EXPANSION OPERATORS FOR MODELLING AGENT REASONING IN POSSIBILISTIC DEFEASIBLE LOGIC PROGRAMMING[1]

Carlos I. Chesñevar[a] Guillermo R. Simari[c] Lluís Godo[b] Teresa Alsinet[a]

[a] *University of Lleida, Department of Computer Science – 25001 Lleida,* SPAIN
Email: {`tracy,cic`}`@diei.udl.es`
[b] *AI Research Institute – IIIA/CSIC – Campus UAB – 08193 Bellaterra,* SPAIN
Email: `godo@iiia.csic.es`
[c] *Dept. of Computer Science & Eng. – Univ. Nacional del Sur– 8000 B. Blanca,* ARGENTINA
Email: `grs@cs.uns.edu.ar`

Abstract

Possibilistic Defeasible Logic Programming (P-DeLP) is a logic programming language which combines features from argumentation theory and logic programming, incorporating as well the treatment of possibilistic uncertainty and fuzzy knowledge at object-language level. Defeasible argumentation in general and P-DeLP in particular provide a way of modelling non-monotonic inference. When modelling intelligent agents, capturing defeasible inference relationships for modelling argument and warrant is particularly important, as well as the study of their logical properties. This paper analyzes two specialized non-monotonic operators for P-DeLP which model the *expansion* of a given program $\mathcal{P}$ by adding new weighed facts associated with argument conclusions and warranted literals. Different logical properties are studied and analyzed, providing useful comparison criteria that can be extended and applied to other argumentation frameworks.

1 Introduction and motivations

Possibilistic Defeasible Logic Programming (P-DeLP) [13, 11] is a logic programming language which combines features from argumentation theory and logic programming, incorporating as well the treatment of possibilistic uncertainty and fuzzy knowledge at object-language level. These knowledge representation features are formalized on the basis of PGL [1, 2], a possibilistic logic based on Gödel fuzzy logic. In PGL formulas are built over fuzzy propositional variables and the certainty degree of formulas is expressed with a necessity measure. In a logic programming setting, the proof method for PGL is based on a complete calculus for determining the maximum degree of possibilistic entailment of a fuzzy goal. The top-down proof procedure of P-DeLP has already been integrated in a number of real-world applications such as intelligent web search [9] and natural language processing [7], among others.

In a MAS context, we propose a model in which intelligent agents will encode their knowledge about the world using a P-DeLP program [10], using the argument and warrant computing procedure to perform their inferences. Clearly, P-DeLP-based agents will be usually performing their activities in a dynamic environment, so that it should also be able to reason, plan, and act according to new perceptions from the outside world. Such perceptions will be sensed by the agents, integrating them into their current beliefs.

Recent research [18, 21] has shown that argument-based approaches to formalize knowledge and reasoning in intelligent agents have proven to be very successful. We contend that in such settings an agent's reasoning capabilities can be better modelled and understood in terms of suitable *inference operators*. The advantages of such inference operators is twofold: on the one hand, from a theoretical viewpoint logical properties of defeasible argumentation can be easier studied with such operators at hand. On the other hand, actual implementations of argumentation systems could benefit from such logical properties for more efficient computation in the context of real-world applications.

[1] A slightly different version of this paper (not considering how to model agent reasoning capabilities) was originally published in [12]

This paper analyzes two non-monotonic *expansion operators* for P-DeLP, intended for modelling the effect of expanding a given program (which stands for an agent's knowledge base) by introducing new facts, associated with argument conclusions and warranted literals, respectively. The associated logical properties are studied and contrasted with a traditional SLD-based Horn logic. We contend that this analysis provides useful comparison criteria that can be extended and applied to other argumentation frameworks. As we will show in this paper, expansion operators provide an interesting counterpart to traditional consequence operators in logic programming [16]. For the sake of simplicity we will restrict our analysis to the fragment of P-DeLP built over classical propositions, hence based on *classical* possibilistic logic [14] and not on PGL itself (which involves fuzzy propositions). The rest of the paper is structured as follows: Section 2 summarizes the fundamentals of the P-DeLP framework. Section 3 discusses how P-DeLP can be used for modelling reasoning in intelligent agents, and the role of expansion operators for understanding the relationships between argument-based inferences the agent could perform. Section 2 summarizes the P-DeLP framework. In Section 4 we characterize two expansion operators for capturing the effect of expanding a P-DeLP program by adding argument conclusions and warranted literals, as well as their emerging logical properties. In Section 5 we discuss related work, and finally in Section 6 we summarize the most important conclusions that have been obtained.

2 The P-DeLP programming language: fundamentals

The classical fragment of P-DeLP language $\mathcal{L}$ is defined from a set of ground atoms (propositional variables) $\{p, q, \ldots\}$ together with the connectives $\{\sim, \wedge, \leftarrow\}$. The symbol $\sim$ stands for *negation*. A *literal* $L \in \mathcal{L}$ is a ground (fuzzy) atom q or a negated ground (fuzzy) atom $\sim q$, where q is a ground (fuzzy) propositional variable. A *rule* in $\mathcal{L}$ is a formula of the form $Q \leftarrow L_1 \wedge \ldots \wedge L_n$, where $Q, L_1, \ldots, L_n$ are literals in $\mathcal{L}$. When $n = 0$, the formula $Q \leftarrow$ is called a *fact* and simply written as Q. The term *goal* will be used to refer to any literal $Q \in \mathcal{L}$.[2] In the following, capital and lower case letters will denote literals and atoms in $\mathcal{L}$, resp.

