

A Level-based Approach to Computing Warranted Arguments in Possibilistic Defeasible Logic Programming

Teresa Alsinet ^{a,1}, Carlos Chesñevar ^b and Lluís Godo ^c

^a *Department of Computer Science, University of Lleida, SPAIN*

^b *Dept. of Computer Science & Eng. Universidad Nacional del Sur, ARGENTINA*

^c *Artificial Intelligence Research Institute (IIIA-CSIC), Bellaterra, SPAIN*

Abstract. Possibilistic Defeasible Logic Programming (P-DeLP) is an argumentation framework based on logic programming which incorporates a treatment of possibilistic uncertainty at object-language level. In P-DeLP, the closure of justified conclusions is not always consistent, which has been detected to be an anomaly in the context of so-called rationality postulates for rule-based argumentation systems. In this paper we present a novel level-based approach to computing warranted arguments in P-DeLP which ensures the above rationality postulate. We also show that our solution presents some advantages in comparison with the use of a transposition operator applied on strict rules.

Keywords. Formal models of argument, Possibilistic logic, Rationality postulates for argumentation

1. Introduction and motivation

Possibilistic Defeasible Logic Programming (P-DeLP) [1] is an argumentation framework based on logic programming which incorporates the treatment of possibilistic uncertainty at the object-language level. Indeed, P-DeLP is an extension of Defeasible Logic Programming (DeLP) [9], a logic programming approach to argumentation which has been successfully used to solve real-world problems in several contexts such as knowledge distribution [3] and recommendations systems [8], among others. As in the case of DeLP, the P-DeLP semantics is skeptical, based on a query-driven proof procedure which computes warranted (justified) arguments. Following the terminology used in [5], P-DeLP can be seen as a member of the family of *rule-based argumentation systems*, as it is based on a logical language defined over a set of (weighed) literals and the notions of *strict* and *defeasible* rules, which are used to characterize a P-DeLP program.

Recently Caminada & Amgoud have defined several *rationality postulates* [5] which every rule-based argumentation system should satisfy. One of such postulates (called *Indirect Consistency*) involves ensuring that the closure of warranted conclusions be guar-

¹Correspondence to: T. Alsinet. Department of Computer Science, University of Lleida. C/Jaume II, 69. Lleida, Spain. Tel.: +34 973 70 2734; Fax: +34 973 70 2702; E-mail: tracy@diei.udl.cat

anteed to be consistent. Failing to satisfy this postulate implies some anomalies un-intuitive results (e.g. the modus ponens rule cannot be applied based on justified conclusions). A number of rule-based argumentation systems are identified in which such postulate does not hold (including DeLP [9] and Prakken & Sartor [11], among others). As an alternative to solve this problem, the use of *transposed rules* is proposed to extend the representation of strict rules. For grounded semantics, the use of a transposition operator ensures that all rationality postulates to be satisfied [5, pp.294].

In this paper we present a novel level-based approach to computing warranted arguments in P-DeLP which ensures the above rationality postulate without requiring the use of transposed rules. Additionally, in contrast with DeLP and other argument-based approaches, we do not require the use of dialectical trees as underlying structures for characterizing our proof procedure. We show that our solution presents some advantages in comparison with the use of a transposition operator applied on strict rules, which might lead to unintuitive results in some cases. In particular, we show that adding transposed rules can turn a valid (consistent) P-DeLP program into an inconsistent one, disallowing further argument-based inferences on the basis of such a program.

The rest of the paper is structured as follows. Section 2 summarizes the main elements of P-DeLP. Section 3 discusses the role of the rationality postulate of indirect consistency introduced in [4], and the solution provided in terms of a transposition operator Cl_{tp} . We also show some aspects which may be problematic from this approach in P-DeLP. Section 4 presents our new level-based definitions of warrant for P-DeLP, as well as some illustrative examples. We also show that this characterization ensures that the above postulate can be now satisfied without requiring the use of transposed rules nor the computation of dialectical trees. Finally Section 5 discusses some related work and concludes.

2. Argumentation in P-DeLP: an overview

In order to make this paper self-contained, we will present next the main definitions that characterize the P-DeLP framework. For details the reader is referred to [1]. The language of P-DeLP is inherited from the language of logic programming, including the usual notions of atom, literal, rule and fact, but over an extended set of atoms where a new atom “ $\sim p$ ” is added for each original atom p . Therefore, a *literal* in P-DeLP is either an atom p or a (negated) atom of the form $\sim p$, and a *goal* is any literal.

A *weighted clause* is a pair of the form (φ, α) , where φ is a rule $Q \leftarrow P_1 \wedge \dots \wedge P_k$ or a fact Q (i.e., a rule with empty antecedent), where Q, P_1, \dots, P_k are literals, and $\alpha \in [0, 1]$ expresses a lower bound for the necessity degree of φ . We distinguish between *certain* and *uncertain* clauses. A clause (φ, α) is referred as certain if $\alpha = 1$ and uncertain, otherwise. A set of P-DeLP clauses Γ will be deemed as *contradictory*, denoted $\Gamma \vdash \perp$, if, for some atom q , $\Gamma \vdash (q, \alpha)$ and $\Gamma \vdash (\sim q, \beta)$, with $\alpha > 0$ and $\beta > 0$, where \vdash stands for deduction by means of the following particular instance of the *generalized modus ponens rule*:

$$\frac{(Q \leftarrow P_1 \wedge \dots \wedge P_k, \alpha) \quad (P_1, \beta_1), \dots, (P_k, \beta_k)}{(Q, \min(\alpha, \beta_1, \dots, \beta_k))} \text{ [GMP]}$$

Formally, we will write $\Gamma \vdash (Q, \alpha)$, where Γ is a set of PGL clauses, Q is a literal and $\alpha > 0$, when there exists a finite sequence of PGL clauses C_1, \dots, C_m such that $C_m = (Q, \alpha)$ and, for each $i \in \{1, \dots, m\}$, either $C_i \in \Gamma$, or C_i is obtained by applying the *GMP* rule to previous clauses in the sequence.

