A characterization of collective conflict for defeasible argumentation

Teresa ALSINET [a,1], Ramón BÉJAR [a] and Lluís GODO [b]

[a] *Department of Computer Science. University of Lleida,* SPAIN

[b] *Artificial Intelligence Research Institute (IIIA-CSIC), Bellaterra,* SPAIN

Abstract.
In this paper we define a recursive semantics for warrant in a general defeasible argumentation framework by formalizing a notion of collective (non-binary) conflict among arguments. This allows us to ensure direct and indirect consistency (in the sense of Caminada and Amgoud) without distinguishing between direct and indirect conflicts. Then, the general defeasible argumentation framework is extended by allowing to attach levels of preference to defeasible knowledge items and by providing a level-wise definition of warranted and blocked conclusions. Finally, we formalize the warrant recursive semantics for the particular framework of Possibilistic Defeasible Logic Programming, characterize the unique output program property and design an efficient algorithm for computing warranted conclusions in polynomial space.

Keywords. Defeasible argumentation, collective conflict, recursive warrant semantics.

1. Introduction and motivation

Possibilistic Defeasible Logic Programming (P-DeLP) [3] is a rule-based argumentation framework which is an extension of Defeasible Logic Programming (DeLP) [10] in which defeasible rules are attached with weights (belonging to the real unit interval $[0, 1]$) expressing their belief or preference strength and formalized as necessity degrees. As many other argumentation frameworks [8,14], P-DeLP can be used as a vehicle for facilitating rationally justifiable decision making when handling incomplete and potentially inconsistent information. Actually, given a P-DeLP program, justifiable decisions correspond to warranted conclusions (with a maximum necessity degree), that is, those which remain undefeated after an exhaustive dialectical analysis of all possible arguments for and against.

In [6] Caminada and Amgoud propose three *rationality postulates* which every rule-based argumentation system should satisfy. One of such postulates (called *Indirect Consistency*) claims that the closure of warranted conclusions with respect to the set of strict rules must be consistent. A number of rule-based argumentation systems are identified in which such postulate does not hold (including DeLP [10] and Prakken & Sartor's [13], among others). As a way to solve this problem, the use of *transposed rules* is proposed in [6] to extend the representation of strict rules.

Since the dialectical analysis based semantics of P-DeLP does not satisfy indirect consistency, in [2,1] a level-wise approach to compute warranted conclusions, called

[1] Correspondence to: T. Alsinet. Department of Computer Science, University of Lleida. C/Jaume II, 69. Lleida, Spain. Tel.: +34 973702734; Fax: +34 973702702; E-mail: tracy@diei.udl.cat

level-based P-DeLP, was defined ensuring the indirect consistency postulate without extending the representation of strict rules with transposed rules. In contrast with DeLP and other argument-based approaches [8,14,5,15], the level-based P-DeLP framework does not require the use of dialectical trees as underlying structures for characterizing the semantics for warranted conclusions. The level-based P-DeLP framework distinguishes two types of conflicts between arguments, direct and indirect. Direct conflicts occur when there exists an inconsistency emerging from arguments supporting contradictory literals. Indirect conflicts occur in a given program when there exists an inconsistency emerging from the set of strict rules of the program and a set of non-defeated (due to a direct conflict) arguments. The level-based P-DeLP framework therefore establishes an implicit evaluation order between conflicts, in the sense that if a conclusion is involved in both a direct and indirect conflict, the direct conflict invalidates the indirect one. On the other hand, although the level-based P-DeLP semantics for warranted conclusions is skeptical, in [1] it was shown that some circular definitions of conflict between arguments can arise and they can lead to different extensions of warranted conclusions.

Recently Pollock defined [12] a recursive semantics for defeasible argumentation (without levels of preference) where circular definitions of defeat between arguments were characterized by means of *inference-graphs*, representing (binary) support and defeat relations between the conclusions of arguments. Following this approach, our aim in this paper is to formally characterize circular definitions of conflict among arguments that cause different extensions of warranted conclusions in the level-based P-DeLP framework. However, because of the above mentioned implicit evaluation order between conflicts and its undesired side-effect, we are in need for a new and general notion of conflict among arguments which, besides of ensuring the Caminda and Amgoud's rationality postulates, allows us to safely reason about circular definitions of conflict between arguments.

To this end, in this paper we first define a recursive semantics for warranted conclusions in a quite general framework (without levels of strength) by formalizing a new collective (non-binary) notion of conflict between arguments ensuring indirect consistency without distinguishing between direct and indirect conflicts. Second we extend the recursive semantics to an argumentation framework with levels of preference by providing a level-wise definition of warranted and blocked conclusions. A warranted conclusion is a justified conclusion which is only based on warranted information and which does not generate a conflict, while a blocked conclusion is a conclusion which, like warranted conclusions, is only based on warranted information, but it does generate a conflict. Third, we specialize the warrant recursive semantics for the particular framework of P-DeLP, we refer to this formalism as RP-DeLP, characterize the condition under which a program has a unique output based on what we call *warrant dependency graph*, and design an efficient algorithm for computing warranted conclusions in polynomial space.

2. General defeasible argumentation framework

We will start by considering a rather general framework for defeasible argumentation based on a propositional logic $(\mathcal{L}, \vdash)$ with a special symbol $\perp$ for contradiction[2]. For any set of formulas A, if $A \vdash \perp$ we will say that A is contradictory, while if $A \not\vdash \perp$ we

[2] If not stated otherwise, in this and in the next section $(\mathcal{L}, \vdash)$ may be taken as classical propositional logic.

will say that A is consistent. A knowledge base (KB) is a triplet $\mathcal{P} = (\Pi, \Delta, \Sigma)$, where $\Pi, \Delta, \Sigma \subseteq \mathcal{L}$, and $\Pi \not\vdash \perp$. Π is a finite set of formulas representing strict knowledge (formulas we take for granted they hold to be true), Δ is another finite set of formulas representing the defeasible knowledge (formulas for which we have reasons to believe they are true) and Σ denotes the set of formulas over which arguments can be built. In many argumentation systems, Σ is taken to be a set of literals.

The notion of *argument* is the usual one. Given a KB $\mathcal{P}$, an argument for a formula $\varphi \in \Sigma$ is a pair $\mathcal{A} = \langle A, \varphi \rangle$, with $A \subseteq \Delta$ such that:

1. $\Pi \cup A \not\vdash \perp$, and
2. A is minimal (w.r.t. set inclusion) such that $\Pi \cup A \vdash \varphi$.

If $A = \emptyset$, then we will call $\mathcal{A}$ a s-argument (s for strict), otherwise it will be a d-argument (d for defeasible). The notion of *subargument* is referred to d-arguments and expresses an incremental prove relationship between arguments which is formalized as follows.