Definition 1 (P-DeLP formulas) *The set Wffs*$(\mathcal{L})$ *of wffs in* $\mathcal{L}$ *are* facts, rules *and* goals *built over the literals of* $\mathcal{L}$. *A* certainty-weighted *clause in* $\mathcal{L}$*, or simply* weighted clause, *is a pair of the form* (φ, α)*, where* $\varphi \in$ *Wffs*$(\mathcal{L})$ *and* $\alpha \in [0, 1]$ *expresses a lower bound for the certainty of* φ *in terms of a necessity measure.*

The original P-DeLP language [13] is based on Possibilistic Gödel Logic or PGL [1], which is able to model both uncertainty and fuzziness and allows for a partial matching mechanism between fuzzy propositional variables. In this paper for simplicity and space reasons we will restrict ourselves to fragment of P-DeLP built on non-fuzzy propositions, and hence based on the necessity-valued classical propositional Possibilistic logic [14]. As a consequence, possibilistic models are defined by possibility distributions on the set of classical interpretations [3] and the proof method for our P-DeLP formulas, written $\vdash$, is defined by derivation based on the following generalized modus ponens rule (GMP):

$$\frac{\begin{array}{c}(L_0 \leftarrow L_1 \wedge \cdots \wedge L_k, \gamma) \\ (L_1, \beta_1), \ldots, (L_k, \beta_k)\end{array}}{(L_0, \min(\gamma, \beta_1, \ldots, \beta_k))}$$

which is a particular instance of the well-known possibilistic resolution rule, and which provides the *non-fuzzy* fragment of P-DeLP with a complete calculus for determining the maximum degree of possibilistic entailment for weighted literals.

In P-DeLP we distinguish between *certain* and *uncertain* clauses. A clause (φ, α) will be referred as certain if $\alpha = 1$ and uncertain, otherwise. Moreover, a set of clauses Γ will be regarded as *contradictory*, denoted $\Gamma \vdash \perp$, if $\Gamma \vdash (q, \alpha)$ and $\Gamma \vdash (\sim q, \beta)$, with $\alpha > 0$ and $\beta > 0$, for some atom q in $\mathcal{L}$[4]. A P-DeLP program is a set of weighted rules and facts in $\mathcal{L}$ in which we distinguish certain from uncertain information. As additional requirement, certain knowledge is required to be non-contradictory. Formally:

[2]Note that a conjunction of literals is not a valid goal.

[3]Although the connective $\leftarrow$ in logic programming is different from the material implication, e.g. $p \leftarrow q$ is not the same as $\sim q \leftarrow \sim p$, regarding the possibilistic semantics we assume here they share the same set interpretations.

[4]Notice that this notion of contradiction corresponds to the case when the inconsistency degree of Γ is strictly positive as defined in possibilistic logic.

Definition 2 (Program) *A P-DeLP program $\mathcal{P}$ (or just program $\mathcal{P}$) is a pair (Π, Δ), where Π is a non-contradictory finite set of certain clauses, and Δ is a finite set of uncertain clauses. If $\mathcal{P} = (\Pi, \Delta)$ is a program, we will also write $\mathcal{P}^{\Pi}$ (resp. $\mathcal{P}^{\Delta}$) to identify the set of certain (resp. uncertain) clauses in $\mathcal{P}$.*

The following notion of argument is based on the one presented in [22] (and similar to [3, 4]), and considers the necessity degree with which the argument supports a conclusion. The procedural mechanism for computing arguments can be found in [11].

Definition 3 (Argument. Subargument) *Given a program $\mathcal{P} = (\Pi, \Delta)$, a set $\mathcal{A} \subseteq \Delta$ of uncertain clauses is an* argument *for a goal Q with necessity degree $\alpha > 0$, denoted $\langle \mathcal{A}, Q, \alpha \rangle$, iff: (1) $\Pi \cup \mathcal{A} \vdash (Q, \alpha)$; (2) $\Pi \cup \mathcal{A}$ is non contradictory; and (3) There is no $\mathcal{A}_1 \subset \mathcal{A}$ such that $\Pi \cup \mathcal{A}_1 \vdash (Q, \beta)$, $\beta > 0$. Let $\langle \mathcal{A}, Q, \alpha \rangle$ and $\langle \mathcal{S}, R, \beta \rangle$ be two arguments. We will say that $\langle \mathcal{S}, R, \beta \rangle$ is a* subargument *of $\langle \mathcal{A}, Q, \alpha \rangle$ iff $\mathcal{S} \subseteq \mathcal{A}$. Notice that the goal R may be a subgoal associated with the goal Q in the argument $\mathcal{A}$.*[5]

As in most argumentation formalisms (see e.g. [20, 8]), in P-DeLP it can be the case that there exist *conflicting* arguments. Defeat among conflicting arguments involves a *preference criterion* defined on the basis of necessity measures associated with arguments.

Definition 4 (Counterargument) *Let $\mathcal{P}$ be a program, and let $\langle \mathcal{A}_1, Q_1, \alpha_1 \rangle$ and $\langle \mathcal{A}_2, Q_2, \alpha_2 \rangle$ be two arguments wrt $\mathcal{P}$. We will say that $\langle \mathcal{A}_1, Q_1, \alpha_1 \rangle$ counterargues $\langle \mathcal{A}_2, Q_2, \alpha_2 \rangle$ iff there exists a subargument (called* disagreement subargument*) $\langle \mathcal{S}, Q, \beta \rangle$ of $\langle \mathcal{A}_2, Q_2, \alpha_2 \rangle$ such that $\Pi \cup \{(Q_1, \alpha_1), (Q, \beta)\}$ is contradictory.*

Definition 5 (Preference criterion $\succeq$) *Let $\mathcal{P}$ be a P-DeLP program, and let $\langle \mathcal{A}_1, Q_1, \alpha_1 \rangle$ be a counterargument for $\langle \mathcal{A}_2, Q_2, \alpha_2 \rangle$. We will say that $\langle \mathcal{A}_1, Q_1, \alpha_1 \rangle$* is preferred *over $\langle \mathcal{A}_2, Q_2, \alpha_2 \rangle$ (denoted $\langle \mathcal{A}_1, Q_1, \alpha_1 \rangle \succeq \langle \mathcal{A}_2, Q_2, \alpha_2 \rangle$) iff $\alpha_1 \geq \alpha_2$. If it is the case that $\alpha_1 > \alpha_2$, then we will say that $\langle \mathcal{A}_1, Q_1, \alpha_1 \rangle$ is* strictly preferred *over $\langle \mathcal{A}_2, Q_2, \alpha_2 \rangle$, denoted $\langle \mathcal{A}_2, Q_2, \alpha_2 \rangle \succ \langle \mathcal{A}_1, Q_1, \alpha_1 \rangle$. Otherwise, if $\alpha_1 = \alpha_2$ we will say that both arguments are* equi-preferred*, denoted $\langle \mathcal{A}_2, Q_2, \alpha_2 \rangle \approx \langle \mathcal{A}_1, Q_1, \alpha_1 \rangle$.*

