

Argument-based Expansion Operators in Possibilistic Defeasible Logic Programming: Characterization and Logical Properties

Carlos I. Chesñevar¹, Guillermo R. Simari², Lluís Godo³, and Teresa Alsinet¹

¹ Departament of Computer Science – Universitat de Lleida

C/Jaume II, 69 – 25001 Lleida, SPAIN – EMAIL: {cic,tracy}@eps.udl.es

² Department of Computer Science and Engineering – Universidad Nacional del Sur
Alem 1253, (8000) Bahía Blanca, ARGENTINA – EMAIL: grs@cs.uns.edu.ar

³ Artificial Intelligence Research Institute (IIIA-CSIC)

Campus UAB - 08193 Bellaterra, Barcelona, SPAIN – EMAIL: godo@iiia.csic.es

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. From a logical viewpoint, 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 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, resp. Different logical properties for the proposed expansion operators are studied and contrasted with a traditional SLD-based Horn logic. We will show that this analysis provides useful comparison criteria that can be extended and applied to other argumentation frameworks.

KEY WORDS: argumentation, logic programming, uncertainty, non-monotonic inference

1 Introduction and motivations

Possibilistic Defeasible Logic Programming (P-DeLP) [1] 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 [2, 3], 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 [4] and natural language processing [5], among others.

Formalizing argument-based reasoning by means of suitable inference operators offers a useful tool. 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. This paper analyzes two non-monotonic *expansion operators* for P-DeLP, intended for modelling the effect of expanding a given program by introducing new facts, associated with argument conclusions and warranted literals, respectively. Their 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 in an argumentative framework like P-DeLP provide an interesting counterpart to traditional consequence operators in logic programming [6]. Our approach differs from such consequence operators as we want to analyze the role of argument conclusions and warranted literals when represented as new weighed facts in the context of object-level program clauses.

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 [7] and not on PGL itself (which involves fuzzy propositions). The rest of the paper is structured as follows: first in Section 2 we outline some fundamentals of (non-monotonic) inference relationships. Section 3 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. Finally, in Section 5 we discuss related work the most important conclusions that have been obtained.

2 Non-monotonic Inference Relationships: fundamentals

In classical logic, inference rules allow us to determine whether a given wff γ follows via “ \vdash ” from a set Γ of wffs, where “ \vdash ” is a *consequence relationship* (satisfying idempotence, *cut* and monotonicity). As non-monotonic and defeasible logics evolved into a valid alternative to formalize commonsense reasoning a similar concept was needed to capture the notion of logical consequence without demanding some of these requirements (*e.g.* monotonicity). This led to the definition of a more generic notion of inference in terms of *inference relationships*. Given a set Γ of wffs in an arbitrary logical language \mathcal{L} , we write $\Gamma \sim \gamma$ to denote an inference relationship “ \sim ”, where γ is a (non-monotonic) consequence of Γ . We define an *inference operator* C_{\sim} associated with an inference relationship, with $C_{\sim}(\Gamma) = \{\gamma \mid \Gamma \sim \gamma\}$. Given an inference relationship “ \sim ” and a set Γ of sentences, the following are called *basic* (or *pure*) properties associated with the inference operator $C_{\sim}(\Gamma)$:

1. **Inclusion (IN)**: $\Gamma \subseteq C(\Gamma)$
2. **Idempotence (ID)**: $C(\Gamma) = C(C(\Gamma))$
3. **Cut (CT)**: $\Gamma \subseteq \Phi \subseteq C(\Gamma)$ implies $C(\Phi) \subseteq C(\Gamma)$
4. **Cautious monotonicity (CM)**: $\Gamma \subseteq \Phi \subseteq C(\Gamma)$ implies $C(\Gamma) \subseteq C(\Phi)$.
5. **Cumulativity (CU)**: $\gamma \in C(\Gamma)$ implies $\phi \in C(\Gamma \cup \{\gamma\})$ iff $\phi \in C(\Gamma)$, for any wffs $\gamma, \phi \in \mathcal{L}$.
6. **Monotonicity (MO)**: $\Gamma \subseteq \Phi$ implies $C(\Gamma) \subseteq C(\Phi)$

These properties are called *pure*, since they can be applied to any language \mathcal{L} , and are abstractly defined for an arbitrary inference relationship “ \sim ”. Nevertheless, other properties which link a classical inference operator Th with an arbitrary inference relationship can be stated. Next we summarize the most important non-pure properties (for an in-depth discussion, see [8]).