Definition 1 (Subargument) *Let $\langle B, \psi \rangle$ and $\langle A, \varphi \rangle$ be two d-arguments such that the minimal sets (w.r.t. set inclusion) $\Pi_\psi \subseteq \Pi$ and $\Pi_\varphi \subseteq \Pi$ such that $\Pi_\psi \cup B \vdash \psi$ and $\Pi_\varphi \cup A \vdash \varphi$ verify that $\Pi_\psi \subseteq \Pi_\varphi$. Then, $\langle B, \psi \rangle$ is a subargument of $\langle A, \varphi \rangle$, written $\langle B, \psi \rangle \sqsubset \langle A, \varphi \rangle$, when either $B \subset A$ (strict inclusion for defeasible knowledge), or $B = A$ and $\Pi_\psi \subset \Pi_\varphi$ (strict inclusion for strict knowledge), or $B = A$ and $\Pi_\psi = \Pi_\varphi$ and $\psi \vdash \varphi$ and $\varphi \not\vdash \psi$* [3].

A formula $\varphi \in \Sigma$ will be called *justifiable* w.r.t. $\mathcal{P}$ if there exists an argument for φ, i.e. there exists $A \subseteq \Delta$ such that $\langle A, \varphi \rangle$ is an argument.

The usual notion of attack or defeat relation in an argumentation system is binary. However in certain situations, the conflict relation among arguments is hardly representable as a binary relation. For instance, consider the following KB $\mathcal{P}_1 = (\Pi, \Delta, \Sigma)$ with

$$\Pi = \{a \wedge b \rightarrow \neg p\},\ \Delta = \{a, b, p\} \text{ and } \Sigma = \{a, b, p, \neg p\}.$$

Clearly, $\mathcal{A}_1 = \langle \{p\}, p \rangle, \mathcal{A}_2 = \langle \{b\}, b \rangle, \mathcal{A}_3 = \langle \{a\}, a \rangle$ are arguments that justify p, b and a respectively, and which do not pair-wisely generate a conflict. Indeed, $\Pi \cup \{a, b\} \not\vdash \perp$, $\Pi \cup \{a, p\} \not\vdash \perp$ and $\Pi \cup \{b, p\} \not\vdash \perp$. However the three arguments are collectively conflicting since $\Pi \cup \{a, b, p\} \vdash \perp$, hence in this $\mathcal{P}_1$ there is a non-binary conflict relation among several arguments. In the following we will formalize this notion of collective, or non-binary, conflict among in principle valid arguments and which arises when we compare them with the strict part of the knowledge base.

The following notion of acceptable argument with respect to a set (possibly empty) of justifiable conclusions W will play a key role. If we think of W as a consistent set of already warranted conclusions, an acceptable argument captures the idea of an argument which is based on subarguments already warranted.

Definition 2 (Acceptable argument) *Let W be a set of justifiable conclusions which is consistent w.r.t. Π, i.e. $\Pi \cup W \not\vdash \perp$. A d-argument $\mathcal{A} = \langle A, \varphi \rangle$ is an acceptable argument for φ w.r.t. W iff:*

[3] Notice that if $(\Pi, \Delta, \Sigma) = (\{r\}, \{r \rightarrow p \wedge q\}, \{p, q, p \wedge q\})$ and $A = \{r \rightarrow p \wedge q\}$ then $\mathcal{A}_1 = \langle A, p \rangle$, $\mathcal{A}_2 = \langle A, q \rangle$ and $\mathcal{A}_3 = \langle A, p \wedge q \rangle$ are arguments for different formulas with a same support and thus, in our framework, $\mathcal{A}_3 \sqsubset \mathcal{A}_1$ and $\mathcal{A}_3 \sqsubset \mathcal{A}_2$ are the subargument relations between arguments $\mathcal{A}_1$, $\mathcal{A}_2$ and $\mathcal{A}_3$ since $p \wedge q \vdash p$, $p \wedge q \vdash q$, $p \not\vdash p \wedge q$ and $q \not\vdash p \wedge q$.

1. *if* $\langle B, \psi\rangle$ *is a subargument of* $\langle A, \varphi\rangle$ *then* $\psi \in W$
2. $\Pi \cup W \cup \{\varphi\} \nvdash \perp$

In the above example, arguments $\mathcal{A}_1$, $\mathcal{A}_2$ and $\mathcal{A}_3$ are acceptable w.r.t. Π and the empty set of conclusions $W = \emptyset$. However $\mathcal{A}_4 = \langle\{a, b\}, \neg p\rangle$ is an argument for $\neg p$, but $\mathcal{A}_4$ is not acceptable w.r.t. $W = \emptyset$ since $\mathcal{A}_2$ and $\mathcal{A}_3$ are subarguments of $\mathcal{A}_4$ but obviously $a, b \notin W$.

Now we are ready to introduce the notion of collective conflict relative to a consistent set of justifiable conclusions. The idea of defining a warrant semantics on the basis of conflicting sets of arguments was proposed in [16] and [11]. The difference between these approaches and our notion of collective conflict is that in [16] the notion of conflict is not relative to a set of already warranted conclusions and [11] defines a generalization of Dung's abstract framework with sets of attacking arguments not relative to the strict part of the knowledge base.

Definition 3 (Conflict among arguments) *Let* $\mathcal{P} = (\Pi, \Delta, \Sigma)$ *be a KB, let* W *be a consistent set of justifiable conclusions w.r.t.* Π *and let* $\mathcal{A}_1 = \langle A_1, \varphi_1\rangle, \ldots, \mathcal{A}_k = \langle A_k, \varphi_k\rangle$ *be acceptable arguments w.r.t.* W*. We say that the set of arguments* $\{\mathcal{A}_1, \ldots, \mathcal{A}_k\}$ *generates a conflict w.r.t.* W *iff the two following conditions hold:*

(C) The set of argument conclusions $\{\varphi_1, \ldots, \varphi_k\}$ *is contradictory w.r.t.* $\Pi \cup W$, *i.e.* $\Pi \cup W \cup \{\varphi_1, \ldots, \varphi_k\} \vdash \perp$.

(M) The set $\{\mathcal{A}_1, \ldots, \mathcal{A}_k\}$ *is minimal w.r.t. set inclusion satisfying (C), i.e. if* $S \subset \{\varphi_1, \ldots, \varphi_k\}$*, then* $\Pi \cup W \cup S \nvdash \perp$.