Definition 6 (Defeat) *Let $\mathcal{P}$ be a program, and let $\langle \mathcal{A}_1, Q_1, \alpha_1 \rangle$ and $\langle \mathcal{A}_2, Q_2, \alpha_2 \rangle$ be two arguments in $\mathcal{P}$. We will say that $\langle \mathcal{A}_1, Q_1, \alpha_1 \rangle$* defeats *$\langle \mathcal{A}_2, Q_2, \alpha_2 \rangle$ (or equivalently $\langle \mathcal{A}_1, Q_1, \alpha_1 \rangle$ is a* defeater *for $\langle \mathcal{A}_2, Q_2, \alpha_2 \rangle$) iff (1) Argument $\langle \mathcal{A}_1, Q_1, \alpha_1 \rangle$ counterargues argument $\langle \mathcal{A}_2, Q_2, \alpha_2 \rangle$ with disagreement subargument $\langle \mathcal{A}, Q, \alpha \rangle$; and (2) Either it holds that $\langle \mathcal{A}_1, Q_1, \alpha_1 \rangle \succ \langle \mathcal{A}, Q, \alpha \rangle$, in which case $\langle \mathcal{A}_1, Q_1, \alpha_1 \rangle$ will be called a* proper defeater *for $\langle \mathcal{A}_2, Q_2, \alpha_2 \rangle$, or $\langle \mathcal{A}_1, Q_1, \alpha_1 \rangle \approx \langle \mathcal{A}, Q, \alpha \rangle$, in which case $\langle \mathcal{A}_1, Q_1, \alpha_1 \rangle$ will be called a* blocking defeater *for $\langle \mathcal{A}_2, Q_2, \alpha_2 \rangle$.*

As in most argumentation systems [8, 20], P-DeLP relies on an exhaustive dialectical analysis which allows to determine if a given argument is *ultimately* undefeated (or *warranted*) wrt a program $\mathcal{P}$. An *argumentation line* starting in an argument $\langle \mathcal{A}_0, Q_0, \alpha_0 \rangle$ is a sequence $[\langle \mathcal{A}_0, Q_0, \alpha_0 \rangle, \langle \mathcal{A}_1, Q_1, \alpha_1 \rangle, \ldots, \langle \mathcal{A}_n, Q_n, \alpha_n \rangle, \ldots]$ that can be thought of as an exchange of arguments between two parties, a *proponent* (evenly-indexed arguments) and an *opponent* (oddly-indexed arguments). In order to avoid *fallacious* reasoning, argumentation theory imposes additional constraints on such an argument exchange to be considered rationally acceptable wrt a P-DeLP program $\mathcal{P}$, namely:

1. **Non-contradiction:** given an argumentation line λ, the set of arguments of the proponent (resp. opponent) should be *non-contradictory* wrt $\mathcal{P}$. Non-contradiction for a set of arguments is defined as follows: a set $S = \bigcup_{i=1}^{n} \{\langle \mathcal{A}_i, Q_i, \alpha_i \rangle\}$ is *contradictory* wrt $\mathcal{P}$ iff $\Pi \cup \bigcup_{i=1}^{n} \mathcal{A}_i$ is contradictory.
2. **No circular argumentation:** there are no repeated arguments in λ (i.e., if $\langle \mathcal{A}_j, Q_j, \alpha_j \rangle \in \lambda$, then it appears only once in λ).
3. **Progressive argumentation:** every blocking defeater $\langle \mathcal{A}_i, Q_i, \alpha_i \rangle$ in λ is defeated by a proper defeater $\langle \mathcal{A}_{i+1}, Q_{i+1}, \alpha_{i+1} \rangle$ in λ.

An argumentation line satisfying the above restrictions is called *acceptable*, and can be proven to be finite. Given a program $\mathcal{P}$ and an argument $\langle \mathcal{A}_0, Q_0, \alpha_0 \rangle$, the set of all acceptable argumentation lines starting in $\langle \mathcal{A}_0, Q_0, \alpha_0 \rangle$ accounts for a whole dialectical analysis for $\langle \mathcal{A}_0, Q_0, \alpha_0 \rangle$ (i.e. all possible dialogues rooted in $\langle \mathcal{A}_0, Q_0, \alpha_0 \rangle$, formalized as a *dialectical tree*, denoted $\mathcal{T}_{\langle \mathcal{A}_0, Q_0, \alpha_0 \rangle}$). Nodes in a dialectical tree $\mathcal{T}_{\langle \mathcal{A}_0, Q_0, \alpha_0 \rangle}$ can be marked as *undefeated* and *defeated* nodes (U-nodes and D-nodes, resp.). A dialectical

[5] Note that from the definition of argument, it follows that on the basis of a P-DeLP program $\mathcal{P}$ there may exist different arguments $\langle \mathcal{A}_1, Q, \alpha_1 \rangle, \langle \mathcal{A}_2, Q, \alpha_2 \rangle, \ldots, \langle \mathcal{A}_k, Q, \alpha_k \rangle$ supporting a given goal Q, with (possibly) different necessity degrees $\alpha_1, \alpha_2, \ldots, \alpha_k$.