1. **Supraclassicality**: $Th(A) \subseteq C(A)$
2. **Left logical equivalence (LL)**: $Th(A) = Th(B)$ implies $C(A) = C(B)$
3. **Right weakening (RW)**: If $x \supset y \in Th(A)$ and $x \in C(A)$ then $y \in C(A)$.⁴
4. **Conjunction of conclusions (CC)**: If $x \in C(A)$ and $y \in C(A)$ then $x \wedge y \in C(A)$.
5. **Subclassical cumulativity (SC)**: If $A \subseteq B \subseteq Th(A)$ then $C(A) = C(B)$.
6. **Left absorption (LA)**: $Th(C(\Gamma)) = C(\Gamma)$.
7. **Right absorption (RA)**: $C(Th(\Gamma)) = C(\Gamma)$.
8. **Rationality of negation (RN)**: if $A \sim z$ then either $A \cup \{x\} \sim z$ or $A \cup \{\sim x\} \sim z$.
9. **Disjunctive rationality (DR)**: if $A \cup \{x \vee y\} \sim z$ then $A \cup \{x\} \sim z$ or $A \cup \{y\} \sim z$.
10. **Rational monotonicity (RM)**: if $A \sim z$ then either $A \cup \{x\} \sim z$ or $A \sim \sim x$.

3 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) $\{p, q, \dots\}$ 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 $Q \leftarrow L_1 \wedge \dots \wedge L_n$, where Q, L_1, \dots, 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 $Q \in \mathcal{L}$.⁵ 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 [1] is based on Possibilistic Gödel Logic or PGL [2], which is able to model both uncertainty and fuzziness and allows for

⁴ It should be noted that “ \supset ” stands for material implication, to be distinguished from the symbol “ \leftarrow ” used in a logic programming setting.

⁵ Note that a conjunction of literals is not a valid goal.

a partial matching mechanism between fuzzy propositional variables. As mentioned before, 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 [7]. As a consequence, possibilistic models are defined by possibility distributions on the set of classical interpretations⁶ and the proof method for our P-DeLP formulas, written \vdash , is defined by derivation based on the following generalized modus ponens rule (GMP).

Generalized modus ponens (GMP):

$$\frac{(L_0 \leftarrow L_1 \wedge \dots \wedge L_k, \gamma) \quad (L_1, \beta_1), \dots, (L_k, \beta_k)}{(L_0, \min(\gamma, \beta_1, \dots, \beta_k))}$$

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.

From now on, and if not stated otherwise, we will simply use P-DeLP to actually refer to the *non-fuzzy* fragment of the original P-DeLP.

3.1 Argumentation in P-DeLP

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 deemed 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}^7 . 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:

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}^Π (resp. \mathcal{P}^Δ) to identify the set of certain (resp. uncertain) clauses in \mathcal{P} .*

The following notion of argument, very similar to [9, 10], is an extension of that in argumentation systems by considering the necessity degree with which the argument supports a conclusion.

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 $\Pi \cup \mathcal{A}_1 \vdash (Q, \beta)$, $\beta > 0$.*

⁶ 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.

⁷ Notice that this notion of contradiction corresponds to the case when the inconsistency degree of Γ is strictly positive as defined in possibilistic logic.

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} .

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 $\langle \mathcal{A}_1, Q, \alpha_1 \rangle, \langle \mathcal{A}_2, Q, \alpha_2 \rangle, \dots, \langle \mathcal{A}_k, Q, \alpha_k \rangle$ supporting a given goal Q , with (possibly) different necessity degrees $\alpha_1, \alpha_2, \dots, \alpha_k$. Given a program $\mathcal{P} = (II, \Delta)$, we will write $\mathcal{P} \vdash_{\Delta} \langle \mathcal{A}, Q, \alpha \rangle$ to denote that the argument $\langle \mathcal{A}, Q, \alpha \rangle$ can be obtained from \mathcal{P} . Actually, the set $Arg(\mathcal{P}) = \{ \langle \mathcal{A}, Q, \alpha \rangle \mid \mathcal{P} \vdash_{\Delta} \langle \mathcal{A}, Q, \alpha \rangle \}$ of arguments with respect to a program \mathcal{P} can be built by means of the following complete set of procedural rules:

1 Building arguments from facts (INTF)

If $\langle Q, 1 \rangle \in II$, then $\langle \emptyset, Q, 1 \rangle \in Arg(\mathcal{P})$