Consider the previous KB $\mathcal{P}_1$. According to Definition 3, it is clear that the set of acceptable arguments $\{\mathcal{A}_1, \mathcal{A}_2, \mathcal{A}_3\}$ for p, b and a respectively generates a (collective) conflict (w.r.t. $W = \emptyset$). The intuition is that this collective conflict should block the conclusions a, b and p to be warranted. Now, this general notion of conflict is used to define a recursive semantics for warranted conclusions of a knowledge base. Actually we define below an output of a KB $\mathcal{P} = (\Pi, \Delta, \Sigma)$ as a pair $(Warr, Block)$ of subsets of Σ of warranted and blocked conclusions respectively, all of them based on warranted information but, while warranted conclusions do not generate any conflict, blocked conclusions do.

Definition 4 (Output for a KB) *An output for a KB* $\mathcal{P} = (\Pi, \Delta, \Sigma)$ *is any pair (Warr, Block), where Warr = s-Warr* $\cup$ *d-Warr with s-Warr* $= \{\varphi \mid \Pi \vdash \varphi\} \cap \Sigma$*, and d-Warr and Block are required to satisfy the following recursive constraints:*

1. *A d-argument* $\langle A, \varphi\rangle$ *is called* valid *(or not rejected) if it satisfies the following two conditions:*

   (i) for every $\langle B, \psi\rangle \sqsubset \langle A, \varphi\rangle$, $\psi \in$ *d-Warr,*

   (ii) $\langle A, \varphi\rangle$ *is acceptable w.r.t. the set* $W = \{\psi \mid \langle B, \psi\rangle \sqsubset \langle A, \varphi\rangle\}$.

2. *For every valid argument* $\langle A, \varphi\rangle$ *we have that*
   - $\varphi \in$ *d-Warr whenever there does not exist a set of valid arguments G such that*

     (i) $\langle A, \varphi\rangle \not\sqsubset \langle C, \chi\rangle$ *for all* $\langle C, \chi\rangle \in G$

     (ii) $G \cup \{\langle A, \varphi\rangle\}$ *generates a conflict w.r.t.* $W = \{\psi \mid$ *there exists* $\langle B, \psi\rangle \sqsubset \langle D, \gamma\rangle$ *for some* $\langle D, \gamma\rangle \in G \cup \{\langle A, \varphi\rangle\}\}$

- *otherwise, $\varphi \in$ Block.*

The intuition underlying this definition is as follows: an argument $\langle A, \varphi \rangle$ is either warranted or blocked whenever for each subargument $\langle B, \psi \rangle$ of $\langle A, \varphi \rangle$, ψ is warranted; then, it is eventually warranted if φ is not involved in any conflict, otherwise it is blocked.

Example 5 *Consider the KB $\mathcal{P}_2 = (\Pi, \Delta, \Sigma)$, with*

$$\Pi = \{a \to y, b \wedge c \to \neg y\},\ \Delta = \{a, b, c, \neg c\} \text{ and } \Sigma = \{a, b, c, \neg c, y, \neg y\}.$$

According to Definition 4, s-Warr $= \emptyset$ and the arguments $\langle \{a\}, a \rangle$, $\langle \{b\}, b \rangle$, $\langle \{c\}, c \rangle$ and $\langle \{\neg c\}, \neg c \rangle$ are valid. Now, for every such valid argument there exists a set of valid arguments which generates a conflict w.r.t. $W = \emptyset$: indeed both sets of valid arguments $\{\langle \{a\}, a \rangle, \langle \{b\}, b \rangle, \langle \{c\}, c \rangle\}$ and $\{\langle \{c\}, c \rangle, \langle \{\neg c\}, \neg c \rangle\}$ generate a conflict (since $\Pi \cup \{a, b, c\} \vdash \bot$ and $\Pi \cup \{c, \neg c\} \vdash \bot$). Therefore a, b, c and $\neg c$ are blocked conclusions. On the other hand, the arguments $\langle \{a, b\}, \neg c \rangle$, $\langle \{a\}, y \rangle$ and $\langle \{b, c\}, \neg y \rangle$ are not valid since they are based on conclusions which are not warranted. Hence y and $\neg y$ are considered as rejected conclusions. Thus, the (unique) output for $\mathcal{P}$ is the pair $(Warr, Block) = (\emptyset, \Delta)$. Intuitively this output for $\mathcal{P}$ expresses that all conclusions in Block are valid, however all together are contradictory w.r.t. Π.

The KB $\mathcal{P}_2$ was also considered in [2], where direct conflicts were evaluated before indirect conflicts, and thus, every blocked literal invalidated all rules in which that literal occurred. Hence, in [2], c and $\neg c$ were considered as blocked conclusions but a, b and y were warranted conclusions.

Next we prove that if (*Warr*, *Block*) is an output for a KB the set *Warr* of warranted conclusions satisfies indirect consistency and the closure postulate (in the sense of Caminada and Amgoud) with respect to the strict knowledge.

Proposition 6 (Indirect consistency) *Let $\mathcal{P} = (\Pi, \Delta, \Sigma)$ be a KB and let (Warr, Block) be an output for $\mathcal{P}$. Then, $\Pi \cup$ Warr $\nvdash \bot$.*

Proof: Suppose that $\Pi \cup$ *Warr* $\vdash \bot$. Obviously, it should be that $\Pi \cup W \vdash \bot$ for some $W \subseteq$ *d-Warr*. However, by Definition 4, for every $\varphi \in$ *d-Warr* there does not exist a set $W' \subseteq$ *d-Warr* such that $\Pi \cup W' \cup \{\varphi\} \vdash \bot$, and therefore, $\Pi \cup W \nvdash \bot$ for all $W \subseteq$ *d-Warr*. □

Proposition 7 (Closure) *Let $\mathcal{P} = (\Pi, \Delta, \Sigma)$ be a KB and let (Warr, Block) be an output for $\mathcal{P}$. If $\Pi \cup$ Warr $\vdash \varphi$ with $\varphi \in \Sigma$, then $\varphi \in$ Warr whenever there exits an acceptable argument for φ w.r.t. Warr.*