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. Formally, given a program $\mathcal{P} = (\Pi, \Delta)$, we say that 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. $\Pi \cup \mathcal{A}$ is non contradictory;
2. $\alpha = \max\{\beta \in [0, 1] \mid \Pi \cup \mathcal{A} \vdash (Q, \beta)\}$, i.e. α is the greatest degree of deduction of Q from $\Pi \cup \mathcal{A}$;
3. \mathcal{A} is minimal wrt set inclusion, i.e. there is no $\mathcal{A}_1 \subset \mathcal{A}$ such that $\Pi \cup \mathcal{A}_1 \vdash (Q, \alpha)$.

Moreover, if $\langle \mathcal{A}, Q, \alpha \rangle$ and $\langle \mathcal{S}, R, \beta \rangle$ are two arguments wrt a program $\mathcal{P} = (\Pi, \Delta)$, we say that $\langle \mathcal{S}, R, \beta \rangle$ is a *subargument* of $\langle \mathcal{A}, Q, \alpha \rangle$, denoted $\langle \mathcal{S}, R, \beta \rangle \sqsubseteq \langle \mathcal{A}, Q, \alpha \rangle$, whenever $\mathcal{S} \subseteq \mathcal{A}$. Notice that the goal R may be any subgoal associated with the goal Q in the argument \mathcal{A} . From the above definition of argument, note that if $\langle \mathcal{S}, R, \beta \rangle \sqsubseteq \langle \mathcal{A}, Q, \alpha \rangle$ it holds that: (i) $\beta \geq \alpha$, and (ii) if $\beta = \alpha$, then $\mathcal{S} = \mathcal{A}$ iff $R = Q$.

Let \mathcal{P} be a P-DeLP 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 say that $\langle \mathcal{A}_1, Q_1, \alpha_1 \rangle$ counterargues $\langle \mathcal{A}_2, Q_2, \alpha_2 \rangle$ ¹ 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 $Q_1 = \sim Q$. Moreover, if the argument $\langle \mathcal{A}_1, Q_1, \alpha_1 \rangle$ counterargues the argument $\langle \mathcal{A}_2, Q_2, \alpha_2 \rangle$ with disagreement subargument $\langle \mathcal{A}, Q, \beta \rangle$, we say that $\langle \mathcal{A}_1, Q_1, \alpha_1 \rangle$ is a *proper* (respectively *blocking*) *defeater* for $\langle \mathcal{A}_2, Q_2, \alpha_2 \rangle$ when $\alpha_1 > \beta$ (respectively $\alpha_1 = \beta$).

In P-DeLP, as in other argumentation systems [7,12], argument-based inference involves a dialectical process in which arguments are compared in order to determine which beliefs or goals are ultimately accepted (or *justified* or *warranted*) on the basis of a given program and is formalized in terms of an exhaustive dialectical analysis of all possible argumentation lines rooted in a given argument. An *argumentation line* starting in an argument $\langle \mathcal{A}_0, Q_0, \alpha_0 \rangle$ is a sequence of arguments $\lambda = [\langle \mathcal{A}_0, Q_0, \alpha_0 \rangle, \langle \mathcal{A}_1, Q_1, \alpha_1 \rangle, \dots, \langle \mathcal{A}_n, Q_n, \alpha_n \rangle, \dots]$ such that each $\langle \mathcal{A}_i, Q_i, \alpha_i \rangle$ is a defeater for the previous argument $\langle \mathcal{A}_{i-1}, Q_{i-1}, \alpha_{i-1} \rangle$ in the sequence, $i > 0$. In order to avoid *fallacious* reasoning additional constraints are imposed, namely:

1. **Non-contradiction:** given an argumentation line λ , the set of arguments of the proponent (respectively opponent) should be *non-contradictory* wrt \mathcal{P} .²
2. **Progressive argumentation:** (i) every blocking defeater $\langle \mathcal{A}_i, Q_i, \alpha_i \rangle$ in λ with $i > 0$ is defeated by a proper defeater³ $\langle \mathcal{A}_{i+1}, Q_{i+1}, \alpha_{i+1} \rangle$ in λ ; and (ii) each argument $\langle \mathcal{A}_i, Q_i, \alpha_i \rangle$ in λ , with $i \geq 2$, is such that $Q_i \neq \sim Q_{i-1}$.

¹In what follows, for a given goal Q , we will write $\sim Q$ as an abbreviation to denote “ $\sim q$ ”, if $Q \equiv q$, and “ q ”, if $Q \equiv \sim q$.

²Non-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.

³It must be noted that the last argument in an argumentation line is allowed to be a blocking defeater for the previous one.

The non-contradiction condition disallows the use of contradictory information on either side (proponent or opponent). The first condition of progressive argumentation enforces the use of a proper defeater to defeat an argument which acts as a blocking defeater, while the second condition avoids non optimal arguments in the presence of a conflict. An argumentation line satisfying the above restrictions is called *acceptable*, and can be proven to be finite. The set of all possible acceptable argumentation lines results in a structure called *dialectical tree*. Given a program $\mathcal{P} = (\Pi, \Delta)$ and a goal Q , we say that Q is *warranted* wrt \mathcal{P} with a *maximum necessity degree* α iff there exists an argument $\langle \mathcal{A}, Q, \alpha \rangle$, for some $\mathcal{A} \subseteq \Delta$, such that: i) every acceptable argumentation line starting with $\langle \mathcal{A}, Q, \alpha \rangle$ has an odd number of arguments; and ii) there is no other argument of the form $\langle \mathcal{A}_1, Q, \beta \rangle$, with $\beta > \alpha$, satisfying the above. In the rest of the paper we will write $\mathcal{P} \mid \sim^w \langle \mathcal{A}, Q, \alpha \rangle$ to denote this fact.