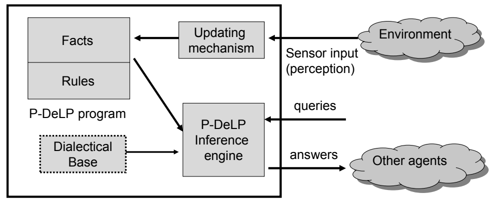

Figure 1: A P-DeLP-based agent in a MAS context

tree will be marked as an AND-OR tree: all leaves in $\mathcal{T}_{\langle \mathcal{A}_0,\, Q_0,\, \alpha_0 \rangle}$ will be marked U-nodes (as they have no defeaters), and every inner node is to be marked as *D-node* iff it has at least one U-node as a child, and as *U-node* otherwise. An argument $\langle \mathcal{A}_0, Q_0, \alpha_0 \rangle$ is ultimately accepted as *warranted* iff the root of $\mathcal{T}_{\langle \mathcal{A}_0,\, Q_0,\, \alpha_0 \rangle}$ is a *U-node*.

Definition 7 (Warrant) *Given a program $\mathcal{P}$, and a goal Q, we will say that Q is warranted wrt $\mathcal{P}$ with a necessity degree α iff there exists a warranted argument $\langle \mathcal{A}, Q, \alpha \rangle$. We will write $\mathcal{P} \mathrel{|\!\!\!\sim_w} \langle \mathcal{A}, Q, \alpha \rangle$ to denote that $\langle \mathcal{A}, Q, \alpha \rangle$ is a warranted argument on the basis of $\mathcal{P}$.*

3 Modelling Agent Reasoning in P-DeLP

In a MAS context, we propose a model in which intelligent agents will encode their knowledge about the world using a P-DeLP program [10], using the argument and warrant computing procedure to perform their inferences. Figure 1 outlines the different elements associated with a P-DeLP-based agent. Clearly, our agent will be usually performing its activities in a dynamic environment, so that it should also be able to reason, plan, and act according to new perceptions from the outside world. Such perceptions will be sensed by the agent, integrating them into its current beliefs. For the sake of simplicity, we will assume that such perceptions constitute new facts to be added to the agent's knowledge base. As already stated in the introduction, fuzzy propositions provide us with a suitable representation model as our agent will probably have vague or imprecise information about the real world, as its sensors are not perfect devices. Defining a generic procedure for updating the agent's knowledge base is not easy, as completely new incoming information (e.g. facts with new predicate names) might result in the strict knowledge Π becoming contradictory (see Def. 2). A naïve approach to model an updating procedure for P-DeLP can be found in [10]. A detailed analysis of the technical aspects concerning this problem are presented in [6].

An interesting problem arises when considering how the agent's reasoning capabilities can be captured on the basis of the P-DeLP formalism. As discussed in Section 2, P-DeLP allows an agent to construct arguments and to analyze which literals are justified by means of the definition of warrant. It must be noted that all conclusions which are based only on certain clauses can be understood as empty arguments or "theorems" that follow from the program $\mathcal{P}$. For a given program $\mathcal{P}$, let $Lit_\vdash(\mathcal{P})$ denote the set of all possible literals provable from strict knowledge, $Lit_\Delta(\mathcal{P})$ the set of all possible weighed literals which can be associated with argument conclusions and $Lit_w(\mathcal{P})$ the set of all possible weighed literals which are conclusions of warranted arguments. Formally:

$$
\begin{array}{lcl}
Lit_\vdash(\mathcal{P}) & = & \{\ (Q,1) \mid \mathcal{P} \vdash (Q,1)\ \} \\
Lit_\Delta(\mathcal{P}) & = & \{\ (Q,\alpha) \mid \text{there exists some argument } \mathcal{A} \text{ for } Q \text{ with necessity degree } \alpha\ \} \\
Lit_w(\mathcal{P}) & = & \{\ (Q,\alpha) \mid \mathcal{P} \mathrel{|\!\!\!\sim_w} \langle \mathcal{A}, Q, \alpha \rangle, \text{ for some argument } \mathcal{A} \text{ for } Q \text{ with necessity degree } \alpha\ \}
\end{array}
$$

Clearly, it holds that:

$$Lit_\vdash(\mathcal{P}) \subseteq Lit_w(\mathcal{P}) \subseteq Lit_\Delta(\mathcal{P})$$

Which is the relationship between these distinguished sets and the program $\mathcal{P}$? An answer to that question can be given in terms of those *logical properties* which relate any non-monotonic inference relationship "$\mid\!\sim$" and a set Γ of sentences. In particular, we distinguish a classical inference operator Th, which stands for theorems that follow from the theory. For an in-depth treatment see [17]. Traditionally, the logical properties analyzed in this context are the following:

1. **Inclusion (IN)**: $\Gamma \subseteq C(\Gamma)$
2. **Idempotence (ID)**: $C(\Gamma) = C(C(\Gamma))$
3. **Cumulativity (CU)**: $\gamma \in C(\Gamma)$ implies $\phi \in C(\Gamma \cup \{\gamma\})$ iff $\phi \in C(\Gamma)$, for any wffs $\gamma, \phi \in \mathcal{L}$.
4. **Monotonicity (MO)**: $\Gamma \subseteq \Phi$ implies $C(\Gamma) \subseteq C(\Phi)$
5. **Supraclassicality**: $Th(A) \subseteq C(A)$
6. **Left logical equivalence (LL)**: $Th(A) = Th(B)$ implies $C(A) = C(B)$
7. **Right weakening (RW)**: If $x \supset y \in Th(A)$ and $x \in C(A)$ then $y \in C(A)$.[6]
8. **Conjunction of conclusions (CC)**:[7] If $x \in C(A)$ and $y \in C(A)$ then $x \wedge y \in C(A)$.
9. **Subclassical cumulativity (SC)**: If $A \subseteq B \subseteq Th(A)$ then $C(A) = C(B)$.
10. **Left absorption (LA)**: $Th(C(\Gamma)) = C(\Gamma)$.
11. **Right absorption (RA)**: $C(Th(\Gamma)) = C(\Gamma)$.
12. **Rationality of negation (RN)**: if $A \mid\!\sim z$ then either $A \cup \{x\} \mid\!\sim z$ or $A \cup \{\sim x\} \mid\!\sim z$.
13. **Disjunctive rationality (DR)**: if $A \cup \{x \vee y\} \mid\!\sim z$ then $A \cup \{x\} \mid\!\sim z$ or $A \cup \{y\} \mid\!\sim z$.
14. **Rational monotonicity (RM)**: if $A \mid\!\sim z$ then either $A \cup \{x\} \mid\!\sim z$ or $A \mid\!\sim \sim x$.