Proof: Suppose that for some $W \subseteq$ *Warr*, $\Pi \cup W \vdash \varphi$ and $\varphi \notin$ *Warr*, for some formula $\varphi \in \Sigma$ such that there exists an argument $\langle A, \varphi \rangle$ satisfying that if $\langle B, \psi \rangle$ is a subargument of $\langle A, \varphi \rangle$ then $\psi \in W$. On the one hand, if $A = \emptyset$, it must be that $\Pi \vdash \varphi$. Then, as $\varphi \in \Sigma$, $\varphi \in$ *Warr*. On the other hand, as $\Pi \cup$ *Warr* $\nvdash \bot$ and $\Pi \cup W \vdash \varphi$, $\Pi \cup W \cup \{\varphi\} \nvdash \bot$, and thus, $\langle A, \varphi \rangle$ is a valid argument for φ. Then, if $\varphi \notin$ *Warr*, there exits a set of valid arguments G such that (i) $\langle A, \varphi \rangle \not\sqsubseteq \langle C, \chi \rangle$ for all $\langle C, \chi \rangle \in G$, and (ii) $G \cup \{\langle A, \varphi \rangle\}$ generates a conflict w.r.t. $W' = \{\psi \mid$ there exists $\langle B, \psi \rangle \sqsubset \langle D, \gamma \rangle$ for some $\langle D, \gamma \rangle \in G \cup \{\langle A, \varphi \rangle\}\}$. According to Definition 3, if $G \cup \{\langle A, \varphi \rangle\}$ generates a conflict w.r.t. W', then $\Pi \cup W' \cup \{\varphi\} \cup \{\psi \mid \langle B, \psi \rangle \in G\} \vdash \bot$ (condition (C)), and $\Pi \cup W' \cup S \not\vdash \bot$, for all set $S \subset \{\varphi\} \cup \{\psi \mid \langle B, \psi \rangle \in G\}$ (condition (M)). Now, as $W \subseteq W'$ and $\Pi \cup W \vdash \varphi$, if $\Pi \cup W' \cup \{\varphi\} \cup \{\psi \mid \langle B, \psi \rangle \in G\} \vdash \bot$, then $\Pi \cup W' \cup \{\psi \mid \langle B, \psi \rangle \in G\} \vdash \bot$, and thus, $\varphi \in$ *Warr*. □

We remark that, as it will be discussed in Section 4, a KB may have multiple outputs. For instance, consider the KB $\mathcal{P}_3 = (\Pi, \Delta, \Sigma)$ with $\Pi = \emptyset$, $\Delta = \{p, q, \neg p \vee \neg q\}$ and $\Sigma = \{p, q, \neg p, \neg q\}$. Then, one can check that there are two outputs, $\mathit{Warr}_1 = \{p\}, \mathit{Block}_1 = \{q, \neg q\}$, and $\mathit{Warr}_2 = \{q\}, \mathit{Block}_2 = \{p, \neg p\}$.

3. Extending the framework with a preference ordering on arguments

In the previous section, we have considered knowledge bases containing formulas describing knowledge at two epistemic levels, strict and defeasible. A natural extension is to introduce several levels of defeasibility or preference among different pieces of defeasible knowledge.

A *stratified knowledge base* (sKB) is a tuple $\mathcal{P} = (\Pi, \Delta, \preceq, \Sigma)$, such that (Π, Δ, Σ) is a KB (in the sense of the previous section) and $\preceq$ is a *suitable* total pre-order on the set of defeasible formulas Δ. Suitable means for us that this pre-order is representable by a necessity measure defined on the set of formulas of $\mathcal{L}$, namely $\varphi \preceq \psi$ iff $N(\varphi) \leq N(\psi)$ for each $\varphi, \psi \in \Delta \cup \Pi$, where N is a mapping $N : \mathcal{L} \to [0, 1]$ such that

1. $N(\top) = 1, N(\bot) = 0$,
2. $N(\varphi \wedge \psi) = \min(N(\varphi), N(\psi))$, and further
3. $N(\varphi) = 1$ iff $\Pi \vdash \varphi$

Then we define the *strength of an argument* $\langle A, \varphi \rangle$, written $s(\langle A, \varphi \rangle)$, as follows[4]:

$$s(\langle A, \varphi \rangle) = 1 \text{ if } A = \emptyset, \text{ and } s(\langle A, \varphi \rangle) = \min\{N(\psi) \mid \psi \in A\}, \text{ otherwise.}$$

Since we are considering several levels of strength among arguments, the intended construction of the sets of conclusions *Warr* and *Block* is done level-wise, starting from the highest level and iteratively going down from one level to next level below. If $1 > \alpha_1 > \ldots > \alpha_p \geq 0$ are the strengths of d-arguments that can be built within a sKB $\mathcal{P} = (\Pi, \Delta, \preceq, \Sigma)$, we define $\textit{d-Warr} = \{\textit{d-Warr}(\alpha_1), \ldots, \textit{d-Warr}(\alpha_p)\}$ and $\mathit{Block} = \{\mathit{Block}(\alpha_1), \ldots, \mathit{Block}(\alpha_p)\}$, where *d-Warr*$(\alpha_i)$ and *Block*(α_i) are respectively the sets of warranted and blocked justifiable conclusions with strength α_i. Then, we safely write *d-Warrr*$(> \alpha_i)$ to denote $\cup_{\beta > \alpha_i}$*d-Warr*(β), and analogously for *Block*$(> \alpha_i)$, defininig *d-Warr*$(> \alpha_1) =$ *s-Warr* and *Block*$(> \alpha_1) = \emptyset$.

Definition 8 (Output for a sKB) *An output for a sKB $\mathcal{P} = (\Pi, \Delta, \preceq, \Sigma)$ is any pair (Warr, Block), where Warr = s-Warr $\cup$ d-Warr with s-Warr $= \{\varphi \mid \Pi \vdash \varphi\} \cap \Sigma$, and d-Warr and Block are required to satisfy the following recursive constraints:*

1. *A d-argument $\langle A, \varphi \rangle$ of strength α_i is called* valid *(or not rejected) if it satisfies the following three conditions*[5]*:*

[4] Actually, several N's may lead to a same pre-order $\preceq$, but we can take any of them to define the degree of strength since only the relative ordering is what matters.

[5] Notice that if $\langle A, \varphi \rangle$ is an acceptable argument w.r.t. *d-Warr*$(> \alpha_i)$, then $\langle A, \varphi \rangle$ is valid whenever condition (iii) holds.

(i) for every subargument $\langle B, \psi\rangle \sqsubset \langle A, \varphi\rangle$ of strength α_i, $\psi \in$ d-Warr(α_i);
(ii) $\langle A, \varphi\rangle$ is acceptable w.r.t.
$W =$ d-Warr($> \alpha_i$) $\cup \{\psi \mid \langle B, \psi\rangle \sqsubset \langle A, \varphi\rangle$ and $s(\langle B, \psi\rangle) = \alpha_i\}$;
(iii) $\varphi \notin$ d-Warr($> \alpha_i$)$\cup$Block($> \alpha_i$) and $\{\varphi, \psi\} \not\vdash \bot$ for all $\psi \in$ Block($> \alpha_i$).