3. Indirect consistency as rationality postulate. Transposition of strict rules

In a recent paper Caminada and Amgoud [5] have defined a very interesting characterization of three *rationality postulates* that –according to the authors– any rule-based argumentation system should satisfy in order to avoid anomalies and unintuitive results. We will summarize next the main aspects of these postulates, and their relationship with the P-DeLP framework. Their formalization is intentionally generic, based on a *defeasible theory* $\mathcal{T} = \langle \mathcal{S}, \mathcal{D} \rangle$, where \mathcal{S} is a set of strict rules and \mathcal{D} is a set of defeasible rules. The notion of negation is modelled in the standard way by means of a function “–”. An *argumentation system* is a pair $\langle \text{Args}, \text{Def} \rangle$, where *Args* is a set of arguments (based on a defeasible theory) and $\text{Def} \subseteq \text{Args} \times \text{Args}$ is a defeat relation. The *closure* of a set of literals \mathcal{L} under the set \mathcal{S} , denoted $CL_{\mathcal{S}}(\mathcal{L})$ is the smallest set such that $\mathcal{L} \subseteq CL_{\mathcal{S}}(\mathcal{L})$, and if $\phi_1, \dots, \phi_n \rightarrow \psi \in \mathcal{S}$, and $\phi_1, \dots, \phi_n \in CL_{\mathcal{S}}(\mathcal{L})$, then $\psi \in CL_{\mathcal{S}}(\mathcal{L})$. A set of literals \mathcal{L} is *consistent* iff there not exist $\psi, \phi \in \mathcal{L}$ such that $\psi = -\phi$, otherwise it is said to be *inconsistent*. An argumentation system $\langle \text{Args}, \text{Def} \rangle$ can have different *extensions* E_1, E_2, \dots, E_n ($n \geq 1$) according to the adopted semantics. The conclusions associated with those arguments belonging to a given extension E_i are defined as $\text{Concs}(E_i)$, and the *output* of the argumentation system is defined skeptically as $\text{Output} = \bigcap_{i=1 \dots n} \text{Concs}(E_i)$.

On the basis of the above concepts, Caminada & Amgoud [5, pp.294] present three important postulates: *direct consistency*, *indirect consistency* and *closure*. Let \mathcal{T} be a defeasible theory, $\langle \text{Args}, \text{Def} \rangle$ an argumentation system built from \mathcal{T} , Output the set of justified (warranted) conclusions, and E_1, \dots, E_n its extensions under a given semantics. Then these three postulates are defined as follows:

- $\langle \text{Args}, \text{Def} \rangle$ satisfies **closure** iff (1) $\text{Concs}(E_i) = CL_{\mathcal{S}}(\text{Concs}(E_i))$ for each $1 \leq i \leq n$ and (2) $\text{Output} = CL_{\mathcal{S}}(\text{Output})$.
- $\langle \text{Args}, \text{Def} \rangle$ satisfies **direct consistency** iff (1) $\text{Concs}(E_i)$ is consistent for each $1 \leq i \leq n$ and (2) Output is consistent.
- $\langle \text{Args}, \text{Def} \rangle$ satisfies **indirect consistency** iff (1) $CL_{\mathcal{S}}(\text{Concs}(E_i))$ is consistent for each $1 \leq i \leq n$ and (2) $CL_{\mathcal{S}}(\text{Output})$ is consistent.

Closure accounts for requiring that the set of justified conclusions as well as the set of conclusions supported by each extension are closed. Direct consistency implies that

the set of justified conclusions and the different sets of conclusions corresponding to each extension are consistent. Indirect consistency involves a more subtle case, requiring that the closure of both $\text{Concs}(E_i)$ and Output is consistent.

Caminada and Amgoud show that many rule-based argumentation system (e.g. Prakken & Sartor [11] and DeLP [9]) fail to satisfy indirect consistency, detecting as a solution the definition of a special *transposition operator* Cl_{tp} for computing the closure of strict rules. This accounts for taking every strict rule $r = \phi_1, \phi_2, \dots, \phi_n \rightarrow \psi$ as a material implication in propositional logic which is equivalent to the disjunction $\phi_1 \vee \phi_2 \vee \dots, \phi_n \vee \neg\psi$. From that disjunction different rules of the form $\phi_1, \dots, \phi_{i-1}, \neg\psi, \phi_{i+1}, \dots, \phi_n \rightarrow \neg\phi_i$ can be obtained (*transpositions* of r). If \mathcal{S} is a set of strict rules, Cl_{tp} is the minimal set such that (1) $\mathcal{S} \subseteq Cl_{tp}(\mathcal{S})$ and (2) If $s \in Cl_{tp}(\mathcal{S})$ and t is a transposition of s , then $t \in Cl_{tp}(\mathcal{S})$. The use of such an operator allows the three rationality postulates to be satisfied in the case of the grounded extension (which corresponds to the one associated with systems like DeLP or P-DeLP).

Theorem 1 [5] *Let $\langle \text{Args}, \text{Def} \rangle$ be an argumentation system built from $\langle Cl_{tp}(\mathcal{S}), \mathcal{D} \rangle$, where $Cl_{tp}(\mathcal{S})$ is consistent, Output is the set of justified conclusions and E its grounded extension. Then $\langle \text{Args}, \text{Def} \rangle$ satisfies closure and indirect consistency.*

Caminada & Amgoud show that DeLP does not satisfy the indirect consistency postulate. The same applies for P-DeLP, as illustrated next. Consider the program $\mathcal{P} = (\Pi, \Delta)$, where $\Pi = \{ (y, 1), (\sim y \leftarrow a \wedge b, 1) \}$ and $\Delta = \{ (a, 0.9), (b, 0.9) \}$. It is easy to see that arguments $\langle \{(a, 0.9)\}, a, 0.9 \rangle$ and $\langle \{(b, 0.9)\}, b, 0.9 \rangle$ have no defeaters wrt \mathcal{P} . Thus $\{y, a, b\} = \text{Output}$ turns out to be warranted, and it holds that $y, \sim y \in CL_{Cl_{tp}(\Pi)}(\{y, a, b\})$, so that indirect consistency does not hold.

We think that Caminada & Amgoud's postulate of indirect consistency is indeed valuable for rule-based argumentation systems, as in some sense it allows to perform "forward reasoning" from warranted literals. However, P-DeLP and DeLP are *Horn*-based systems, so that strict rules should be read as inference rules rather than as material implications. In this respect, the use of transposed rules might lead to unintuitive situations in a logic programming context. Consider e.g. the program $\mathcal{P} = \{ (q \leftarrow p \wedge r, 1), (s \leftarrow \sim r, 1), (p, 1), (\sim q, 1), (\sim s, 1) \}$. In P-DeLP, the facts $(p, 1)$, $(\sim q, 1)$ and $(\sim s, 1)$ would be warranted literals. However, the closure under transposition $Cl_{tp}(\mathcal{P})$ would include the rule $(\sim r \leftarrow p \wedge \sim q, 1)$, resulting in inconsistency (both $(\sim s, 1)$ and $(s, 1)$ can be derived), so that the whole program would be deemed as invalid. Our goal is to retain a Horn-based view for a rule-based argumentation system like P-DeLP, satisfying at the same time the indirect consistency postulate. To do this we will not take into account transposed rules, introducing instead the notion of *level-based* warranted literals, as discussed in the next Section.

