic Programming* and *Defeasible Argumentation*. DeLP allows representing information in the form of weak rules in a declarative way, from which arguments supporting conclusions are constructed, and provides a defeasible argumentation inference mechanism for determining *warranted* conclusions. The defeasible argumentation basis of DeLP allows to build applications that deal with incomplete and contradictory information in dynamic domains.

Below we present the definitions of *program* and *argument* in DeLP.

Definition 1 (DeLP Program) *A DeLP program \mathcal{P} is a pair (Π, Δ) where (1) Δ is a set of defeasible rules of the form $L \prec P_1, \dots, P_n$, with $n > 0$, where L and each P_i are literals, and (2) Π is a set of strict rules of the form $L \leftarrow P_1, \dots, P_n$, 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 $L \leftarrow P_1, \dots, P_n$ with $n = 0$ is called *fact*

¹ The usefulness of some of these parameters were published in previous works [6, 7]

and will be denoted just as L , and a defeasible rule $L \prec P_1, \dots, P_n$ with $n = 0$ is called *presumption* and will be denoted just as $L \prec$. It is important to remark that the set Π must be consistent as it represents strict (undisputed) information. In contrast, the set Δ will generally be inconsistent, since it represents tentative information.

We say that a given set of DeLP clauses is contradictory if and only if exist 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 $\langle A, L \rangle$, where A is a set of defeasible rules of Δ , such that:*

1. *there is a defeasible derivation for L from $\Pi \cup A$.*
2. *$\Pi \cup A$ is not contradictory, and*
3. *A is a minimal, i.e., there exist no proper subset A' , $A' \subset A$ satisfying conditions (1) and (2).*

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

DeLP provides an argumentation based mechanism to determine *warranted* conclusions. This procedure involves constructing arguments from programs, identifying conflicts or *attacks* among arguments, evaluating pairs of arguments in conflict to determine if the attack is successful, becoming a *defeat*, and finally analyzing defeat interaction among all relevant arguments to determine warrant.

Below we briefly present the formalization of the previously mentioned notions, as introduced in [1].

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

Definition 4 (Attack) *Let $\mathcal{P} = (\Pi, \Delta)$ be a DeLP program. Let $\langle A_1, L_1 \rangle$ and $\langle A_2, L_2 \rangle$ be two arguments in \mathcal{P} . We say that $\langle A_1, L_1 \rangle$ counter-argues, rebuts, or attacks $\langle A_2, L_2 \rangle$ at the literal L if and only if there is a sub-argument $\langle A, L \rangle$ of $\langle A_2, L_2 \rangle$ such that L and L_1 are in disagreement. The argument $\langle A, L \rangle$ is called disagreement sub-argument, and the literal L will be the counter-argument point.*

In order to decide if a partial attack really succeeds, constituting a defeat, a comparison criterion must be used, establishing the relative strength of the arguments involved in the attack. In this work we will use the criterion adopted by default in DeLP, called *specificity*, 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 $\langle A_1, L_1 \rangle$ and $\langle A_2, L_2 \rangle$ be two arguments in \mathcal{P} . We say that $\langle A_2, L_2 \rangle$ defeats $\langle A_1, L_1 \rangle$ if and only if exist a sub-argument $\langle A, L \rangle$ of $\langle A_1, L_1 \rangle$ such that $\langle A_2, L_2 \rangle$ counter-argues $\langle A, L \rangle$ at literal L and it holds that:*

1. $\langle A_2, L_2 \rangle$ is strictly more specific than $\langle A, L \rangle$ (proper defeater), or
2. $\langle A_2, L_2 \rangle$ is unrelated to $\langle A, L \rangle$ (blocking defeater)

In DeLP a literal L will be warranted if there exists a non-defeated argument structure $\langle A, L \rangle$. In order to establish whether $\langle A, L \rangle$ is non-defeated, the set of defeaters for 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 [1] *The dialectical tree for an argument $\langle A, L \rangle$, denoted $\mathcal{T}_{\langle A, L \rangle}$, is recursively defined as follows: (1) A single node labeled with an argument $\langle A, L \rangle$ with no defeaters (proper or blocking) is by itself the dialectical tree for $\langle A, L \rangle$; (2) Let $\langle A_1, L_1 \rangle, \langle A_2, L_2 \rangle, \dots, \langle A_n, L_n \rangle$ be all the defeaters (proper or blocking) for $\langle A, L \rangle$. The dialectical tree for $\langle A, L \rangle$, $\mathcal{T}_{\langle A, L \rangle}$, is obtained by labeling the root node with $\langle A, L \rangle$, and making this node the parent of the root nodes for the dialectical trees of $\langle A_1, L_1 \rangle, \langle A_2, L_2 \rangle, \dots, \langle A_n, L_n \rangle$.*

For the marking procedure we start labeling the leaves as **U-nodes**. Then, for any inner node $\langle A_2, Q_2 \rangle$, it will be marked as **U-node** if and only if every child of $\langle A_2, Q_2 \rangle$ is marked as a **D-node**. If $\langle A_2, Q_2 \rangle$ has at least one child marked as **U-node** then it is marked as a **D-node**.