2. *For every valid argument $\langle A, \varphi\rangle$ of strength α_i we have that*
   - *$\varphi \in$ d-Warr(α_i) whenever there does not exist a set G of valid arguments of strength α_i such that*

     (i) $\langle A, \varphi\rangle \not\sqsubset \langle C, \chi\rangle$ for all $\langle C, \chi\rangle \in G$
     (ii) $G \cup \{\langle A, \varphi\rangle\}$ generates a conflict w.r.t. $W =$ d-Warr($> \alpha_i$) $\cup \{\psi \mid$ there exists $\langle B, \psi\rangle \sqsubset \langle D, \gamma\rangle$ for some $\langle D, \gamma\rangle \in G \cup \{\langle A, \varphi\rangle\}\}$

   - *otherwise, $\varphi \in$ Block(α_i).*

There are two main remarks when considering several levels of strength among arguments. On the one hand a d-argument $\langle A, \varphi\rangle$ of strength α_i is valid whenever there does not exist a different valid argument for φ of strength greater than α_i and φ is consistent with each valid argument of strength greater than α_i. On the other hand, a valid argument $\langle A, \varphi\rangle$ of strength α_i becomes blocked as soon as it leads to some conflict among arguments of strength α_i.

Example 9 *Consider the KB $\mathcal{P}_1$ in the previous section*

$$\Pi = \{a \wedge b \rightarrow \neg p\},\ \Delta = \{a, b, p\} \text{ and } \Sigma = \{a, b, p, \neg p\}.$$

extended with levels of defeasibility as follows: $\{a, b\} \prec p$. Assume α_1 is the level of p and α_2 the level of a and b, obviously with $1 > \alpha_1 > \alpha_2$. According to Definition 8, s-Warr $= \emptyset$ and the argument for $\langle\{p\}, p\rangle$ is the only valid argument with strength α_1. Then, at level α_1, we get d-Warr(α_1) $= \{p\}$ and Block(α_1) $= \emptyset$. At level α_2, we have that $\langle\{a\}, a\rangle$ and $\langle\{b\}, b\rangle$ are valid arguments for conclusions a and b respectively. However, since $\Pi \cup$ d-Warr(α_1) $\cup \{a, b\} \vdash \bot$, the conclusions a and b are blocked, and thus, d-Warr(α_2) $= \emptyset$ and Block(α_2) $= \{a, b\}$. Notice that the argument $\langle\{a, b\}, \neg p\rangle$ for $\neg p$ is not acceptable since it is based on a and b and $a, b \notin$ d-Warr(α_2).

Example 10 *Consider the KB $\mathcal{P}_2$ of Example 5:*

$$\Pi = \{a \rightarrow y, b \wedge c \rightarrow \neg y\},\ \Delta = \{a, b, c, \neg c\}, \text{ and } \Sigma = \{a, b, c, \neg c, y, \neg y\},$$

extended with three levels of defeasibility as follows: $\neg c \prec c \prec \{a, b\}$. Assume α_1 is the level of a and b, α_2 is the level of c, and α_3 is the level of $\neg c$, with $1 > \alpha_1 > \alpha_2 > \alpha_3$. Then, s-Warr $= \emptyset$ and, at level α_1, we have not only the conclusions a, b and y with valid arguments not generating conflict, but also $\langle\{a, b\}, \neg c\rangle$ is a valid argument for $\neg c$ which does not generate conflict. Therefore, d-Warr(α_1) $= \{a, b, y, \neg c\}$ and Block(α_1) $= \emptyset$. At level α_2, we have arguments for c and $\neg y$. Since $\Pi \cup$d-Warr(α_1)$\cup$ $\{c\} \vdash \bot$, the argument $\langle\{c\}, c\rangle$ is not acceptable w.r.t. d-Warr(α_1), and thus, c is a rejected conclusion. Then, as the argument $\langle\{b, c\}, \neg y\rangle$ for $\neg y$ is based on c, $\neg y$ is also a rejected conclusion, and therefore d-Warr(α_2) $= Block(\alpha_2) = \emptyset$. Finally, at level α_3 we have the argument $\langle\{\neg c\}, \neg c\rangle$, but since $\neg c$ is already in d-Warr(α_1), we also have d-Warr(α_3) $= Block(\alpha_3) = \emptyset$.

4. A particular case: recursive P-DeLP

In this section we particularize the framework and recursive warrant semantics for stratified knowledge bases defined in the previous section to the case of the P-DeLP programs. As mentioned in Section 1, P-DeLP is a rule-based argumentation system extending the well-known DeLP system in which weights are attached to defeasible rules expressing their belief or preference strength and formalized as necessity degrees. For a detailed description of the P-DeLP argumentation system based on dialectical trees the reader is referred to [3].

Although the original syntax and inference of P-DeLP are a bit different (e.g. the weights are explicit in the formulas and arguments), here we will present them in a way so to adapt them to the framework introduced in the previous sections. We will refer to this particular framework as RP-DeLP. Hence we define the logic $(\mathcal{L}_R, \vdash_R)$ underlying RP-DeLP as follows. The language of RP-DeLP is inherited from the language of logic programming, including the notions of atom, literal, rule and fact. Formulas are built over a finite set of propositional variables $p, q, ...$ which is extended with a new (negated) atom "$\sim p$" for each original atom p. Atoms of the form p or $\sim p$ will be referred as literals, and if P is a literal, we will use $\sim P$ to denote $\sim p$ if P is an atom p, and will denote p if P is a negated atom $\sim p$. Formulas of $\mathcal{L}_R$ consist of rules of the form $Q \leftarrow P_1 \wedge \ldots \wedge P_k$, where $Q, P_1, \ldots, P_k$ are literals. A fact will be a rule with no premises. We will also use the name *clause* to denote a rule or a fact. The inference operator $\vdash_R$ is defined by instances of the modus ponens rule of the form: $\{Q \leftarrow P_1 \wedge \ldots \wedge P_k, P_1, \ldots, P_k\} \vdash_R Q$. A set of clauses Γ is *contradictory*, denoted $\Gamma \vdash \perp$, if , for some atom q, $\Gamma \vdash_R q$ and $\Gamma \vdash_R \sim q$.

A RP-DeLP program $\mathcal{P}$ is just a stratified knowledge base $(\Pi, \Delta, \preceq, \Sigma)$ over the logic $(\mathcal{L}_R, \vdash_R)$, where Σ consists of the set of all literals of $\mathcal{L}_R$. We will assume that $\preceq$ is representable by a necessity measure N, so we will often refer to numerical weights for defeasible clauses and arguments rather than to the pre-ordering $\preceq$. Also, for the sake of a simpler notation we will get rid of Σ from a program specification.