If $\langle Q, \alpha \rangle \in \Delta$ then $\langle \{ \langle Q, \alpha \rangle \}, Q, \alpha \rangle \in Arg(\mathcal{P})$

2 Building Arguments by GMP (MPA):

If $\langle \mathcal{A}_1, L_1, \alpha_1 \rangle \langle \mathcal{A}_2, L_2, \alpha_2 \rangle \dots \langle \mathcal{A}_k, L_k, \alpha_k \rangle \in Arg(\mathcal{P})$

and $II \cup \{ (L_0 \leftarrow L_1 \wedge L_2 \wedge \dots \wedge L_k, \gamma) \} \cup \bigcup_{i=1}^k \mathcal{A}_i \not\vdash \perp$

and $\langle L_0 \leftarrow L_1 \wedge L_2 \wedge \dots \wedge L_k, \gamma \rangle \in \Delta$

then $\langle \bigcup_{i=1}^k \mathcal{A}_i \cup \{ (L_0 \leftarrow L_1 \wedge L_2 \wedge \dots \wedge L_k, \gamma) \}, L_0, \beta \rangle \in Arg(\mathcal{P})$,

with $\beta = \min(\alpha_1, \dots, \alpha_k, \gamma)$.

3 Extending Arguments (EAR):

If $\langle \mathcal{A}, P, \alpha \rangle \in Arg(\mathcal{P})$, and $II \cup \{ \langle P, \alpha \rangle \} \vdash \langle Q, \alpha \rangle$ then $\langle \mathcal{A}, Q, \alpha \rangle \in Arg(\mathcal{P})$.

3.2 Computing Warrant in P-DeLP

As in most argumentation formalisms (see e.g. [11, 12]), in P-DeLP it can be the case that there exist conflicting arguments. This is formalized next by the notions of counterargument and defeat.

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 $II \cup \{ \langle Q_1, \alpha_1 \rangle, \langle Q, \beta \rangle \}$ is contradictory.

Defeat among arguments involves a *preference criterion* on conflicting arguments, defined on the basis of necessity measures associated with arguments.

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 [12,11], 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 $[\langle \mathcal{A}_0, Q_0, \alpha_0 \rangle, \langle \mathcal{A}_1, Q_1, \alpha_1 \rangle, \dots, \langle \mathcal{A}_n, Q_n, \alpha_n \rangle, \dots]$ 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:** no argument $\langle \mathcal{A}_j, Q_j, \alpha_j \rangle$ in λ is a sub-argument of an argument $\langle \mathcal{A}_i, Q_i, \alpha_i \rangle$ in λ , $i < j$.
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 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 valid (or *warranted*) iff the root of $\mathcal{T}_{\langle \mathcal{A}_0, Q_0, \alpha_0 \rangle}$ is labelled as *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 $\mathcal{P} \vdash_w \langle \mathcal{A}, Q, \alpha \rangle$ to denote that $\langle \mathcal{A}, Q, \alpha \rangle$ is a warranted argument on the basis of \mathcal{P} .*

4 Logical properties of argument and warrant in P-DeLP

Our aim is to study the behavior of P-DeLP programs in the context of non-monotonic inference relationships. In order to do this, we will define different inference operators associated with arguments and with warranted goals. As stated in Section 1, we refer to such operators as *expansion operators* in order to stress the fact that their output is associated with the effect of expanding a given program \mathcal{P} given as an input by adding new weighed facts. Formally:

Definition 8 (Expansion operators C_{\perp} , C_{Δ} and C_w). *Let \mathcal{P} be a P-DeLP program. We define the operators C_{\perp} , C_{Δ} and C_w associated with \mathcal{P} as follows:*

$$C_{\perp}(\mathcal{P}) = \mathcal{P} \cup \{ (Q, 1) \mid \mathcal{P} \vdash (Q, 1) \}$$

$$\begin{aligned}
C_{\Delta}(\mathcal{P}) &= \mathcal{P} \cup \{ (Q, \alpha) \mid \mathcal{P} \vdash_{\Delta} \langle \mathcal{A}, Q, \alpha \rangle, \text{ for some argument } \mathcal{A} \\
&\quad \text{for a goal } Q \text{ with necessity degree } \alpha \} \\
C_w(\mathcal{P}) &= \mathcal{P} \cup \{ (Q, \alpha) \mid \mathcal{P} \vdash_w \langle \mathcal{A}, Q, \alpha \rangle, \text{ for some argument } \mathcal{A} \\
&\quad \text{for a goal } Q \text{ with necessity degree } \alpha \}
\end{aligned}$$

Operator C_{\perp} 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_{Δ} computes the expansion of \mathcal{P} with new facts corresponding to defeasible knowledge derivable as argument conclusions. $C_{\Delta}(\mathcal{P})$ incorporates a new uncertain fact (Q, α) whenever there exists an argument $\langle \mathcal{A}, Q, \alpha \rangle$ in \mathcal{P} . Notice that C_{Δ} 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_{Δ} , namely the expansion of \mathcal{P} including all new facts which correspond to conclusions of warranted arguments in \mathcal{P} .