This marking allows us to characterize the set of literals sanctioned by a given *de.lp*, called *warranted literals*. A literal L is *warranted* if and only if exist an argument structure $\langle A, L \rangle$ for L , such that the root of its marked dialectical tree $\mathcal{T}_{\langle A, L \rangle}^*$ is a **U-node**.

3 An Argumentative Labels Algebra

In any argumentation systems, can be of interest propagate certain information through the existing relations between the arguments.

In other words, can be useful the analysis of a characteristic associated with a certain argument dependent on the environment or a characteristic to reflect changes in the environment. This information can be applied for different proposed such as: (1) determine which argument defeat another, analyzing a feature that is relevant to the domain (eg. time availability, degree of reliability, particular characteristics, among others) and (2) define a acceptability threshold which will determine if the arguments are strong enough to be accepted.

For the first postulate, we will use the labels associated to the arguments for comparison among them. For example, assume a scenario in which we must decide whether to invest in the company J or K , and we have the arguments proposed by the consultants A and B :

- (Arg_A) Invest in the company J because is stable and safe.
- (Arg_B) Invest in the company K because the shares are rising.

Our approach offers the possibility to handle extra information associated with the arguments, i.e., its possible associate a degree of reliability at the arguments (Arg_A) and (Arg_B), that represent the reliability degree of the consultants A and B . The argument (Arg_A) has a degree of reliability $[0.8]$ and (Arg_B) has a degree of reliability $[0.5]$, then the (Arg_A) defeat (Arg_B) such that $[0.8] > [0.5]$ (or the argument Arg_A is more reliable than Arg_B).

For the second postulate, suppose an environment in which decisions must be critical or high-risk, in this way we must accept the arguments that remain under a reliability level above the threshold. That is, no argument can be accepted with a reliability level below the threshold. For example, if we have a recommendation system to decide on which company would be wise to invest (buy shares), is of interest determine the result using only those arguments that have a level of belief above the threshold. That is, only considered the information given by sources with some degree of reliability.

In this section we present a different approach to argumentation, that uses an algebra of labels for the propagation of meta-data through an argumentation system. Next we formalize the definition of *Argumentative Label Algebra (ALA)*, which provides the elements required for this proposed.

Definition 7 *An Argumentative Labels Algebra (or simply ALA) is a 4-tupla $\langle Arg, \ominus, \oplus, N \rangle$ where:*

- Let **Arg** be a set of arguments. An argument is a 3-tupla $\langle A, L, E \rangle$, where A the set of knows pieces that support the conclusion, L is the conclusion of the argument, and E is a labels that represent a particular characteristic.
- Let \ominus be an operator of weakness affecting the labels associated with the arguments. Let $\langle A, L, E_A \rangle$ and $\langle B, Q, E_B \rangle$ be two arguments. Let $\langle B, Q, E_B \rangle$ disagreement $\langle A_i, L_i, E_i \rangle$ where $\langle A_i, L_i, E_i \rangle$ is a sub-argument of $\langle A, L, E_A \rangle$. We defined the operator \ominus as: $\langle B, Q, E_B \rangle \ominus \langle A_i, L_i, E_i \rangle = \langle A_i, L_i, E_i - E_B \rangle$.
- Let \oplus be an operator of fortress affecting the labels associated with the internal structure (rules or knows pieces) of each arguments. Let $\langle A, L, E \rangle$ be an argument.
 - If $A = \emptyset$ then $E_A = \ell_L$ where ℓ_L is the label associate with the literal L . (in some argumentative system represent a fact o presumption).
 - If $A \neq \emptyset$, L is obtained through a derivation R and the sub-arguments $\langle A_1, L_1, E_1 \rangle, \langle A_2, L_2, E_2 \rangle, \dots, \langle A_n, L_n, E_n \rangle$ then $E_A = \langle A_1, L_1, E_1 \rangle \oplus \langle A_2, L_2, E_2 \rangle \oplus \dots \oplus \langle A_n, L_n, E_n \rangle \oplus \ell_R = E_A = E_1 + E_2 + \dots + E_n + \ell_R$, where ℓ_R is the label associate with the derivation R .
- Let **N** be the neutral element for the operators \ominus and \oplus .