As we have mentioned in the previous section, in some cases the output (*Warr*, *Block*) for a stratified knowledge base in general, and for a RP-DeLP program in particular, is not unique, due to circular definitions of warranty that emerge when considering conflicts among arguments. Such circular definitions of warranty are characterized next by means of what we call *warrant dependency graph* of a RP-DeLP program. In [12] a similar graph structure, called inference-graph, was defined to represent inference (support) and defeat relations among arguments allowing to detect circular defeat relations when considering recursive semantics for defeasible reasoning. The main difference between both approaches is that in our case we handle collective conflicts among arguments in order to preserve direct consistency among warranted conclusions and indirect consistency with respect to the strict knowledge .

In the following, given a RP-DeLP program $\mathcal{P} = (\Pi, \Delta)$ with preference levels $1 > \alpha_1 > \ldots > \alpha_m > 0$, if W denotes a set of justifiable literals, we will denote by $W(\alpha)$ the subset of literals Q from W for which there exist an argument $\langle A, Q \rangle$ with maximum strength α, $W(\geq \alpha) = \cup_{\beta \geq \alpha} W(\beta)$, and $W(> \alpha) = \cup_{\beta > \alpha} W(\beta)$ with $W(> \alpha_1) = W(1)$.

Definition 11 (Warrant dependency graph) *Let $\mathcal{P} = (\Pi, \Delta)$ be a RP-DeLP program and let W be a set of justifiable conclusions consistent with Π. Let $\mathcal{A}_1 =$*

$\langle A_1, Q_1\rangle, \ldots, \mathcal{A}_k = \langle A_k, Q_k\rangle$ be acceptable arguments of a same strength α w.r.t. W such that for all i, $Q_i \notin W(\geq \alpha)$. Moreover, let $\mathcal{B}_1 = \langle B_1, P_1\rangle, \ldots, \mathcal{B}_n = \langle B_n, P_n\rangle$ be arguments of the same strength α such that for all j, $P_j \notin W(\geq \alpha)$, $P_j \notin \{Q_1, \ldots, Q_k\}$, and there exists an argument $\mathcal{S} \in \{\mathcal{A}_1, \ldots, \mathcal{A}_k\}$ with $\mathcal{S} \sqsubset \mathcal{B}_j$. Then, the warrant dependency graph (V, E) for $\{\mathcal{A}_1, \ldots, \mathcal{A}_k\}$ w.r.t. W and $\{\mathcal{B}_1, \ldots, \mathcal{B}_n\}$ is defined as follows:

1. *For every literal $L \in \{Q_1, \ldots, Q_k\} \cup \{P_1, \ldots, P_n\}$, the set of vertices V includes one vertex v_L.*
2. *For every pair of literals $\{L_1, L_2\}$ such that $L_1 = \sim L_2$ with $L_1 \in \{P_1, \ldots, P_n\}$ and $L_2 \in \{Q_1, \ldots, Q_k\}$, the set of directed edges E includes one edge (v_{L_1}, v_{L_2})*[6].
3. *For every pair of literals $\{L_1, L_2\}$ such that $L_1 \in \{Q_1, \ldots, Q_k\}$, $L_2 \in \{P_1, \ldots, P_n\}$ and the argument of L_1 is a subargument of the argument of L_2, the set of directed edges E includes one edge (v_{L_1}, v_{L_2})*[7].
4. *For every strict rule $L \leftarrow L_1 \wedge \ldots \wedge L_m$ of Π such that*

   - *either $\sim L \in W(\geq \alpha)$ or $\sim L \in \{Q_1, \ldots, Q_k\}$, and*
   - *for every L_i $(i = 1, ..., m)$, either $L_i \in W(\geq \alpha)$ or $L_i \in \{Q_1, \ldots, Q_k\} \cup \{P_1, \ldots, P_n\}$,*

   the set of directed edges E includes one edge (v_{L_i}, v_{L_j})[8] *for every pair of literals $\{L_i, L_j\} \subseteq \{L_1, \ldots, L_m\}$ with $L_i \in \{P_1, \ldots, P_n\}$ and $L_j \in \{Q_1, \ldots, Q_k\}$ whenever the argument of L_j is not a subargument of the argument of L_i.*
5. *Elements of V and E are only obtained by applying the above construction rules.*

Intuitively, the warrant dependency graph for a set of arguments represents conflict and support dependences among arguments in $\{\mathcal{A}_1, \ldots, \mathcal{A}_k\}$ and arguments in $\{\mathcal{B}_1, \ldots, \mathcal{B}_n\}$ w.r.t. a set of justified conclusions W.

Example 12 *Consider a RP-DeLP program defined from the KB $\mathcal{P}_3$ of Section 2; i.e. a RP-DeLP program with an empty set of strict clauses and the following set of defeasible clauses with just one defeasibility level:*

$$\Delta = \{p, q, \sim p \leftarrow q, \sim q \leftarrow p\}.$$

Now, consider the empty set of conclusions $W = W(1) = \emptyset$ and arguments for conclusions p and q; i.e. $\mathcal{A}_1 = \langle\{p\}, p\rangle$ and $\mathcal{A}_2 = \langle\{q\}, q\rangle$. Finally, consider the arguments for conclusions $\sim p$ and $\sim q$; i.e. $\mathcal{B}_1 = \langle\{q,\ \sim p \leftarrow q\}, \sim p\rangle$ and $\mathcal{B}_2 = \langle\{p,\ \sim q \leftarrow p\}, \sim q\rangle$.

Figure 1 (a) shows the warrant dependency graph for $\mathcal{A}_1$ and $\mathcal{A}_2$ w.r.t. $W = \emptyset$, $\mathcal{B}_1$, and $\mathcal{B}_2$. Conflict and support dependences between literals are represented as dashed and solid arrows, respectively. The cycle of the graph expresses that (1) the warranty of p depends on a (possible) conflict with $\sim p$; (2) the support of $\sim p$ depends on q (i.e. the validity of $\sim p$ depends on the warranty of q); (3) the warranty of q depends on a (possible) conflict with $\sim q$; and (4) the support of $\sim q$ depends on p (i.e. the validity of $\sim q$ depends on the warranty of p).

Consider now the RP-DeLP program $\mathcal{P}_4 = (\Pi, \Delta)$, with

$$\Pi = \{y, \sim y \leftarrow p \wedge r, \sim y \leftarrow q \wedge s\} \text{ and } \Delta = \{p, q, r \leftarrow q, s \leftarrow p\}$$

with just one defeasibility level. Moreover consider the set of justified conclusions $W =$

[6] The directed edge (v_{L_1}, v_{L_2}) represents a conflict dependence of L_2 w.r.t. L_1.