Proposition 1. *Operators C_{\perp} , C_{Δ} 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: $C_{\perp}(\mathcal{P}) \subseteq C_w(\mathcal{P}) \subseteq C_{\Delta}(\mathcal{P})$.*

Proof. Given a P-DeLP program \mathcal{P} , we want to determine that $C_{\perp}(\mathcal{P})$, $C_{\Delta}(\mathcal{P})$ and $C_w(\mathcal{P})$ are also programs. From Def. 8, it is clear that all operators return syntactically valid programs as their output. From Def. 2, it remains to check that the strict knowledge $C_{\perp}(\mathcal{P})^H$ (analogously for $C_{\Delta}(\mathcal{P})$ and $C_w(\mathcal{P})$) is not a contradictory set of P-DeLP clauses. Let us suppose that $C_{\perp}(\mathcal{P})^H$ is contradictory. By definition of C_{\perp} , this is only possible if \mathcal{P}^H is contradictory, which cannot be the case, as \mathcal{P} is a P-DeLP program (absurd). Consequently, $C_{\perp}(\mathcal{P})$ is a P-DeLP program. The same line of reasoning applies for $C_{\Delta}(\mathcal{P})$ and $C_w(\mathcal{P})$. The inclusion relationship $C_{\perp}(\mathcal{P}) \subseteq C_{\Delta}(\mathcal{P})$ holds as it can be shown that $\mathcal{P} \vdash (Q, 1)$ iff $\mathcal{P} \vdash_{\Delta} \langle \emptyset, Q, 1 \rangle$. Since every warranted argument is an argument wrt \mathcal{P} , a similar analysis applies to conclude that $C_w(\mathcal{P}) \subseteq C_{\Delta}(\mathcal{P})$.

4.1 Logical properties for C_{Δ}

Proposition 2. *The operator C_{Δ} satisfies inclusion and idempotence.*

Proof. Inclusion holds trivially, as $\mathcal{P} \subseteq C_{\Delta}(\mathcal{P}) \subseteq C_{\Delta}(C_{\Delta}(\mathcal{P}))$ by definition of C_{Δ} . Proof for idempotence is not included for space reasons.

Monotonicity does not hold for C_{Δ} , as expected. As a counterexample consider the program $\mathcal{P} = \{ (q, 1), (p \leftarrow q, 0.9) \}$. Then $(p, 0.9) \in C_{\Delta}(\mathcal{P})$, 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, $(p, 0.9) \notin C_{\Delta}(\mathcal{P} \cup \{ (\sim p, 1) \})$ (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.

⁸ Operator C_{\perp} defines in fact a 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 $Q \equiv \sim q$.

Proposition 3. *The operator C_{Δ} satisfies semi-monotonicity when new defeasible information is added, i.e. $C_{\Delta}(\mathcal{P}_1) \subseteq C_{\Delta}(\mathcal{P}_1 \cup \mathcal{P}_2)$, when $\mathcal{P}_2^H = \emptyset$.*

Proof. Follows directly from the structure of the inference rules for \vdash_{Δ} . Suppose $\mathcal{P}_1 \vdash_{\Delta} \langle \mathcal{A}, Q, \alpha \rangle$, and consider a program \mathcal{P}_2 such that $\mathcal{P}_2^H = \emptyset$. Clearly, $\langle \mathcal{A}, Q, \alpha \rangle$ can be derived from $\mathcal{P}_1 \cup \mathcal{P}_2$ by applying the same sequence of steps as in $\mathcal{P}_1 \vdash_{\Delta} \langle \mathcal{A}, Q, \alpha \rangle$, since all preconditions in inference rules are defined wrt \mathcal{P}_1^H , the set of strict knowledge in \mathcal{P}_1 , and by hypothesis, $(\mathcal{P}_1 \cup \mathcal{P}_2)^H = \mathcal{P}_1^H$.