Below we will apply our algebra *ALA* to DeLP. Then, we analyze the effect of the postulates mentioned at the beginning of the section.

4 A Argumentative Labels Algebra applied in DeLP: Examples and Analysis

In the previous sections, we present a formalism that provides the ability to manipulate extra information through models of argument and the useful points of view for handling this information associated with the arguments.

In this section, we apply this theory in DeLP. The combination of the *ALA* and DeLP incorporates the ability to represent a particular characteristic associated with rules composing arguments. This information is then propagated to the level of arguments, and will be used to define the represented characteristics of an argument. The association of this information to DeLP clauses is formalized through 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 (γ, ℓ) , called ℓ -clause, where: (1) γ is a DeLP clause, (2) ℓ is a labels that represent the information associated with the clause γ .*

We will say that (γ, ℓ) is a strict (defeasible) *ℓ -clause* iff γ is a strict (defeasible) DeLP clause. Then, given a *ℓ -program* \mathcal{P} we will distinguish the subset Π of strict *ℓ -clauses*, and the subset Δ of defeasible *ℓ -clauses*.

In the previous section, we will present a notion of arguments as a 3-tupla $\langle A, L, E \rangle$, where A the set of knows pieces that support the conclusion, L is the conclusion of the argument, and E is a labels that represent a particular characteristics. Now, using the formalism DeLP, we can specified the set of knows pieces by a set of *ℓ -clauses*.

Given a set S of *ℓ -clauses*, we will use $Clauses(S)$ to denote the set of all DeLP clauses involved in *ℓ -clauses* of S . $Clauses(S) = \{\gamma \mid (\gamma, \ell) \in S\}$

Definition 9 (*ℓ -argument*) *Let L be a literal, and \mathcal{P} be a ℓ -program. We say that $\langle A, L, E_A \rangle$ is an ℓ -argument for a goal L from \mathcal{P} , if $A \subseteq \Delta$, where:*

- (1) $Clauses(\Pi \cup A) \vdash L$
- (2) $Clauses(\Pi \cup A)$ is non contradictory.
- (3) $Clauses(A)$ is such that there is no $A_1 \subsetneq A$ such that A_1 satisfies conditions (1) and (2) above.
- (4) $E_A = \langle A_1, L_1, E_1 \rangle \oplus \langle A_2, L_2, E_2 \rangle \oplus \dots \oplus \langle A_n, L_n, E_n \rangle \oplus \ell_R$, where $\langle A_i, L_i, E_i \rangle$ is a sub-argument of $\langle A, L, E \rangle$ with $1 \leq i \leq n$, and ℓ_R is the rule with head L and body L_1, L_2, \dots, L_n .

We say that $\langle B, Q, E_B \rangle$ is a sub-argument of $\langle A, L, E_A \rangle$ iff $B \subseteq A$.

The classical DeLP provides an argumentation based mechanism to determine *warranted* conclusions. Now we defined for this new DeLP version the argumentation mechanism to determine *warranted* conclusions. On the one hand, we preserve the definitions of disagreement(Definition 3) and attack(Definition 4). On the other hand, we defined the news concepts of weaken and defeat.

ALA in DeLP can be applying in order to decide if a partial attack really succeeds a defeat. The labels associated with the arguments defined the relative

strength of the arguments involved in the attack. Now we present a new concept for DeLP, called *weaken*, this new concept add to DeLP the treatment of weakening between arguments using the operator \ominus of *ALA*.

Definition 10 (Weaken) *Let \mathcal{P} be a ℓ – program. Let $\langle A, L, E_A \rangle$ and $\langle B, Q, E_B \rangle$ be two arguments in \mathcal{P} . We say that $\langle B, Q, E_B \rangle$ weaken $\langle A, L, E_A \rangle$, if and only if exist a sub-argument $\langle A_1, L_1, E_{A_1} \rangle$ of $\langle A, L, E_A \rangle$ such that $\langle B, Q, E_B \rangle$ counter-argues $\langle A, L, E_A \rangle$ at literal L_1 and $E_{A_1} > E_B$. The force of the argument $\langle A_1, L_1, E_{A_1} \rangle$ decreases according to the force of the argument $\langle B, Q, E_B \rangle$, which implied an indirect decrease force of the argument $\langle A, L, E_A \rangle$, formally:*

If $\langle B, Q, E_B \rangle \ominus \langle A_1, L_1, E_{A_1} \rangle$ and $E_{A_1} > E_B$ then $\langle A_1, L_1, E_{A_1} - E_B \rangle$ and an indirect decrease force $\langle A, L, E_A - E_B \rangle$.