Our aim is to study the behavior of a P-DeLP program (which stands for an agent's knowledge base) in the context of the above properties. In order to do this, we will define suitable inference operators for expressing argument conclusions and warranted literals.

4 Logical properties of argument and warrant in P-DeLP

First, we will formalize the notion of *expansion operator* as follows:

Definition 8 (Expansion operators $C_{\vdash}$, $C_{\triangle}$ and C_{w}) *Let $\mathcal{P}$ be a P-DeLP program. We define the operators $C_{\vdash}$, $C_{\triangle}$ and C_{w} associated with $\mathcal{P}$ as follows: (1) $C_{\vdash}(\mathcal{P}) = \mathcal{P} \cup Lit_{\vdash}(\mathcal{P})$; (2) $C_{\triangle}(\mathcal{P}) = \mathcal{P} \cup Lit_{\triangle}(\mathcal{P})$; (3) $C_{w}(\mathcal{P}) = \mathcal{P} \cup Lit_{w}(\mathcal{P})$.*

Operator $C_{\vdash}$ computes the expansion of $\mathcal{P}$ by adding new certain facts $(Q, 1)$ whenever such facts can be derived in $\mathcal{P}$ via $\vdash$.[8] Operator $C_{\triangle}$ computes the expansion of $\mathcal{P}$ with new facts corresponding to defeasible knowledge derivable as argument conclusions. $C_{\triangle}(\mathcal{P})$ incorporates a new uncertain fact (Q, α) whenever there exists an argument $\langle \mathcal{A}, Q, \alpha \rangle$ in $\mathcal{P}$. Notice that $C_{\triangle}$ may contain contradictory knowledge ($i.e.$ it may be the case that two arguments $\langle \mathcal{A}_1, Q, \alpha \rangle$ and $\langle \mathcal{A}_2, \sim Q, \beta \rangle$ could be inferred from a given program $\mathcal{P}$).[9] Finally, operator C_{w} computes a subset of $C_{\triangle}$, namely the expansion of $\mathcal{P}$ including all new facts which correspond to conclusions of warranted arguments in $\mathcal{P}$.

Proposition 9 *Operators $C_{\vdash}$, $C_{\triangle}$ and C_{w} are well-defined (ie, given a P-DeLP program $\mathcal{P}$ as input, the associated output is also a P-DeLP program $\mathcal{P}$'). Besides, they satisfy the following relationship: $C_{\vdash}(\mathcal{P}) \subseteq C_{w}(\mathcal{P}) \subseteq C_{\triangle}(\mathcal{P})$.*[10]

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

[7] Sometimes also called "Right and".

[8] Operator $C_{\vdash}$ defines in fact a classical consequence relationship, as it satisfies idempotence, cut and monotonicity. It can be seen as the SLD Horn resolution counterpart in the context of P-DeLP restricted to certain clauses.

[9] For a given goal Q, we write $\sim Q$ as an abbreviation to denote "$\sim q$" if $Q \equiv q$ and "q" if $Q \equiv \sim q$.

[10] Proofs for propositions in this paper can be found in [11, 12].

4.1 Logical properties for $C_{\triangle}$

Proposition 10 *The operator $C_{\triangle}$ satisfies inclusion and idempotence.*

Monotonicity does not hold for $C_{\triangle}$, as expected. As a counterexample consider the program $\mathcal{P}$ = { $(q, 1)$, $(p \leftarrow q, 0.9)$ }. Then $(p, 0.9) \in C_{\triangle}(\mathcal{P})$, as there is an argument $\langle \mathcal{A}, p, 0.9 \rangle$ on the basis of $\mathcal{P}$ for concluding $(p, 0.9)$, with $\mathcal{A}$ ={ $(p \leftarrow q, 0.9)$ }. However, $(p, 0.9) \notin C_{\triangle}(\mathcal{P} \cup \{(\sim p, 1)\})$ (as no argument for $(p, 0.9)$ could exist, as condition 2 in Def. 3 would be violated). Semi-monotonicity is an interesting property for analyzing non-monotonic consequence relationships. It is satisfied if all defeasible consequences from a given theory are preserved when the theory is augmented with new *defeasible* information.

Proposition 11 *The operator $C_{\triangle}$ satisfies semi-monotonicity when new defeasible information is added,* $i.e.$ *$C_{\triangle}(\mathcal{P}_1) \subseteq C_{\triangle}(\mathcal{P}_1 \cup \mathcal{P}_2)$, when $\mathcal{P}_2^{\Pi} = \emptyset$.*

Cumulativity for argument construction shows us that any argument obtained from a program $\mathcal{P}$ can be kept as an intermediate proof or lemma to be later used for building more complex arguments. Formally:

Proposition 12 *The operator $C_{\triangle}$ satisfies cumulativity,* $i.e.$ *$\gamma \in C_{\triangle}(\Gamma)$ implies $\phi \in C_{\triangle}(\Gamma \cup \{\gamma\})$ iff $\phi \in C_{\triangle}(\Gamma)$.*

Note that the property of right weakening cannot be considered (in a strict sense) in P-DeLP, since the underlying logic does not allow the application of the deduction theorem. Therefore, wffs of the form $(x \leftarrow y, \alpha)$ cannot be derived. However, an alternative approach can be intended, introducing a new property in which right weakening is restricted to Horn-like clauses:

Proposition 13 *The operator $C_{\triangle}$ satisfies (Horn) supraclassicality wrt $C_{\vdash}$ ($i.e.$ $C_{\vdash}(\mathcal{P}) \subseteq C_{\triangle}(\mathcal{P})$), and (Horn) right weakening, ($i.e.$ if $(Y, \alpha) \in C_{\triangle}(\mathcal{P})$ and $(X \leftarrow Y, 1) \in C_{\vdash}(\mathcal{P})$, then $(X, \alpha) \in C_{\triangle}(\mathcal{P})$).*

Most of the non-pure logical properties for $C_{\triangle}$ do not hold. In particular, $C_{\triangle}$ does not satisfy the properties of (LL) left-logical equivalence; (CC) conjunction of conclusions; (LA) left absorption; (RA) right absorption; (RN) rational negation; (RM) rational monotonicity; (DR) disjunctive rationality, as shown next.