Proposition 4. *The operator C_{Δ} satisfies cumulativity, i.e. $\gamma \in C_{\Delta}(\Gamma)$ implies $\phi \in C_{\Delta}(\Gamma \cup \{\gamma\})$ iff $\phi \in C_{\Delta}(\Gamma)$.*

Proof. (Sketch) Without loss of generality, we can assume $\gamma = (Q, \alpha)$ is not in Γ (otherwise the proof is straightforward). By hypothesis, $(Q, \alpha) \in C_{\Delta}(\Gamma)$ and there is a sequence $s_1^Q, s_2^Q, \dots, s_t^Q$ of application of inference rules in $\{ \text{INTF}, \text{MPA}, \text{EAR} \}$ such that $\Gamma \vdash_{\Delta} \langle \mathcal{A}_1, Q, \alpha \rangle$. Let us assume now that $(R, \beta) \in C_{\Delta}(\Gamma \cup \{(Q, \alpha)\})$. This means that there is a sequence r_1, r_2, \dots, r_n of application of inference rules as before such that $\Gamma \vdash_{\Delta} \langle \mathcal{A}_2, R, \beta \rangle$. Suppose now that $\langle \mathcal{A}_2, R, \beta \rangle$ does not include $\langle \mathcal{A}_1, Q, \alpha \rangle$ as a subargument. This happens iff from the structure of inference rules for \vdash_{Δ} , (Q, α) will not be required as intermediate step in the proof of (R, β) iff $(R, \beta) \in C_{\Delta}(\Gamma)$. Suppose now that $\langle \mathcal{A}_2, R, \beta \rangle$ does include $\langle \mathcal{A}_1, Q, \alpha \rangle$ as a subargument. This happens iff in the sequence r_1, r_2, \dots, r_n we have that $r_{i+k} = s_i^Q$, for $i = 1 \dots t$, for some $1 \leq k \leq n$. But from the initial hypothesis this sequence can be built from Γ alone. Hence $\Gamma \vdash_{\Delta} \langle \mathcal{A}_2, R, \beta \rangle$ or equivalently $(R, \beta) \in C_{\Delta}(\Gamma)$.

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

Proof. Supraclassicality follows from Prop. 1. For the case of right weakening, let us suppose $(Y, \alpha) \in C_{\Delta}(\mathcal{P})$, i.e. $\mathcal{P} \vdash_{\Delta} \langle \mathcal{A}_1, Y, \alpha \rangle$, for some argument $\langle \mathcal{A}_1, Y, \alpha \rangle$. If $(X \leftarrow Y, 1) \in C_{\perp}(\mathcal{P})$, then necessarily $(X \leftarrow Y, 1) \in \mathcal{P}^H$ (by def. of C_{\perp}). From $(X \leftarrow Y, 1) \in \mathcal{P}$ and $\mathcal{P} \vdash_{\Delta} \langle \mathcal{A}_1, Y, \alpha \rangle$, by applying inference rule EAR we get $\langle \mathcal{A}_1, X, \alpha \rangle$.

Proposition 6. *The operator C_{Δ} satisfies subclassical cumulativity, i.e. $\mathcal{P}_1 \subseteq \mathcal{P}_2 \subseteq C_{\perp}(\mathcal{P}_1)$ implies $C_{\Delta}(\mathcal{P}_1) = C_{\Delta}(\mathcal{P}_2)$.*