In this work, we will use two criterion to determine which argument is defeated: (1) determine which argument defeat another, analyzing a feature that is relevant to the domain and (2) define a acceptability threshold which will determine if the arguments are strong enough to be accepted. The formal definition of this postulates are presented below:

Definition 11 (Defeat and Argument Force) *Let \mathcal{P} be a ℓ – program. Let N be the neutral element for the operator \ominus . Let $\langle A, L, E_A \rangle$ and $\langle B, Q, E_B \rangle$ be two arguments in \mathcal{P} . We say that $\langle B, Q, E_B \rangle$ defeat $\langle A, L, E_A \rangle$, if and only if exist a sub-argument $\langle A_1, L_1, E_{A_1} \rangle$ of $\langle A, L, E_A \rangle$ such that $\langle B, Q, E_B \rangle$ counter-argues $\langle A, L, E_A \rangle$ at literal L_1 and $E_{A_1} \leq E_B$. The argument $\langle A_1, L_1, E_{A_1} \rangle$ is weakened and defeated for the argument $\langle B, Q, E_B \rangle$, because the force of the argument $\langle A_1, L_1, E_{A_1} \rangle$ is weakened to a force equal to N , formally:*

If $\langle B, Q, E_B \rangle \ominus \langle A_1, L_1, E_{A_1} \rangle$ and $E_{A_1} \leq E_B$ then $\langle A_1, L_1, E_{A_1} = N \rangle$ and an indirect defeated $\langle A, L, E_A = N \rangle$.

An intuitive notion for the definition 11 would be that an argument is defeated if one of its supports (or piece of knows that composed the argument) are weakened to a force equal to the neutral element N . In other words, if one of the supports of the argument is defeated then the argument is defeated. Now if combine this concept with the notion of threshold, we obtain the following definition of defeat.

Definition 12 (Defeat and Threshold) *Let \mathcal{P} be a ℓ – program. Let N be the neutral element for the operator \ominus . Let $\langle A, L, E_A \rangle$ and $\langle B, Q, E_B \rangle$ be two arguments in \mathcal{P} . We say that $\langle B, Q, E_B \rangle$ defeat $\langle A, L, E_A \rangle$, if and only if exist a sub-argument $\langle A_1, L_1, E_{A_1} \rangle$ of $\langle A, L, E_A \rangle$ such that $\langle B, Q, E_B \rangle$ counter-argues $\langle A, L, E_A \rangle$ at literal L_1 and $E_A, E_B \geq T$. We say that $\langle A, L, E_A \rangle$ is defeated if and only if : $E_{A_1} - E_B = N$ or $E_A - E_B < T$*

The difference between definitions 11 and 12 is that, in the definition 12 we consider only those arguments which have a force greater than the threshold T . That is, all arguments that have a force below T are taken as defeated.

Now we calculate the label associated with the argument B through the operator \oplus , for the same procedure can be obtained the labels for the others arguments.

$$E_B = \langle B_1, t, E_{B_1} \rangle \oplus \langle B_2, r, E_{B_2} \rangle \oplus \ell_{s \prec t, r} = E_{B_1} + E_{B_2} + \ell_{s \prec t, r} = 0.5 + 1.7 + 0.5 = 2.7$$

$$E_{B_1} = \ell_t = 0.5 \text{ because the literal } t \text{ is a fact (see in definition 7).}$$

$$E_{B_2} = \langle B_{21}, z, E_{B_{21}} \rangle \oplus \langle B_{22}, p, E_{B_{22}} \rangle \oplus \ell_{r \prec z, p} = E_{B_{21}} + E_{B_{22}} + \ell_{r \prec z, p} = 1.2 + 0.1 + 0.4 = 1.7$$

$$E_A = \langle A_1, s, E_{A_1} \rangle \oplus \langle A_2, c, E_{A_2} \rangle \oplus \ell_{a \prec s, c} = 5.0$$

$$E_C = \langle C_1, b, E_{C_1} \rangle \oplus \langle C_2, n, E_{C_2} \rangle \oplus \ell_{k \prec b, n} = 3.1$$

$$E_D = \langle D_1, w, E_{D_1} \rangle \oplus \langle D_2, g, E_{D_2} \rangle \oplus \ell_{b \prec w, g} = 1.3$$

Once obtained the arguments and the respective labels associated with them, we analyze the relationship between the arguments. In this example, there exist contradictions between the arguments so it continues applying the concept of *weaknes by definition 10* and *defeat by definition 11*, then we determine which arguments are warranted.