LL: Given two programs $\mathcal{P}_1$ and $\mathcal{P}_2$, $C_{\vdash}(\mathcal{P}_1) = C_{\vdash}(\mathcal{P}_2)$ does not imply $C_{\triangle}(\mathcal{P}_1) = C_{\triangle}(\mathcal{P}_2)$. Consider $\mathcal{P}_1$ = { $(y \leftarrow, 1)$ } and $\mathcal{P}_2 = \mathcal{P}_1 \cup$ { $(x \leftarrow y, 0.9)$ }.

LA: Consider the program $\mathcal{P} = \{(Q, \alpha)\}$, where Q is a literal, $\alpha < 1$.
Then $C_{\vdash}(C_{\triangle}(\mathcal{P})) = C_{\vdash}(\{(Q, \alpha)\}) = \emptyset \neq C_{\triangle}(\mathcal{P})$.

RA: Consider the same counterexample given for LA. Analogously, $C_{\triangle}(C_{\vdash}(\mathcal{P})) = C_{\triangle}(\emptyset) = \emptyset \neq C_{\triangle}(\mathcal{P})$.

RN: Consider $\mathcal{P}_1$ = { $(\sim p \leftarrow x, 1)$, $(\sim p \leftarrow \sim x, 1)$, $(r \leftarrow, 1)$, $(z \leftarrow p, 1)$, $(p \leftarrow r, 0.9)$ }. Then it holds that $\mathcal{P}_1 \mathrel{|\!\!\sim_{\triangle}} \langle \mathcal{A}_1, z, 0.9 \rangle$, with $\mathcal{A}_1$ = { $(p \leftarrow r, 0.9)$ } However, $\mathcal{P}_1 \cup$ { $(x \leftarrow, 1)$ } $\not\mathrel{|\!\!\sim_{\triangle}} \langle \mathcal{A}_1, z, 0.9 \rangle$, and $\mathcal{P}_1 \cup$ { $(\sim x \leftarrow, 1)$ } $\not\mathrel{|\!\!\sim_{\triangle}} \langle \mathcal{A}_1, z, 0.9 \rangle$.

RM: Consider the same counterexample as given for RN. Then $\mathcal{P}_1 \mathrel{|\!\!\sim_{\triangle}} \langle \mathcal{A}_1, z, 0.9 \rangle$, but it is not the case that $\mathcal{P}_1 \cup$ { $(x \leftarrow, 1)$ } $\mathrel{|\!\!\sim_{\triangle}} \langle \mathcal{A}_1, z, 0.9 \rangle$ nor $\mathcal{P}_1 \mathrel{|\!\!\sim_{\triangle}} (\sim x \leftarrow, 1)$.

CC,DR: Clearly, $C_{\triangle}$ does not satisfy property CC nor DR; disjunctions and conjunctions of goals supported by an argument cannot be expressed as wffs in the P-DeLP object language.

4.2 Logical properties for C_w

In what follows we will analyze some relevant logical properties for C_w. Monotonicity does not hold for C_w, as expected. As a counterexample consider the program $\mathcal{P}$ = { $(q, 1)$, $(p \leftarrow q, 0.9)$ }. Then $(p, 0.9) \in C_w(\mathcal{P})$, as there is an undefeated argument $\langle \mathcal{A}, p, 0.9 \rangle$ on the basis of $\mathcal{P}$ for concluding $(p, 0.9)$, with $\mathcal{A}$ ={ $(p \leftarrow q, 0.9)$ }. However, $(p, 0.9) \notin C_{\triangle}(\mathcal{P} \cup \{(\sim p, 1)\})$ (as no argument for $(p, 0.9)$ could exist, as condition 2 in Def. 3 would be violated). Moreover, cumulativity, idempotence and right-weakening do not hold for C_w, as shown in the following examples.

Example 1 *Operator C_w does not satisfy idempotence. Consider program $\mathcal{P}_{sample}$ given in Fig. 2. Note that $q \notin C_w(\mathcal{P}_{sample})$: there is an argument $\langle \mathcal{A}, q, 0.7 \rangle$, with $\mathcal{A}$ ={ $(q \leftarrow z, 0.7)$, $(z \leftarrow p, 0.7)$, $(p, 0.7)$ } supporting $(q, 0.7)$. In this case, argument $\langle \mathcal{A}, q, 0.7 \rangle$ is defeated by $\langle \mathcal{B}, \sim q, 0.8 \rangle$, with $\mathcal{B}$ ={ $(\sim q \leftarrow r, 0.8)$, $(r, 0.8)$ }. There is a third argument $\langle \mathcal{C}, \sim r, 0.9 \rangle$, with $\mathcal{C}$ ={ $(\sim r, 0.9)$ }. Even though this argument defeats*

(1)	$(\sim y \leftarrow p, \sim r, 1)$	(5)	$(q \leftarrow z, 0.7)$
(2)	$(y, 1)$	(6)	$(z \leftarrow p, 0.7)$
(3)	$(p, 0.7)$	(7)	$(\sim q \leftarrow r, 0.8)$
(4)	$(r, 0.8)$	(8)	$(\sim r, 0.9)$

Figure 2: Program $\mathcal{P}_{sample}$ (see examples 1 and 2)