4. A level-based approach to computing warranted arguments

In a logic programming system like P-DeLP the use of transposed rules to ensure indirect consistency may have some drawbacks that have to be taken into consideration. Apart from the problem mentioned at the end of last section of turning an apparently valid program into a non-valid one, there are two other issues: (i) a computational lim-

itation, in the sense that extending a P-DeLP program with all possible transpositions of every strict rule may lead to an important increase in the number of arguments to be computed; and (ii) when doing so, the system can possibly establish as warranted goals conclusions which are not explicitly expressed in the original program. For instance, consider the program $\mathcal{P} = \{(\sim y \leftarrow a \wedge b, 1), (y, 1), (a, 0.9), (b, 0.7)\}$. Transpositions of the strict rule $(\sim y \leftarrow a \wedge b, 1)$ are $(\sim a \leftarrow y \wedge b, 1)$ and $(\sim b \leftarrow y \wedge a, 1)$. Then, the argument $\langle \mathcal{A}, \sim b, 0.9 \rangle$, with $\mathcal{A} = \{(y, 1), (a, 0.9), (\sim b \leftarrow a \wedge y, 1)\}$, is warranted wrt \mathcal{P} , although no explicit information is given for the literal $\sim b$ in \mathcal{P} . In this paper we will provide a new formal definition of warranted goal with maximum necessity degree which will take into account direct and indirect conflicts between arguments. Indirect conflicts will be detected without explicitly transposing strict rules, distinguishing between *warranted* and *blocked* goals.

Direct conflicts between arguments refer to the case of both proper and blocking defeaters. For instance, consider the program $\mathcal{P} = \{(a \leftarrow b, 0.9), (b, 0.8), (\sim b, 0.8)\}$. Thus arguments $\langle \{(b, 0.8)\}, b, 0.8 \rangle$ and $\langle \{(\sim b, 0.8)\}, \sim b, 0.8 \rangle$ are a pair of blocking defeaters expressing (direct) contradictory information, and therefore b and $\sim b$ will be considered a pair of blocked goals with maximum necessity degree 0.8. Note that although the argument $\langle \{(\sim b, 0.8)\}, \sim b, 0.8 \rangle$ is a blocking defeater for the argument $\langle \mathcal{A}, a, 0.8 \rangle$, with $\mathcal{A} = \{(a \leftarrow b, 0.9), (b, 0.8)\}$, goals a and $\sim b$ do not express contradictory information, and therefore a is a neither blocked nor warranted goal with necessity degree 0.8.

On the other hand, we will refer to indirect conflicts between arguments when there exists an inconsistency emerging from the set of certain (strict) clauses of a program and arguments with no defeaters. For instance, consider the program $\mathcal{P} = (\Pi, \Delta)$ with $\Pi = \{(\sim y \leftarrow a \wedge b, 1), (y, 1), (\sim x \leftarrow c \wedge d, 1), (x, 1)\}$ and $\Delta = \{(a, 0.7), (b, 0.7), (c, 0.7), (d, 0.6)\}$. In standard P-DeLP [1], (i.e. without extending the program with transpositions of rules of Π) $\langle \{(a, 0.7)\}, a, 0.7 \rangle$ and $\langle \{(b, 0.7)\}, b, 0.7 \rangle$ are arguments with no defeaters and therefore their conclusions would be warranted. However, since $\Pi \cup \{(a, 0.7), (b, 0.7)\} \vdash \perp$, arguments $\langle \{(a, 0.7)\}, a, 0.7 \rangle$ and $\langle \{(b, 0.7)\}, b, 0.7 \rangle$ express (indirect) contradictory information. Moreover, as both goals are supported by arguments with the same necessity degree 0.7, none of them can be warranted nor rejected, and therefore we will refer to them as (indirect) blocked goals with maximum necessity degree 0.7. On the other hand, a similar situation appears with $\langle \{(c, 0.7)\}, c, 0.7 \rangle$ and $\langle \{(d, 0.6)\}, d, 0.6 \rangle$. As before, $\Pi \cup \{(c, 0.7), (d, 0.6)\} \vdash \perp$, but in this case the necessity degree of goal c is greater than the necessity degree of goal d . Therefore c will be considered a warranted goal with maximum necessity degree 0.7.

Let $ARG(\mathcal{P}) = \{\langle \mathcal{A}, Q, \alpha \rangle \mid \mathcal{A} \text{ is an argument for } Q \text{ with necessity } \alpha \text{ wrt } \mathcal{P}\}$ and let $Concl(\mathcal{P}) = \{(Q, \alpha) \mid \langle \mathcal{A}, Q, \alpha \rangle \in ARG(\mathcal{P})\}$. An *output* for a P-DeLP program \mathcal{P} will be a pair $(Warr, Block)$, where $Warr, Block \subseteq Concl(\mathcal{P})$, denoting respectively a set of warranted and blocked goals (together with their degrees) and fulfilling a set of conditions that will ensure a proper handling of the problem of global inconsistency discussed earlier, and that will specified in the following definition. Since the intended construction of the sets $Warr, Block$ is done level-wise, starting from the first level and iteratively going from one level to next level below, we introduce some useful notation. Indeed, if $1 \geq \alpha_1 > \alpha_2 > \dots > \alpha_p > 0$ are the weights appearing in arguments from $ARG(\mathcal{P})$, we can stratify the sets by putting $Warr = Warr(\alpha_1) \cup \dots \cup Warr(\alpha_p)$ and similarly $Block = Block(\alpha_1) \cup \dots \cup Block(\alpha_p)$, where $Warr(\alpha_i)$ and $Block(\alpha_i)$ are

respectively the sets of the warranted and blocked goals with maximum degree α_i . We will also write $Warr(> \alpha_i)$ to denote $\cup_{\beta > \alpha_i} Warr(\beta)$, and analogously for $Block(> \alpha_i)$. In what follows, given a program $\mathcal{P} = (\Pi, \Delta)$ we will denote by $rules(\Pi)$ and $facts(\Pi)$ the set of strict rules and strict facts of \mathcal{P} respectively.