[7] The directed edge (v_{L_1}, v_{L_2}) represents a support dependence of L_2 w.r.t. L_1.

[8] The directed edge (v_{L_i}, v_{L_j}) represents a conflict dependence of L_j w.r.t. L_i.

$W(1) = \{y\}$ and arguments for conclusions p and q; i.e.

$$\mathcal{A}_1 = \langle \{p\}, p \rangle \text{ and } \mathcal{A}_2 = \langle \{q\}, q \rangle.$$

Finally, consider arguments for conclusions r and s; i.e.

$$\mathcal{B}_1 = \langle \{q, r \leftarrow q\}, r \rangle \text{ and } \mathcal{B}_2 = \langle \{p, s \leftarrow p\}, s \rangle.$$

Figure 1 (b) shows the warrant dependency graph for $\mathcal{A}_1$ and $\mathcal{A}_2$ w.r.t. W, $\mathcal{B}_1$, and $\mathcal{B}_2$. The cycle of the graph expresses that (1) the warranty of p depends on a (possible) conflict with r; (2) the support of r depends on q (i.e. the validity of r depends on the warranty of q); (3) the warranty of q depends on a (possible) conflict with s; and (4) the support of s depends on p (i.e. the validity of s depends on the warranty of p).

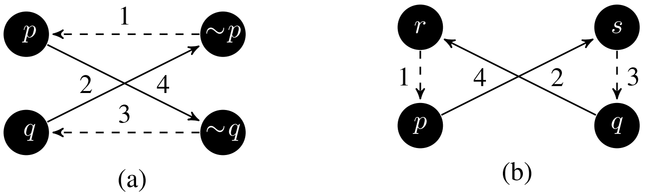

Figure 1. Warrant dependency graphs: (a) for $\mathcal{P}_3$, (b) for $\mathcal{P}_4$.

The characterization of the *unique output property* for a program $\mathcal{P} = (\Pi, \Delta)$ is done level-wise, starting from the highest level and iteratively going down from one level to next level below. For every level it consists in checking whether for some literal L, the warranty of L recursively depends on itself based on the topology of a warrant dependency graph defined as follows.

Definition 13 (Graph for a literal) *Let $\mathcal{P} = (\Pi, \Delta)$ be a RP-DeLP program, let $(Warr, Block)$ be an output for $\mathcal{P}$ and let L be a literal such that $L \in Warr(\alpha)$, for some level α. The graph for L w.r.t. Warr is the warrant dependency graph (V, E) for arguments $\{\mathcal{A}_1, \ldots, \mathcal{A}_k\}$ w.r.t. W and $\{\mathcal{B}_1, \ldots, \mathcal{B}_n\}$ where*

- *$W = Warr(\geq \alpha) \backslash \{L\}$,*
- *$\mathcal{A}_1 = \langle A_1, Q_1 \rangle, \ldots, \mathcal{A}_k = \langle A_k, Q_k \rangle$ are all arguments with strength α[9] that are acceptable w.r.t. W (according to Definition 2) and such that $Q_j \notin Warr(> \alpha)$ and $Q_j, \sim Q_j \notin Block(> \alpha)$, and*
- *$\mathcal{B}_1 = \langle B_1, P_1 \rangle, \ldots, \mathcal{B}_n = \langle B_n, P_n \rangle$ are all arguments with strength α that satisfy the following conditions [10]:*

  (i) $P_j \notin Warr(> \alpha)$ and $P_j \notin \{Q_1, \ldots, Q_k\}$,
  (ii) $P_j, \sim P_j \notin Block(> \alpha)$,
  (iii) for all $\langle C, R \rangle \sqsubset \mathcal{B}_j$ with strength $\beta > \alpha$, $R \in Warr(\beta)$ and for all $\langle C, R \rangle \sqsubset \mathcal{B}_j$ with strength α, $R \in \{Q_1, \ldots, Q_k\} \cup \{P_1, \ldots, P_n\}$,
  (iv) $\Pi \cup Warr(> \alpha) \cup \{\langle C, R \rangle \sqsubset \mathcal{B}_j\} \cup \{P_j\} \not\vdash \bot$,
  (v) there exists an argument $\mathcal{S} \in \{\mathcal{A}_1, \ldots, \mathcal{A}_k\}$ such that $\mathcal{S} \sqsubset \mathcal{B}_j$, and

[9] Remark that for all argument $\mathcal{A}_j \in \{\mathcal{A}_1, \ldots, \mathcal{A}_k\}$ with $Q_j \neq L$, $\mathcal{A}_j$ does not depend on L and either $Q_j \in Warr(\alpha)$ or $Q_j \in Block(\alpha)$.

[10] Remark that for all argument $\mathcal{B}_j \in \{\mathcal{B}_1, \ldots, \mathcal{B}_k\}$, either $\mathcal{B}_j$ depends on L and $(P_j, \alpha) \in Warr(\alpha) \cup Block(\alpha)$ or $\mathcal{B}_j$ depends on some $Q_j \in Block(\alpha)$.

(vi) for every argument $\mathcal{S} \in \{\mathcal{A}_1, \dots, \mathcal{A}_k\}$ such that $\mathcal{S} \sqsubset \mathcal{B}_j$, there does not exist a set of arguments $G \subseteq \{\mathcal{A}_1, \dots, \mathcal{A}_k\} \backslash \{\mathcal{S}\}$ such that $G \cup \{\mathcal{S}\}$ generates a conflict w.r.t. W.

Proposition 14 (RP-DeLP program with unique output) *Let $\mathcal{P} = (\Pi, \Delta)$ be a RP-DeLP program and let $(Warr, Block)$ be an output for $\mathcal{P}$. $(Warr, Block)$ is the unique output for $\mathcal{P}$ iff for all literal $L \in Warr$ there is no cycle in the graph for L w.r.t. Warr.*

Intuitively, given a literal L such that $L \in Warr(\alpha)$, for some program preference level α, Definition 13 builds the warrant dependency graph for L and all acceptable arguments $\{\mathcal{A}_1, \dots, \mathcal{A}_k\}$ of strength α that do not depend on L w.r.t. arguments $\{\mathcal{B}_1, \dots, \mathcal{B}_n\}$ of strength α whose supports depend on L or on some argument in $\{\mathcal{A}_1, \dots, \mathcal{A}_k\}$. Then, according to Definition 11, the existence of a cycle expresses that the warranty of the argument for L depends on the validity of some $\mathcal{B} \in \{\mathcal{B}_1, \dots, \mathcal{B}_n\}$, which depends on the warranty of some $L' \in \{\mathcal{A}_1, \dots, \mathcal{A}_k\}$ with $L \neq L'$, which in turn depends on the validity of some $\mathcal{B}' \in \{\mathcal{B}_1, \dots, \mathcal{B}_n\}$ with $B' \neq B$, which in turn depends on the warranty of L. Thus, for arguments of L and L' there does not exist a (unique) conflict evaluation order. Obviously, for RP-DeLP programs with unique output the set of warranted conclusions for every level α can be (computed) defined by an unique conflict evaluation order between arguments. Next we show that programs of Example 12 have multiple outputs.