$\langle \mathcal{B}, \sim q, 0.8 \rangle$, it cannot be introduced as a defeater in the above analysis, as it would be in conflict with argument $\langle \mathcal{A}, q, 0.7 \rangle$, violating the non-contradiction consistency constraint in argumentation lines (since $(\sim y, 1)$ and $(y, 0.7)$ would follow from $\mathcal{P}^{\Pi}_{sample} \cup \mathcal{A} \cup \mathcal{B}$, where $\mathcal{P}^{\Pi}_{sample}$ stands for the certain knowledge in $\mathcal{P}_{sample}$. The set of all warranted literals supported by $\mathcal{P}_{sample}$ is W = { $(p, 0.7)$, $(z, 0.7)$, $(\sim r, 0.9)$ }. Consider now the program $\mathcal{P}' = \mathcal{P}_{sample} \cup W$. Let us analyze whether q is warranted or not wrt $\mathcal{P}'$. There is an argument $\langle \mathcal{A}', q, 0.7 \rangle$, with $\mathcal{A}' = \{(q \leftarrow z, 0.7)\}$, which is defeated by $\langle \mathcal{B}, \sim q, 0.8 \rangle$ (as before). This defeater is defeated by $\langle \mathcal{C}', \sim r, 0.9 \rangle$, with $\mathcal{C}' = \emptyset$. There are no more arguments to consider, and therefore $(q, 0.7)$ is warranted. Hence $q \in C_w(\mathcal{P}') = C_w(C_w(\mathcal{P}_{sample}))$, and as shown above $q \notin C_w(\mathcal{P}_{sample})$. Therefore C_w does not satisfy idempotence.

Example 2 *Operator C_w does not satisfy cumulativity. We must show that there exists a weighed literal for some program $\mathcal{P}$ such that if $(Q, \alpha) \in C_w(\mathcal{P})$, then $(R, \beta) \in C_w(\mathcal{P} \cup \{(Q, \alpha)\})$ does not imply $(R, \beta) \in C_w(\mathcal{P})$. Consider program $\mathcal{P}_{sample}$ in Fig. 2. As shown in Example 1, $(z, 0.7) \in C_w(\mathcal{P}_{sample})$, and $(q, 0.7) \in C_w(\mathcal{P}_{sample} \cup \{(z, 0.7)\})$. However, $(q, 0.7) \notin C_w(\mathcal{P}_{sample})$. Hence cumulativity does not hold for C_w.*

Example 3 *Operator C_w does not satisfy right weakening. Consider program $\mathcal{P}_{sample}$ in Fig. 2. Note that $(p, 0.7) \in C_w(\mathcal{P}_{sample})$ and $(\sim r, 0.9) \in C_w(\mathcal{P}_{sample})$. Besides, $(\sim y \leftarrow p, \sim r, 1) \in \mathcal{P}^{\Pi}_{sample}$. However, the conclusion of this certain rule is* not *warranted, $i.e.$ $(\sim y, 0.7) \notin C_w(\mathcal{P}_{sample})$, since $(y, 1) \in \mathcal{P}^{\Pi}_{sample}$ and thus there exists no argument with conclusion $(\sim y, 0.7)$ (as it would violate condition 2 in Def. 3).*

Proposition 14 summarizes the properties that hold for C_w. Notice that C_w satisfies inclusion trivially (by definition).

Proposition 14 *The operator C_w satisfies inclusion, (Horn) supraclassicality wrt $C_{\vdash}$ ($i.e.$ $C_{\vdash}(\mathcal{P}) \subseteq C_w(\mathcal{P})$) and subclassical cumulativity, $i.e.$ $\mathcal{P}_1 \subseteq \mathcal{P}_2 \subseteq C_{\vdash}(\mathcal{P}_1)$ implies $C_w(\mathcal{P}_1)$ = $C_w(\mathcal{P}_2)$.*

Operator C_w does not satisfy the properties of LL, CC, LA, RA, RN, RM and DR. In all cases this is based on the impossibility of computing arguments satisfying these properties. Suitable counterexamples can be found in [11].

5 Discussion. Related work

Research in logical properties for defeasible argumentation can be traced back to Benferhat $et\ al.$ [3, 4] and Vreeswijk [23]. In the context of his abstract argumentation systems, Vreeswijk showed that many logical properties for non-monotonic inference relationships turned out to be counter-intuitive for argument-based systems. Benferhat $et\ al.$ [3] were the first who studied argumentative inference in uncertain and inconsistent knowledge bases. They defined an argumentative consequence relationship $\vdash_{\mathcal{A}}$ taking into account the existence of arguments favoring a given conclusion against the absence of arguments in favor of its contrary. In contrast, the $\mid\!\sim_w$ relationship proposed in this paper takes into account the *whole* dialectical analysis for arguments derivable from the program for any given goal.

In [3, 4] the authors also extend the argumentative relation $\vdash_{\mathcal{A}}$ to prioritized knowledge bases, assessing weights to conclusions on the basis of the $\vdash_{\pi}$-entailment relationship from possibilistic logic [14]. A direct comparison to our $\mid\!\sim_w$ relationship is not easy since we are using a logic programming framework and not general propositional logic, but roughly speaking while $\vdash_{\pi}$ takes into account the inconsistency degree associated with the whole knowledge base, our logic programming framework allows us to perform a dialectical analysis restricted only to conflicting arguments related with the goal being solved.

The complexity of computing warranted beliefs can be better understood in the light of the logical properties for C_w presented in this paper. There are only three properties (inclusion, supraclassicality and subclassical cumulativity) which hold for this operator. Next we will briefly discuss some of the relevant properties which do not hold for C_w. In [20] some examples are informally presented to show that argumentation systems should assign facts a special status, and therefore should *not* be cumulative. In the particular case of cumulativity (traditionally the most defended property associated with non-monotonic inference), we have shown that it does not hold for C_w even when warranted conclusions are assigned the epistemic status of uncertain facts of the form (Q, α), $\alpha < 1$, which provides an even stronger result than the one suggested originally in [20].