$$\langle D, b, E_D \rangle \ominus \langle C_1, b, E_{C_1} \rangle =_{E_{C_1} > E_D} \langle C_1, b, E_{C_1} - E_D \rangle = \langle C_1, b, 1.7 - 1.3 \rangle = \langle C_1, b, 0.4 \rangle$$

$$\text{and } \langle C, k, 3.1 - 1.3 \rangle = \langle C, k, 1.8 \rangle \text{ (Weaken)}$$

$$\langle B, s, E_B \rangle \ominus \langle A_1, s, E_{A_1} \rangle =_{E_{A_1} \leq E_B} \langle A_1, s, E_{A_1} - E_B \rangle = \langle A_1, s, 1.4 - 2.7 \rangle = \langle A_1, s, 0 \rangle$$

$$\text{and } \langle A, a, 0 \rangle \text{ (Defeated)}$$

Now we can defined a threshold $T = 3.0$ necessary for the definition of defeat 12.

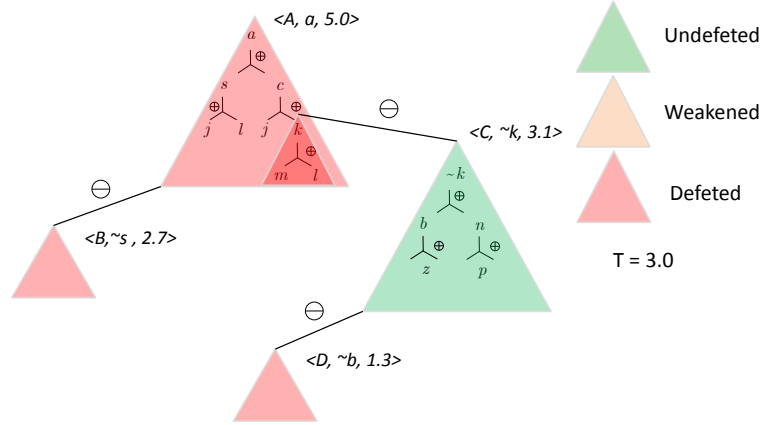


Fig. 2. Marking procedure, Arguments force and Thresholds.

As in the conventional DeLP, a literal L is *warranted* if and only if exist an argument structure $\langle A, L, E_A \rangle$ for L , such that its marked label as U-node. Then, we can reach the conclusion that the argument $\langle A, L, E_A \rangle$ is not accepted.

5 Conclusion. Related 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.

In this work, increase the ability of real-world representation modeling different attributes associated to the arguments, using an *algebra of labels (ALA)* for propagate this information. We combined *ALA* and DeLP, introducing a rule-based argumentation framework considering different attributes represented by labels at the object language level. This information was used to two proposed: determine which argument defeat another, analyzing a feature that is relevant to the domain and define a acceptability threshold which will determine if the arguments are strong enough to be accepted which is a necessary in environments that require some degree of strength in their answers.

As future work we will develop an implementation of the application of *ALA* in the existing DeLP system ² as a basis.

The resulting implementation will be exercised in different domains requiring to model extra information associated to the arguments.

References

1. A. J. García and G. R. Simari, Defeasible Logic Programming: An Argumentative Approach (2004), Theory Practice of Logic Programming, Vol. 4, Num. 1, Pag. 95-138.
2. N. D. Rotstein, M. O. Moguillansky, A. J. García and G. R. Simari (2010), A Dynamic Argumentation Framework, COMMA 2010, Pag. 427-438.
3. P. Pasquier, R. Hollands, I. Rahwan, F. Dignum and L. Sonenberg. An empirical study of interest-based negotiation. AAMAS 2011. Vol:22,Num.2, Pag. 249-288.
4. I. Rahwan, S. D. Ramchurn, N. R. Jennings, P. Mcburney, S. Parsons, L. Sonenberg . Argumentation-based negotiation. KR 2003. Vol:18,Issue 4, Pag. 343-375.
5. T. L. van der Weide, F. Dignum, J. J. Ch. Meyer, H. Prakken, G. W. A. Vreeswijk. Multi-criteria argument selection in persuasion dialogues. AAMAS 2011. Taipei, Taiwan. 921-928.
6. M. C. D. Budn, M. Gmez Lucero, C. I. Chesevar, G. R. Simari. An Approach to Argumentation Considering Attacks Through Time. SUM 2012 - (Accepted), Marburg, Germany.
7. M. C. D. Budn, M. Gmez Lucero, C. I. Chesevar, G. R. Simari. Modeling Time and Reliability in Structured Argumentation Frameworks. KR 2012, Rome, Italy, June 2012, pp 578-582.

² See <http://lidia.cs.uns.edu.ar/delp>