Definition 1 (Warranted and blocked goals) *Given a program $\mathcal{P} = (\Pi, \Delta)$, an output for \mathcal{P} is a pair $(Warr, Block)$ where the sets $Warr(\alpha_i)$ and $Block(\alpha_i)$, for $i = 1 \dots p$ are required to satisfy the following constraints:*

1. An argument $\langle \mathcal{A}, Q, \alpha_i \rangle \in ARG(\mathcal{P})$ is called acceptable if it satisfies the following three conditions:
 - (i) $(Q, \beta) \notin Warr(> \alpha_i) \cup Block(> \alpha_i)$ and $(\sim Q, \beta) \notin Block(> \alpha_i)$, for all $\beta > \alpha$
 - (ii) for any subargument $\langle \mathcal{B}, R, \beta \rangle \sqsubseteq \langle \mathcal{A}, Q, \alpha_i \rangle$ such that $R \neq Q$, $(R, \beta) \in Warr(\beta)$
 - (iii) $rules(\Pi) \cup Warr(> \alpha_i) \cup \{(R, \alpha_i) \mid \langle \mathcal{B}, R, \alpha_i \rangle \sqsubseteq \langle \mathcal{A}, Q, \alpha_i \rangle\} \not\vdash \perp$.
2. For each acceptable $\langle \mathcal{A}, Q, \alpha_i \rangle \in ARG(\mathcal{P})$, $(Q, \alpha_i) \in Block(\alpha_i)$ whenever
 - (i) either there exists an acceptable $\langle \mathcal{B}, \sim Q, \alpha_i \rangle \in ARG$; or
 - (ii) there exists $G \subseteq \{(P, \alpha_i) \mid \langle \mathcal{C}, P, \alpha_i \rangle \in ARG \text{ is acceptable and } \sim P \notin Block(\alpha_i)\}$ such that $rules(\Pi) \cup Warr(> \alpha_i) \cup G \not\vdash \perp$ and $rules(\Pi) \cup Warr(> \alpha_i) \cup G \cup \{(Q, \alpha_i)\} \vdash \perp$;

otherwise, $(Q, \alpha_i) \in Warr(\alpha_i)$.

Note that in Def. 1 the notion of argument ensures that for each argument $\langle \mathcal{A}, Q, \alpha \rangle \in ARG$, the goal Q is non-contradictory wrt the set Π of certain clauses of \mathcal{P} . However, it does not ensure non-contradiction wrt Π together with the set $Warr(> \alpha)$ of warranted goals with degree greater than α (as required by the indirect consistency postulate [5]). Therefore, for each argument $\langle \mathcal{A}, Q, \alpha \rangle \in ARG$ satisfying that each subgoal is warranted, the goal Q can be warranted at level α only after explicitly checking indirect conflicts wrt the set $Warr(> \alpha)$, i.e. after verifying that $rules(\Pi) \cup Warr(> \alpha) \cup \{(Q, \alpha)\} \not\vdash \perp$. For instance, consider the program $\mathcal{P} = (\Pi, \Delta)$ with

$$\begin{aligned} \Pi &= \{(y, 1), (\sim y \leftarrow a \wedge c, 1)\} \text{ and} \\ \Delta &= \{(a, 0.9), (b, 0.9), (c \leftarrow b, 0.8)\}. \end{aligned}$$

According to Def. 1, the goal y is warranted with necessity degree 1 and goals a and b are warranted with necessity degree 0.9. Then

$$\langle \{(b, 0.9), (c \leftarrow b, 0.8)\}, c, 0.8 \rangle$$

is an argument for c such that the subargument $\langle \{(b, 0.9)\}, b, 0.9 \rangle$ is warranted. However, as $rules(\Pi) \cup \{(y, 1), (a, 0.9), (b, 0.9)\} \cup \{(c, 0.8)\} \vdash \perp$, the goal c is neither warranted nor blocked wrt \mathcal{P} .

Suppose now in Def. 1, that an argument $\langle \mathcal{A}, Q, \alpha \rangle \in ARG$ involves a warranted subgoal with necessity degree α . Then Q can be warranted only after explicitly checking indirect conflicts wrt its set of subgoals, i.e. after verifying that $rules(\Pi) \cup Warr(> \alpha) \cup \{(R, \alpha) \mid \langle \mathcal{B}, R, \alpha \rangle \sqsubseteq \langle \mathcal{A}, Q, \alpha \rangle\} \not\vdash \perp$. For instance, consider the program $\mathcal{P} = (\Pi, \Delta)$, with

$$\begin{aligned}\Pi &= \{(y, 1), (\sim y \leftarrow a \wedge b, 1)\} \text{ and} \\ \Delta &= \{(a, 0.7), (b \leftarrow a, 0.7)\}.\end{aligned}$$

Then y and a are warranted goals with necessity degrees 1 and 0.7, respectively, and although it is not possible to compute a defeater for the argument

$$\langle \{(a, 0.7), (b \leftarrow a, 0.7)\}, b, 0.7 \rangle$$

in \mathcal{P} and the subgoal a is warranted with necessity degree 0.7, b is not warranted since $\{(\sim y \leftarrow a \wedge b, 1)\} \cup \{(y, 1)\} \cup \{(a, 0.7)\} \cup \{(b, 0.7)\} \vdash \perp$. Finally, note that in Def. 1, direct conflicts invalidate possible indirect conflicts in the following sense. Consider the program $\mathcal{P} = (\Pi, \Delta)$, with

$$\begin{aligned}\Pi &= \{(y \leftarrow a, 1), (\sim y \leftarrow b \wedge c, 1)\} \text{ and} \\ \Delta &= \{(a, 0.7), (b, 0.7), (c, 0.7), (\sim c, 0.7)\}.\end{aligned}$$

Then, c and $\sim c$ are blocked goals with necessity degree 0.7 and thus a , b and y are warranted goals with necessity degree 0.7. The next example illustrates some interesting cases of the notion of warranted and blocked goals in P-DeLP.