Horn right weakening indicates that a certain rule of the form $(Y \leftarrow X, 1)$ does *not* ensure that every warranted argument for (X, α) (with $\alpha < 1$) implies that (Y, α) is also warranted. In fact, it can be the case that the certain fact $(\sim Y, 1)$ is present in a given program, so that an argument for the goal Y cannot be even computed (as shown in Example 3). In a recent paper [5], Caminada & Amgoud identify this situation as a particular anomaly in several argumentation formalisms ($e.g.$ [19, 15]) and provide an interesting solution in terms of *rationality postulates* which –the authors claim– should hold in any well-defined argumentative system. In the case of P-DeLP the problem seems to require a different conceptualization, as the necessity degree 1 of the rule $(Y \leftarrow X, 1)$ is attached to the rule itself, and the necessity degree of the conclusion Y *depends* on the necessity degree α of the antecedent X. As an example, consider the program $\mathcal{P} = \{ (\sim g \leftarrow a, 1), (a, 0.7), (g \leftarrow b, 1), (b, 0.4) \}$. In this case, $(a, 0.7)$ and $(b, 0.4)$ are warranted conclusions. However, we cannot warrant g and $\sim g$ with necessity degree 1. In fact, only $(\sim g, 0.7)$ can be warranted. In this respect, the behavior of strict rules (as used in most argumentation systems) seems to be different from the behavior of certain rules in our framework.

6 Conclusions. Future work

In this paper we have shown that P-DeLP provides a useful framework for making a formal analysis of logical properties in defeasible argumentation. We contend that a formal analysis of defeasible consequence is mandatory to get an in-depth understanding of the behavior of argumentation frameworks, particularly when used for modelling reasoning in intelligent agents. Expansion operators like $C_\triangle$ and C_w provide a natural tool for characterizing that behavior, as well as useful criteria when developing new argumentation frameworks and assessing their expressive power.

Our current research work in P-DeLP will follow two main directions: on the one hand, we are concerned with characterizing different *degrees* of non-monotonicity. We think that the C_w operator can be used to better understand how complex non-monotonic systems behave. On the other hand, we will extend the current formalization to include fuzzy constants and thus fuzzy unification features [2].

Acknowledgments: We want to thank anonymous reviewers for their comments which helped improved the final version of this paper. We also want to thank Leila Amgoud for interesting discussions concerning some of the properties associated with argument-based inference. This work was supported by Spanish Projects TIC2003-00950, TIN2004-07933-C03-01/03, TIN2004-07933-C03-03, by Ramón y Cajal Program (MCyT, Spain), by PICT 2002 No. 13.096 (Agencia Nacional de Promoción Científica y Tecnológica, Argentina), and by CONICET (Argentina).

References

[1] T. Alsinet and L. Godo. A complete calculus for possibilistic logic programming with fuzzy propositional variables. In *Proc. of the UAI-2000 Conference*, pages 1–10, 2000.

[2] T. Alsinet and L. Godo. A proof procedure for possibilistic logic programming with fuzzy constants. In *Proc. of the ECSQARU-2001 Conference*, pages 760–771, 2001.

[3] S. Benferhat, D. Dubois, and H. Prade. Argumentative inference in uncertain and inconsistent knowledge bases. In *Proc. of UAI*, pages 411–419, 1993.

[4] S. Benferhat, D. Dubois, and H. Prade. Some syntactic approaches to the handling of inconsistent knowledge bases: A comparative study. part ii: The prioritized case. In Ewa Orlowska, editor, *Logic at work*, volume 24, pages 473–511. Physica-Verlag , Heidelberg, 1998.

[5] M. Caminada and L. Amgoud. An axiomatic account of formal argumentation. In *Proc. of AAAI 2005 Conference. Pittsburgh, USA*, 2005.

[6] M. Capobianco, C. Chesñevar, and G. Simari. Argumentation and the dynamics of warranted beliefs in changing environments. *Intl. Journal on Autonomous Agents and Multiagent Systems (JAAMAS)*, 11:127–151, September 2005.

[7] C. Chesñevar and A. Maguitman. An Argumentative Approach to Assessing Natural Language Usage based on the Web Corpus. In *Proc. ECAI Conf., Spain*, pages 581–585, 2004.

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

[9] C. Chesñevar, A. Maguitman, and G. Simari. A first approach to argument-based recommender systems based on defeasible logic programming. In *Proc. 10th Intl. Workshop on Non-Monotonic Reasoning. Whistler, Canada*, pages 109–117, June 2004.

[10] C. Chesñevar, G. Simari, T. Alsinet, and L. Godo. Modelling Agent Reasoning in a Logic Programming Framework for Possibilistic Argumentation. In *Proc. 2nd. European Workshop on Multiagent Systems. Barcelona, Spain*, pages 135–142, December 2004.

[11] C. Chesñevar, G. Simari, L. Godo, and T. Alsinet. Argument-based expansion operators in possibilistic defeasible logic programming: Characterization and logical properties. In *Proc. of ECSQARU 2005 Conference. Barcelona, Spain*, pages 353–365, July 2005.

[12] C. Chesñevar, G. Simari, L. Godo, and T. Alsinet. *Artificial Intelligence Research and Development (Series Frontiers in Artificial Intelligence and Applications)*, volume 131, chapter *On Warranted Inference in Possibilistic Defeasible Logic Programming*, pages 265–272. IOS Press (ISBN 1-58603-560-6), 2005.

[13] Carlos I. Chesñevar, Guillermo Simari, Teresa Alsinet, and Lluis Godo. A Logic Programming Framework for Possibilistic Argumentation with Vague Knowledge. In *Proc. Intl. Conf. in Uncertainty in Artificial Intelligence (UAI 2004). Banff, Canada*, pages 76–84, July 2004.

[14] D. Dubois, J. Lang, and H. Prade. Possibilistic logic. In D.Gabbay, C.Hogger, and J.Robinson, editors, *Handbook of Logic in Art. Int. and Logic Prog. (Nonmonotonic Reasoning and Uncertain Reasoning)*, pages 439–513. Oxford Univ. Press, 1994.

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

[16] V. Lifschitz. Foundations of logic programming. In *Principles of Knowledge Representation*, pages 69–127. CSLI Publications, 1996.

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

[18] S. Parsons, C. Sierra, and N. Jennings. Agents that reason and negotiate by arguing. *Journal of Logic and Computation*, 8(3):261–292, 1998.

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

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

[21] I. Rahwan, S. Ramchurn, N. Jennings, P. McBurney, S. Parsons, and L. Sonenberg. Argumentation-based negotiation. *Knowledge Engineering Review*, 18(4), 2003.

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

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