Example 15 *According to Definition 8, $Output_1 = (Warr_1, Block_1)$ with $Warr_1 = \{p\}$ and $Block_1 = \{q, \sim q\}$, is an output for program $\mathcal{P}_3$ of Example 12. Then, according to Definition 13, Figure 1 shows the graph for p w.r.t. $Warr_1$; i.e. the warrant dependency graph for arguments $\{\mathcal{A}_1, \mathcal{A}_2\}$ w.r.t. W and $\{\mathcal{B}_1, \mathcal{B}_2\}$ with*

$\mathcal{A}_1 = \langle\{p\}, p\rangle$, $\mathcal{A}_2 = \langle\{q\}, q\rangle$, $W = \emptyset$,
$\mathcal{B}_1 = \langle\{q, \ \sim p \leftarrow q\}, \sim p\rangle$ *and* $\mathcal{B}_2 = \langle\{p, \ \sim q \leftarrow p\}, \sim q\rangle$.

Therefore, according to Proposition 14, $Output_1$ is not the unique output for $\mathcal{P}_3$ since there is a cycle in the graph for p w.r.t. $Warr_1$. Notice that $Output_2 = (Warr_2, Block_2)$ with $Warr_2 = \{q\}$ and $Block_2 = \{p, \sim p\}$, is also an output for program $\mathcal{P}_3$ and the graph for q w.r.t. $Warr_2$ also contains a cycle.

Consider now the RP-DeLP program $\mathcal{P}_4$ of Example 12. According to Definition 8, $Output_1 = (Warr_1, Block_1)$ with $Warr_1 = \{y, p\}$ and $Block_1 = \{q, s\}$, is an output for $\mathcal{P}_4$. Then, according to Definition 13, Figure 1 shows the graph for p w.r.t. $W = Warr_1(1) = \{y\}$ proving that the output for $\mathcal{P}_4$ is not unique. Indeed, notice that $Output_2 = (Warr_2, Block_2)$ with $Warr_2 = \{y, q\}$ and $Block_2 = \{p, r\}$, is also an output for program $\mathcal{P}_4$ and the graph for q w.r.t. $Warr_2$ also contains a cycle.

One of the main advantages of the warrant recursive semantics for RP-DeLP is from the implementation point of view. Actually, warrant semantics based on dialectical trees and, in general, rule-based argumentation frameworks like DeLP [7,9], might consider an exponential number of arguments with respect to the number of rules of a given program. In contrast, in our framework, at least for the particular case of RP-DeLP programs with unique output, it is not necessary to explicitly compute all the possible arguments for a given literal to check whether it is warranted, as we can implement an algorithm[11] (not shown here due to space limitations) with a worst-case complexity in P^{NP}.

[11] Details can be found in the extended version at *http://ia.udl.cat/ramon/comma2010full.pdf*

5. Conclusions and future work

In this paper we have introduced a new recursive semantics for determining the warranty status of arguments in defeasible argumentation. The distinctive features of this semantics, e.g. with respect to Pollock's critical link semantics, are: (i) it is based on a non-binary notion of conflict in order to preserve consistency with the strict knowledge and (ii) besides the set of warranted and rejected conclusions, we introduce the set of blocked conclusions, which are those conclusions which are based on warranted information but they generate a conflict with other already warranted conclusions of the same strength.

As future work we plan to formalize the maximal ideal output for RP-DeLP programs which will allow us to characterize the relationship between this unique output based on the recursive warrant semantics and the output of DeLP [10] and other general argumentation frameworks [5,4] based on the use of dialectical trees as underlying structures for characterizing the semantics of warranted conclusions.

Acknowledgments Authors are thankful to the anonymous reviewers for their helpful comments. Research partially funded by the Spanish MICINN projects MULOG2 (TIN2007-68005-C04-01/02) and ARINF (TIN2009-14704-C03-01/03), CONSOLIDER (CSD2007-0022), and ESF Eurocores-LogICCC/MICINN (FFI2008-03126-E/FILO), and the grant JC2009-00272 from the Ministerio de Educación.

References

[1] T. Alsinet, C.I. Chesñevar, and L Godo. Enforcing indirect consistency in possibilistic defeasible logic programming: a level-based approach. In *IPMU'08*, pages 497–504, 2008.

[2] T. Alsinet, C.I. Chesñevar, and L Godo. A level-based approach to computing warranted arguments in possibilistic defeasible logic programming. In *COMMA'08*, pages 1–12, 2008.

[3] T. Alsinet, C.I. Chesñevar, L. Godo, and G. Simari. A logic programming framework for possibilistic argumentation: Formalization and logical properties. *Fuzzy Sets and Systems*, 159 (10): 1208–1228, 2008.

[4] L. Amgoud and C. Cayrol. On the Acceptability of Arguments in Preference-based Argumentation. In UAI'08, pages 1–7, 1998.

[5] P. Besnard and A. Hunter. *Elements of Argumentation*. The MIT Press, 2008.

[6] M. Caminada and L. Amgoud. On the evaluation of argumentation formalisms. *Artificial Intelligence*, 171 (5-6): 286–310, 2007.

[7] L. Cecchi, P. Fillottrani, and G. Simari. On the complexity of DeLP through game semantics. In NMR Workshop, pages 386–394, 2006.

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

[9] C.I. Chesñevar, G. Simari, and L. Godo. Computing dialectical trees efficiently in possibilistic defeasible logic programming. In *LPNMR'05*, pages 158–171, 2005.

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

[11] S.H. Nielsen and S. Parsons. A Generalization of Dung's Abstract Framework for Argumentation: Arguing with Sets of Attacking Arguments. In *ArgMAS'06*, pages 54–73, 2006.

[12] J.L. Pollock. A recursive semantics for defeasible reasoning. In Iyad Rahwan and Guillermo R. Simari, editors, *Argumentation in Artificial Intelligence*, chapter 9, pages 173–198. Springer, 2009.

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

[14] 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.

[15] I. Rahwan and G. Simari, editors. *Argumentation in Artificial Intelligence*. Springer, 2009.

[16] G. Vreeswijk. Abstract Argumentation Systems. *Artificial Intelligence*, 90(1-2): 225–279,1997.