Example 2 Consider the program $\mathcal{P}_1 = (\Pi_1, \Delta_1)$, with

$$\begin{aligned}\Pi_1 &= \{(y, 1), (\sim y \leftarrow a \wedge b, 1)\} \text{ and} \\ \Delta_1 &= \{(a, 0.7), (b, 0.7), (\sim a, 0.5)\}.\end{aligned}$$

According to Def. 1, $(y, 1)$ is warranted and $(a, 0.7)$ and $(b, 0.7)$ are blocked. Then, as a is blocked with necessity degree 0.7, $\langle \{(\sim a, 0.5)\}, \sim a, 0.5 \rangle$ is not an acceptable argument and hence the goal $\sim a$ is neither warranted nor blocked wrt \mathcal{P}_1 .

Now consider the program $\mathcal{P}_2 = (\Pi_2, \Delta_2)$ with

$$\begin{aligned}\Pi_2 &= \{(y, 1), (\sim y \leftarrow a \wedge c, 1)\} \text{ and} \\ \Delta_2 &= \{(a, 0.9), (b, 0.9), (c \leftarrow b, 0.9)\}.\end{aligned}$$

According to Def. 1, $(y, 1)$ and $(b, 0.9)$ are warranted. On the other hand, $\langle \{(a, 0.9)\}, a, 0.9 \rangle$ is an argument for a with an empty set of subarguments and $\langle \{(b, 0.9), (c \leftarrow b, 0.9)\}, c, 0.9 \rangle$ is an argument for c satisfying that the subargument $\langle \{(b, 0.9)\}, b, 0.9 \rangle$ is warranted. However, as $\{(\sim y \leftarrow a \wedge c, 1)\} \cup \{(y, 1), (b, 0.9)\} \cup \{(a, 0.9), (c, 0.9)\} \vdash \perp$, a and c are a pair of blocked goals wrt \mathcal{P}_2 with necessity degree 0.9.

Finally, consider the program $\mathcal{P}_3 = (\Pi_2, \Delta_3)$ with

$$\Delta_3 = \{(a, 0.9), (c, 0.9), (b \leftarrow c, 0.9), (d \leftarrow a \wedge c, 0.9)\}.$$

In that case $(y, 1)$ is warranted and $(a, 0.9)$ and $(c, 0.9)$ are blocked. Then, according to Def. 1, as c is a blocked goal with necessity 0.9, $\langle \{(c, 0.9), (b \leftarrow c, 0.9)\}, b, 0.9 \rangle$ is not an acceptable argument and hence the goal b is neither warranted nor blocked wrt \mathcal{P}_3 . Notice that since a and c are contradictory wrt Π_2 , no argument can be computed for goal d .

It can be shown that if $(Warr, Block)$ is an output of a P-DeLP program, the set $Warr$ of warranted goals (according to Def. 1) is indeed non-contradictory and satisfies indirect consistency with respect to the set of strict rules.

Proposition 3 (Indirect consistency) *Let $\mathcal{P} = (\Pi, \Delta)$ be a P-DeLP program and let $(Warr, Block)$ be an output for \mathcal{P} . Then:*

- (i) $facts(\Pi) \subseteq Warr$,
- (ii) $Warr \not\vdash \perp$, and
- (iii) $rules(\Pi) \cup Warr \vdash (Q, \alpha)$ implies $(Q, \beta) \in Warr$, for some $\beta \geq \alpha$.

Actually, (iii) above can be read also as saying that $Warr$ satisfies (somewhat softened) the *closure* postulate with respect to the set of strict rules. Indeed, it could be recovered in the full sense if the deduction characterized by \vdash would be defined taking only into account those derivations yielding maximum degrees of necessity.

Proof: We prove (ii) and (iii), as (i) is straightforward.

(ii) Suppose that for some goal Q , $\{(Q, \alpha), (\sim Q, \beta)\} \subseteq Warr$. Then, there should exist $\mathcal{A} \subseteq \Delta$ and $\mathcal{B} \subseteq \Delta$ such that $\Pi \cup \mathcal{A} \vdash (Q, \alpha)$ and $\Pi \cup \mathcal{B} \vdash (\sim Q, \beta)$. If $\alpha = \beta$, $\langle \mathcal{A}, Q, \alpha \rangle$ and $\langle \mathcal{B}, \sim Q, \beta \rangle$ are a pair of blocking arguments; otherwise, one is a proper defeater for the other one and $rules(\Pi) \cup \{(Q, \alpha), (\sim Q, \beta)\} \vdash \perp$. Hence, by Def. 1, $\{(Q, \alpha), (\sim Q, \beta)\} \not\subseteq Warr$.

(iii) Suppose that, for some goal Q , $rules(\Pi) \cup Warr \vdash (Q, \alpha)$ and $(Q, \beta) \notin Warr$, for all $\beta \geq \alpha$. Then, there should exist a strict rule in Π of the form $(Q \leftarrow P_1 \wedge \dots \wedge P_k, 1)$ such that either for each $i = 1, \dots, k$, $(P_i, \alpha_i) \in Warr$ or, recursively, $rules(\Pi) \cup Warr \vdash (P_i, \alpha_i)$, and $\min(\alpha_1, \dots, \alpha_k) = \alpha$. Now, if $(Q, \alpha) \notin Warr$, by Def. 1, it follows that either $(Q, \beta) \in Warr$ or $(\sim Q, \beta) \in Warr$ for some $\beta > \alpha$, or $rules(\Pi) \cup Warr \vdash (\sim Q, \alpha)$. As $\alpha = \min(\alpha_1, \dots, \alpha_k)$, it follows that $\alpha = \alpha_i$, for some $1 \leq i \leq k$. Then, if $(\sim Q, \beta) \in Warr$, with $\beta > \alpha$, or $\Pi \cup Warr \vdash (\sim Q, \alpha)$, by Def. 1, there should exist, at least, a goal P_i , with $1 \leq i \leq k$, such that $(P_i, \alpha_i) \notin Warr$ and $rules(\Pi) \cup Warr \not\vdash (P_i, \alpha_i)$. Hence, if $rules(\Pi) \cup Warr \vdash (Q, \alpha)$, then $(Q, \beta) \in Warr$, for some $\beta \geq \alpha$. \square

Next we show that if $(Warr, Block)$ is an output of a P-DeLP program $\mathcal{P} = (\Pi, \Delta)$, the set $Warr$ of warranted goals contains indeed each literal Q satisfying that $\mathcal{P}^* \mid \sim^w \langle \mathcal{A}, Q, \alpha \rangle$ and $\Pi \cup \mathcal{A} \vdash (Q, \alpha)$, with $\mathcal{P}^* = (\Pi \cup Cl_{tp}(rules(\Pi)), \Delta)$ and whenever $\Pi \cup Cl_{tp}(rules(\Pi))$ is non-contradictory.