Most non-pure logical properties for C_{Δ} do not hold. In particular, C_{Δ} 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 , $C_{\perp}(\mathcal{P}_1) = C_{\perp}(\mathcal{P}_2)$ does not imply $C_{\Delta}(\mathcal{P}_1) = C_{\Delta}(\mathcal{P}_2)$. Consider $\mathcal{P}_1 = \{ (y \leftarrow , 1) \}$ and $\mathcal{P}_2 = \mathcal{P}_1 \cup \{ (x \leftarrow y, 0.9) \}$.
- CC: Arguments supporting conjunctions of conclusions cannot be expressed in P-DeLP language, as goals are restricted to literals.
- LA: Consider the program $\mathcal{P} = \{(Q, \alpha)\}$, where Q is a literal, $\alpha < 1$. Then $C_{\perp}(C_{\Delta}(\mathcal{P})) = C_{\perp}(\{(Q, \alpha)\}) = \emptyset \neq C_{\Delta}(\mathcal{P})$.
- RA: Consider the same counterexample given for LA. Analogously, $C_{\Delta}(C_{\perp}(\mathcal{P})) = C_{\Delta}(\emptyset) = \emptyset \neq C_{\Delta}(\mathcal{P})$.
- 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 $\mathcal{P}_1 \vdash_{\Delta} \langle \mathcal{A}_1, z, 0.9 \rangle$, with $\mathcal{A}_1 = \{ (p \leftarrow r, 0.9) \}$. However, $\mathcal{P}_1 \cup \{ (x \leftarrow , 1) \} \not\vdash_{\Delta} \langle \mathcal{A}_1, z, 0.9 \rangle$, and $\mathcal{P}_1 \cup \{ (\sim x \leftarrow , 1) \} \not\vdash_{\Delta} \langle \mathcal{A}_1, z, 0.9 \rangle$.
- RM: Consider the same counterexample as given for RN. Then $\mathcal{P}_1 \vdash_{\Delta} \langle \mathcal{A}_1, z, 0.9 \rangle$, but it is not the case that $\mathcal{P}_1 \cup \{ (x \leftarrow , 1) \} \vdash_{\Delta} \langle \mathcal{A}_1, z, 0.9 \rangle$ nor $\mathcal{P}_1 \vdash_{\Delta} (\sim x \leftarrow , 1)$.
- DR: Clearly, C_{Δ} does not satisfy property (e), as disjunctions cannot be expressed as wffs in the P-DeLP object language.

4.2 Logical properties for C_w

Next we will analyze some relevant logical properties for C_w . Notice that by definition C_w satisfies inclusion.

Proposition 7. *The operator C_w satisfies inclusion.*

Monotonicity does not hold, as can be seen from the counterexample used for monotonicity in C_{Δ} ; in that case, $(q, 0.9) \in C_w(\mathcal{P})$, but $(q, 0.9) \notin C_w(\mathcal{P} \cup \{(\sim p, 1)\})$. Semi-monotonicity does not hold either for C_w , as adding new defeasible clauses cannot invalidate already derivable arguments, but it can enable new ones that were not present before, thus modifying the dialectical relationships among arguments. Arguments that were warranted may therefore no longer keep that epistemic status. Consider a variant of the previous counterexample: let $\mathcal{P} = \{ (q, 1), (p \leftarrow q, 0.9) \}$. Then $(p, 0.9) \in C_w(\mathcal{P})$, as there is an argument $\langle \mathcal{A}, p, 0.9 \rangle$ on the basis of \mathcal{P} . However, $(p, 0.9) \notin C_w(\mathcal{P} \cup \{(\sim p, 0.95)\})$, as $\langle \mathcal{A}, p, 0.9 \rangle$ is defeated by $\langle \mathcal{B}, \sim p, 0.95 \rangle$, with $\mathcal{B} = \{(\sim p, 0.95)\}$. There are no more arguments to consider, and hence $\langle \mathcal{A}, p, 0.9 \rangle$ is not warranted. From our current analysis cumulativity and idempotence seem to hold for the C_w operator: we have not found any counterexample showing that these two properties do not hold, and we are currently studying the formulation of a proof. In comparison with C_{Δ} such a formal analysis is much more complex, as dialectical trees are not structures that can be recursively defined (notice that subtrees of dialectical trees are not dialectical trees). The reason for this is given by the different dialectical constraints that have to be taken into account (see previous discussion on acceptability in argumentation lines in Section 3).

Conjecture 1 *The operator C_w satisfies cumulativity i.e. $\mathcal{P}_1 \subseteq \mathcal{P}_2 \subseteq C_w(\mathcal{P}_1)$ implies $C_w(\mathcal{P}_1) = C_w(\mathcal{P}_2)$.*

Conjecture 2 *The operator C_w satisfies idempotence i.e. $C_w(\mathcal{P}) = C_w(C_w(\mathcal{P}))$.*

Property	C_{Δ}	C_w	Comments	Property	C_{Δ}	C_w	Comments
Inclusion	o	o	Prop. 2 and 7.	Conj. concl.	x	x	
Idempotence	o	o?	Prop. 2 & Conj. 2.	Subclass. cumm.	o	o	Prop. 6 and 9.
Cumulativity	o	o?	Prop. 4 & Conj. 1.	Left absorption	x	x	
Monotonicity	x	x		Right absorption	x	x	
Semi-monotonicity	o	x	Prop. 3 and 7.	Rational Negation	x	x	
(Horn) Supraclass.	o	o	Prop. 5 and 8	Disj. Rationality	x	x	
Left-logical equiv.	x	x		Rational Monoton.	x	x	
Horn Right Weak.	o	o	Prop. 5 and 8				