Proposition 4 *Let $\mathcal{P} = (\Pi, \Delta)$ be a P-DeLP program such that $\Pi \cup Cl_{tp}(rules(\Pi))$ is non-contradictory and let Q be a literal such that $\mathcal{P}^* \mid \sim^w \langle \mathcal{A}, Q, \alpha \rangle$. If $\Pi \cup \mathcal{A} \vdash (Q, \alpha)$, $(Q, \alpha) \in Warr$ for all output $(Warr, Block)$ of \mathcal{P} .⁴*

Notice that the inverse of Prop. 4 does not hold; i.e. assuming that $\Pi \cup Cl_{tp}(rules(\Pi))$ is non-contradictory it can be the case that $(Q, \alpha) \in Warr$ and (Q, α) is not warranted wrt the extended program \mathcal{P}^* . This is due to the fact that the new level-wise approach for computing warranted goals allows us to consider a more specific treatment of both direct and indirect conflicts between literals. In particular we have that each blocked literal invalidates all rules in which the literal occurs. For instance, consider the program $\mathcal{P}_1 = (\Pi_1, \Delta_1)$, with

⁴In what follows, proofs are omitted for space reasons

$$\begin{aligned}\Pi_1 &= \{(y, 1), (\sim y \leftarrow a \wedge b, 1)\} \text{ and} \\ \Delta_1 &= \{(a, 0.7), (b, 0.7), (\sim b, 0.7)\}.\end{aligned}$$

According to Def. 1, $(y, 1)$ and $(a, 0.7)$ are warranted and $(b, 0.7)$ and $(\sim b, 0.7)$ are blocked. However, when considering the extended program $\mathcal{P}_1^* = (\Pi_1 \cup Cl_{tp}(rules(\Pi_1)), \Delta_1)$ one is considering the transposed rule $(\sim a \leftarrow y \wedge b, 1)$ and therefore,

$$\lambda_1 = [\langle \{(a, 0.7)\}, a, 0.7 \rangle, \langle \{(b, 0.7)\}, \sim a, 0.7 \rangle]$$

is an acceptable argumentation line wrt \mathcal{P}_1^* with an even number of arguments, and thus, $(a, 0.7)$ is not warranted wrt \mathcal{P}_1^* . Another case that can be analyzed is the following one: Consider now the program $\mathcal{P}_2 = (\Pi_1, \Delta_2)$, with

$$\Delta_2 = \{(a, 0.7), (b \leftarrow a, 0.7), (\sim b, 0.7)\}.$$

According to Def. 1, $(y, 1)$ and $(a, 0.7)$ are warranted and, as indirect conflicts are not allowed, $\langle \mathcal{B}, b, 0.7 \rangle$ with $\mathcal{B} = \{(b \leftarrow a, 0.7), (a, 0.7)\}$ is not an acceptable argument for $(b, 0.7)$, and therefore $(\sim b, 0.7)$ is warranted. However, when considering the extended program $\mathcal{P}_2^* = (\Pi_1 \cup Cl_{tp}(rules(\Pi_1)), \Delta_2)$,

$$\lambda_2 = [\langle \{(\sim b, 0.7)\}, \sim b, 0.7 \rangle, \langle \mathcal{B}, b, 0.7 \rangle]$$

is an acceptable argumentation line wrt \mathcal{P}_2^* with an even number of arguments, and thus, $(\sim b, 0.7)$ is not warranted wrt \mathcal{P}_2^* .

Actually, the intuition underlying Def. 1 can be defined as follows: An argument $\langle \mathcal{A}, Q, \alpha \rangle$ is warranted or blocked if each subargument $\langle \mathcal{B}, R, \beta \rangle \sqsubseteq \langle \mathcal{A}, Q, \alpha \rangle$, with $Q \neq R$, is warranted. Then, it is warranted if it does not induce direct nor indirect conflicts and blocked, otherwise. The following results provide an interesting characterization of the relationship between warranted and blocked goals in a P-DeLP program.

Proposition 5 *Let $\mathcal{P} = (\Pi, \Delta)$ be a P-DeLP program and let $(Warr, Block)$ be an output for \mathcal{P} . Then:*

1. *If $(Q, \alpha) \in Warr \cup Block$, then there exists $\langle \mathcal{A}, Q, \alpha \rangle \in ARG(\mathcal{P})$ and, for each subargument $\langle \mathcal{B}, R, \beta \rangle \sqsubseteq \langle \mathcal{A}, Q, \alpha \rangle$ with $R \neq Q$, $(R, \beta) \in Warr$.*
2. *If $(Q, \alpha) \in Warr \cup Block$, then for every argument $\langle \mathcal{A}, Q, \beta \rangle$, with $\beta > \alpha$, there exists a subargument $\langle \mathcal{B}, R, \gamma \rangle \sqsubseteq \langle \mathcal{A}, Q, \beta \rangle$ with $R \neq Q$, such that $(R, \gamma) \notin Warr$.*
3. *If $(Q, \alpha) \in Warr$, there is no $\beta > 0$ such that $(Q, \beta) \in Block$ or $(\sim Q, \beta) \in Block$.*
4. *If $(Q, \alpha) \notin Warr \cup Block$ for each $\alpha > 0$, then either $(\sim Q, \beta) \in Block$ for some $\beta > 0$, or for each argument $\langle \mathcal{A}, Q, \alpha \rangle$, there exists a subargument $\langle \mathcal{B}, R, \beta \rangle \sqsubseteq \langle \mathcal{A}, Q, \alpha \rangle$ with $R \neq Q$, such that $(R, \beta) \notin Warr$, or $rules(\Pi) \cup Warr(\geq \alpha) \cup \{(Q, \alpha)\} \vdash \perp$.*

Finally, we will come to the question of whether a program \mathcal{P} always has a *unique* output $(Warr, Block)$ according to Def. 1. In general, the answer is yes, although we have identified some recursive situations that might lead to different outputs. For instance, consider the program

$$\mathcal{P} = \{(p, 0.9), (q, 0.9), (\sim p \leftarrow q, 0.9), (\sim q \leftarrow p, 0.9)\}.$$

Then, according to Def. 1, p is a warranted goal iff q and $\sim q$ are a pair of blocked goals and viceversa, q is a warranted goal iff p and $\sim p$ are a pair of blocked goals. Hence, in that case we have two possible outputs: $(Warr_1, Block_1)$ and $(Warr_2, Block_2)$ where

$$\begin{aligned} Warr_1 &= \{(p, 0.9)\}, Block_1 = \{(\sim q, 0.9), (q, 0.9)\} \\ Warr_2 &= \{(q, 0.9)\}, Block_2 = \{(p, 0.9), (\sim p, 0.9)\} \end{aligned}$$

In such a case, either p or q can be warranted goals (but just one of them).⁵ Thus, although our approach is skeptical, we can get sometimes alternative extensions for warranted beliefs. A natural solution for this problem would be adopting the intersection of all possible outputs in order to define the set of those literals which are ultimately warranted. Namely, Let \mathcal{P} be a P-DeLP program, and let $output_i(\mathcal{P}) = (Warr_i, Block_i)$ denote all possible outputs for \mathcal{P} , $i = 1 \dots n$. Then the *skeptical output* of \mathcal{P} could be defined as $output_{skep}(\mathcal{P}) = (\bigcap_{i=1 \dots n} Warr_i, \bigcap_{i=1 \dots n} Block_i)$. It can be shown that that $output_{skep}(\mathcal{P})$ satisfies by construction also Prop. 3 (indirect inconsistency). It remains as a future task to study the formal properties of this definition.

5. Related Work. Conclusions

We have presented a novel level-based approach to computing warranted arguments in P-DeLP. In order to do so, we have refined the notion of conflict among arguments, providing refined definitions of blocking and proper defeat. The resulting characterization allows to compute warranted goals in P-DeLP without making use of dialectical trees as underlying structures. More importantly, we have also shown that our approach ensures the satisfiability of the indirect consistency postulate proposed in [4,5], without requiring the use of transposed rules.

Assigning levels or grades to warranted knowledge has been source of research within the argumentation community in the last years, and to the best of our knowledge can be traced back to the notion of *degree of justification* addressed by John Pollock [10]. In this paper, Pollock concentrates on the “on sum” degree of justification of a conclusion in terms of the degrees of justification of all relevant premises and the strengths of all relevant reasons. However, his work is more focused on epistemological issues than ours, not addressing the problem of indirect inconsistency, nor using the combination of logic programming and possibilistic logic to model argumentative inference. An alternative direction is explored by Besnard & Hunter [2] by characterizing *aggregation functions* such as categorisers and accumulators which allow to define more evolved forms of computing warrant (e.g. counting arguments for and against, etc.). However, this research does not address the problem of indirect consistency, and performs the grading on top of a classical first-order language, where clauses are weighed as in our case. More recently, the research work of Cayrol & Lagasque-Schiex [6] pursues a more ambitious goal, providing a general framework for formalizing the notion of *graduality* in valuation models for argumentation frameworks, focusing on the valuation of arguments and

⁵A complete characterization of these pathological situations is a matter of current research.

the acceptability according to different semantics. The problem of indirect consistency is not addressed here either, and the underlying system is Dung’s abstract argumentation systems, rather than a logic programming framework as in our case.

We contend that our level-based characterization of warrant can be extended to other alternative argumentation frameworks in which weighted clauses are used for knowledge representation. Part of our current research is focused on finding a suitable generalization for capturing the results presented in this paper beyond the P-DeLP framework.

Acknowledgments

This research was partially supported by CICYT Projects MULOLOG (TIN2004-07933-C03-01/03) and IEA (TIN2006-15662-C02-01/02), by CONICET (Argentina), and by the Secretaría General de Ciencia y Tecnología de la Universidad Nacional del Sur (Project PGI 24/ZN10).

References

- [1] 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*, 2008 (to appear). Preliminary manuscript available from http://cs.uns.edu.ar/~cic/2007/2007_fss.pdf.
- [2] P. Besnard and A. Hunter. A logic-based theory of deductive arguments. *Artif. Intell.*, 128(1-2):203–235, 2001.
- [3] R. Brena, J. Aguirre, C. Chesñevar, E. Ramírez, and L. Garrido. Knowledge and information distribution leveraged by intelligent agents. *Knowl. Inf. Syst.*, 12(2):203–227, 2007.
- [4] M. Caminada and L. Amgoud. An axiomatic account of formal argumentation. In *Proc. of the 20th AAI Conference, Pittsburgh, Pennsylvania, USA*, pages 608–613. AAAI Press AAAI Press / The MIT Press, 2005.
- [5] M. Caminada and L. Amgoud. On the evaluation of argumentation formalisms. *Artif. Intell.*, 171(5-6):286–310, 2007.
- [6] C. Cayrol and M. Lagasque-Schiex. Graduality in argumentation. *J. Artif. Intell. Res. (JAIR)*, 23:245–297, 2005.
- [7] C. Chesñevar, A. Maguitman, and R. Loui. Logical Models of Argument. *ACM Computing Surveys*, 32(4):337–383, December 2000.
- [8] C. Chesñevar, A. Maguitman, and G. Simari. Argument-Based Critics and Recommenders: A Qualitative Perspective on User Support Systems. *Journal of Data and Knowledge Engineering*, 59(2):293–319, 2006.
- [9] A. García and G. Simari. Defeasible Logic Programming: An Argumentative Approach. *Theory and Practice of Logic Programming*, 4(1):95–138, 2004.
- [10] J. Pollock. Defeasible reasoning with variable degrees of justification. *Artif. Intell.*, 133(1-2):233–282, 2001.
- [11] H. Prakken and G. Sartor. Argument-based extended logic programming with defeasible priorities. *Journal of Applied Non-classical Logics*, 7:25–75, 1997.
- [12] 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.