Fig. 1. Logical properties in P-DeLP: summary

Proposition 8. *The operator C_w satisfies (Horn) supraclassicality wrt C_{\perp} (i.e. $C_{\perp}(\mathcal{P}) \subseteq C_w(\mathcal{P})$), and (Horn) right weakening, (i.e. if $(Y, \alpha) \in C_w(\mathcal{P})$ and $(X \leftarrow Y, 1) \in C_{\perp}(\mathcal{P})$, then $(X, \alpha) \in C_w(\mathcal{P})$).*

Proof. (Sketch) Supraclassicality follows from Prop. 1. For the case of right weakening, let us suppose $(Y, \alpha) \in C_w(\mathcal{P})$, i.e. $\mathcal{P} \vdash_w \langle \mathcal{A}_1, Y, \alpha \rangle$, for some argument $\langle \mathcal{A}_1, Y, \alpha \rangle$. If $(X \leftarrow Y, 1) \in C_{\perp}(\mathcal{P})$, then necessarily $(X \leftarrow Y, 1) \in \mathcal{P}$ (by def. of C_{\perp}). By Prop. 5, if $\mathcal{P} \vdash_{\Delta} \langle \mathcal{A}_1, X, \alpha \rangle$. Clearly argument $\langle \mathcal{A}_1, X, \alpha \rangle$ and $\langle \mathcal{A}_1, Y, \alpha \rangle$ have the same set of associated defeaters. Hence if $\langle \mathcal{A}_1, Y, \alpha \rangle$ is warranted, then $\langle \mathcal{A}_1, X, \alpha \rangle$ also is.

Proposition 9. *The operator C_{Δ} satisfies subclassical cumulativity, i.e. $\mathcal{P}_1 \subseteq \mathcal{P}_2 \subseteq C_{\perp}(\mathcal{P}_1)$ implies $C_w(\mathcal{P}_1) = C_w(\mathcal{P}_2)$.*

Proof. Not included for space reasons.

As for C_{Δ} , most non-pure logical properties for C_w do not hold. In particular, C_w does not satisfy the properties of LL, CC, LA, RA, RN, RM and DR. In all cases this is based on the existence of counterexamples following the same line of reasoning as for C_{Δ} .

4.3 Discussion

Figure 1 summarizes the logical properties discussed before. When analyzing argumentative inference under the operator C_{Δ} , idempotence shows us that adding argument conclusions as new facts to a given program does not add any new inference capabilities. Cumulativity shows us that any argument obtained from a program \mathcal{P} can be kept as an intermediate proof (lemma) to be used in building more complex arguments. (Horn) supraclassicality indicates that every conclusion that follows via traditional SLD inference (involving only certain clauses) can be considered as a special form of argument (namely, an empty argument), whereas Horn right weakening tells us that certain rules in P-DeLP preserve the usual semantics for Horn rules (the existence of a certain rule $X \leftarrow Y$ causes that every argument concluding Y is also an argument for X). Computing warrant also can be better understood in the light of the logical properties

for C_w . From Horn supraclassicality it follows that every conclusion obtained from certain clauses is a particular case of warranted literal, whereas Horn right weakening indicates that non-defeasible rules behave as such in the meta-level (a strong rule $(Y \leftarrow X, 1)$ ensures that every warranted argument \mathcal{A} for (X, α) allow us to ensure that (Y, α) is also warranted. Cumulativity for C_w is specially interesting, as we will further discuss in the next Section.

5 Related work. Conclusions

Research in logical properties for defeasible argumentation can be traced back to Benferhat *et al.* [9, 10] and Vreeswijk [13]. 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.* [9] 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 \vdash_{Δ} relationship proposed in this paper takes into account *any* possible argument derivable from the program. In [9, 10] the authors also extend the argumentative relation $\vdash_{\mathcal{A}}$ to prioritized knowledge bases, assessing weights to conclusions on the basis of the \vdash_{π} -entailment relationship from possibilistic logic [7]. A direct comparison to our \vdash_w relationship is not easy since we are using a logic programming framework and not general propositional logic, but roughly speaking while \vdash_{π} takes into account the inconsistency degree associated with the whole knowledge base, our logic programming frame allows us to perform a dialectical analysis restricted only to conflicting arguments related with the query being solved. More recently there have been generic approaches connecting defeasible reasoning and possibilistic logic (e.g.[14]). Preference-based approaches to argumentation have been also developed, many of them oriented towards formalizing conflicting desires in multiagent systems [15, 16]. Part of our current work involves studying the formalization of expansion operators for such contexts.

In [11] 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 and idempotence we have conjectured that they hold in the context of P-DeLP. Should such conjectures be true, this would provide an interesting result in comparison with [11], as it would mean that warranted conclusions (Q, α) in a given program \mathcal{P} could be introduced as new uncertain facts, speeding up computation of new future queries. However, such facts would not have the same epistemic status as discussed in [11], where warranted conclusions are analyzed as certain facts for the study of cumulativity.

We have shown that P-DeLP provides a useful framework for making a formal analysis of logical properties in defeasible argumentation under uncertainty. We contend that a formal analysis of defeasible consequence is mandatory to get an in-depth understanding of the behavior of argumentation frameworks. Expansion

operators provide a natural tool for characterizing that behavior, as well as useful criteria when developing and implementing new argumentation frameworks or assessing their expressive power.

Acknowledgments: We want to thank anonymous reviewers for their useful comments. 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) and by CONICET (Argentina).

References

1. Chesñevar, C.I., Simari, G., Alsinet, T., Godo, L.: A Logic Programming Framework for Possibilistic Argumentation with Vague Knowledge. In: Proc. Intl. Conf. in Uncertainty in Artificial Intelligence (UAI 2004). Banff, Canada. (2004) 76–84
2. Alsinet, T., Godo, L.: A complete calculus for possibilistic logic programming with fuzzy propositional variables. In: Proc. of the UAI-2000 Conference. (2000) 1–10
3. Alsinet, T., Godo, L.: A proof procedure for possibilistic logic programming with fuzzy constants. In: Proc. of the ECSQARU-2001 Conference. (2001) 760–771
4. Chesñevar, C., Maguitman, A., Simari, G.: A first approach to argument-based recommender systems based on defeasible logic programming. In: Proc. 10th Intl. Workshop on Non-Monotonic Reasoning. Whistler, Canada. (2004) 109–117
5. Chesñevar, C., Maguitman, A.: An Argumentative Approach to Assessing Natural Language Usage based on the Web Corpus. In: Proc. of the ECAI-2004 Conference. Valencia, Spain. (2004) 581–585
6. Lifschitz, V.: Foundations of logic programming. In: Principles of Knowledge Representation. CSLI Publications (1996) 69–127
7. Dubois, D., Lang, J., Prade, H.: Possibilistic logic. In D.Gabbay, C.Hogger, J.Robinson, eds.: Handbook of Logic in Art. Int. and Logic Prog. (Nonmonotonic Reasoning and Uncertain Reasoning). Oxford Univ. Press (1994) 439–513
8. Makinson, D.: General patterns in nonmonotonic reasoning. In D.Gabbay, C.Hogger, J.Robinson, eds.: Handbook of Logic in Art. Int. and Logic Prog. Volume Nonmonotonic and Uncertain Reasoning. Oxford University Press (1994) 35–110
9. Benferhat, S., Dubois, D., Prade, H.: Argumentative inference in uncertain and inconsistent knowledge bases. In: Proc. of UAI. (1993) 411–419
10. Benferhat, S., Dubois, D., Prade, H.: Some syntactic approaches to the handling of inconsistent knowledge bases: A comparative study. part ii: The prioritized case. In Orłowska, E., ed.: Logic at work. Volume 24. Physica-Verlag, Heidelberg (1998) 473–511
11. Prakken, H., Vreeswijk, G.: Logical Systems for Defeasible Argumentation. In Gabbay, D., F.Guenther, eds.: Handbook of Phil. Logic. Kluwer (2002) 219–318
12. Chesñevar, C., Maguitman, A., Loui, R.: Logical Models of Argument. ACM Computing Surveys **32** (2000) 337–383
13. Vreeswijk, G.A.: Studies in Defeasible Argumentation. PhD thesis, Vrije University, Amsterdam (Holanda) (1993)
14. Benferhat, S., Dubois, D., Prade, H.: The possibilistic handling of irrelevance in exception-tolerant reasoning. Annals of Math. and AI **35** (2002) 29–61
15. Amgoud, L.: A formal framework for handling conflicting desires. In: Proc. of the ECSQARU-2003 Conference. (2003) 552–563
16. Amgoud, L., Cayrol, C.: Inferring from inconsistency in preference-based argumentation frameworks. Journal of Automated Reasoning **29** (2002) 125–169