215

On the Characterization of the Maximal Ideal Recursive Semantics of RP-DeLP

Teresa Alsinet [a,1], Ramón Béjar [a] Lluís Godo [b] and Francesc Guitart [a]

[a] *Department of Computer Science. University of Lleida,* SPAIN
[b] *Artificial Intelligence Research Institute (IIIA-CSIC), Bellaterra,* SPAIN

Abstract. Possibilistic Defeasible Logic Programming (P-DeLP) is a logic programming framework which combines features from argumentation theory and logic programming, in which defeasible rules are attached with weights expressing their relative belief or preference strength. In P-DeLP a conclusion succeeds if there exists an argument that entails the conclusion and this argument is found to be undefeated by a warrant procedure that systematically explores the universe of arguments in order to present an exhaustive synthesis of the relevant chains of pros and cons for the given conclusion. Recently, we have proposed a new warrant recursive semantics for P-DeLP, called *Recursive* P-DeLP (RP-DeLP for short), based on the claim that the acceptance of an argument should imply also the acceptance of all its subarguments which reflect the different premises on which the argument is based. In RP-DeLP, an output of a program is a pair of sets, a set of warranted and a set of blocked conclusions.Arguments for both warranted and blocked conclusions are recursively based on warranted conclusions but, while warranted conclusions do not generate any conflict with the set of already warranted conclusions and the strict part of program (information we take for granted they hold true), blocked conclusions do. Conclusions that are neither warranted nor blocked correspond to rejected conclusions. This paper explores the relationship between the exhaustive dialectical analysis based semantics of P-DeLP and the recursive based semantics of RP-DeLP, and analyzes a non-monotonic inference operator for RP-DeLP which models the expansion of a given program by adding new weighed facts associated with warranted conclusions.

Keywords. argumentation, logic programming, uncertainty, non-monotonic inference

1. Introduction and motivations

Defeasible argumentation is a natural way of identifying relevant assumptions and conclusions for a given problem which often involves identifying conflicting information, resulting in the need to look for pros and cons for a particular conclusion [10]. This process may involve chains of reasoning, where conclusions are used in the assumptions for deriving further conclusions and the task of finding pros and cons may be decomposed recursively. Logic-based formalizations of argumentation that take pros and cons for some conclusion into account assume a set of formulas and then lay out arguments and counterarguments that can be obtained from these assumed formulas [4].

---

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

Defeasible Logic Programming (DeLP) [8] is a formalism that combines techniques of both logic programming and defeasible argumentation. As in logic programming, knowledge is represented in DeLP using facts and rules; however, DeLP also provides the possibility of representing defeasible knowledge under the form of weak (defeasible) rules, expressing reasons to believe in a given conclusion.

In DeLP, a conclusion succeeds if it is warranted, i.e., if there exists an argument (a consistent sets of defeasible rules) that, together with the non-defeasible rules and facts, entails the conclusion, and moreover, this argument is found to be undefeated by a warrant procedure which builds a dialectical tree containing all arguments that challenge this argument, and all counterarguments that challenge those arguments, and so on, recursively. Actually, dialectical trees systematically explore the universe of arguments in order to present an exhaustive synthesis of the relevant chains of pros and cons for a given conclusion. In fact, the interpreter for DeLP [7] (http://lidia.cs.uns.edu.ar/DeLP) takes a knowledge base (program) P and a conclusion (query) Q as input, and it then returns one of the following four possible answers: YES, if Q is warranted from P; NO, if the complement of Q is warranted from P; UNDECIDED, if neither Q nor its complement are warranted from P; or UNKNOWN, if Q is not in the language of the program P.

Possibilistic Defeasible Logic Programming (P-DeLP) [2] is an extension of DeLP in which defeasible facts and rules are attached with weights (belonging to the real unit interval $[0, 1]$) expressing their relative belief or preference strength. As many other argumentation frameworks [6,10], 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 (to some necessity degree), that is, those which remain undefeated after an exhaustive dialectical analysis of all possible arguments for and against.

Recently in [1], we have proposed a new semantics for P-DeLP based on a general notion of collective (non-binary) conflict among arguments and on the claim that the acceptance of an argument should imply also the acceptance of all its subarguments which reflect the different premises on which the argument is based. In this framework, called *Recursive* P-DeLP (RP-DeLP for short), an output (extension) of a program is now a pair of sets, a set of warranted and a set of blocked conclusions, with maximum necessity degrees. Arguments for both warranted and blocked conclusions are recursively based on warranted conclusions but, while warranted conclusions do not generate any conflict with the set of already warranted conclusions and the strict part of program (information we take for granted they hold true), blocked conclusions do. Conclusions that are neither warranted nor blocked correspond to rejected conclusions.

The key feature that our warrant recursive semantics addresses corresponds with the *closure under subarguments postulate* recently proposed by Amgoud [3], claiming that if an argument is excluded from an output, then all the arguments built on top of it should also be excluded from that output. As stated in [9], this recursive definition of acceptance among arguments can lead to different outputs (extensions) for warranted conclusions. For RP-DeLP programs with multiple outputs we have also considered in [1] the problem of deciding the set of conclusions that could be ultimately warranted. We have called this output (extension) *maximal ideal output* of an RP-DeLP program.

In this paper we explore the relationship between the exhaustive dialectical analysis based semantics of P-DeLP and the maximal ideal output of RP-DeLP, and we analyze a

non-monotonic inference operator for RP-DeLP which models the expansion of a given program by adding new weighed facts associated with warranted conclusions.

2. The language of P-DeLP and RP-DeLP

In order to make this paper self-contained, we will present next the main definitions that characterize P-DeLP and RP-DeLP frameworks. For details the reader is referred to [2,1].

The *language* of P-DeLP and RP-DeLP, denoted $\mathcal{L}$, 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 $\mathcal{L}$ 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.

P-DeLP and RP-DeLP frameworks are based on the propositional logic $(\mathcal{L}, \vdash)$ where the inference operator $\vdash$ is defined by instances of the modus ponens rule of the form: $\{Q \leftarrow P_1 \wedge \ldots \wedge P_k, P_1, \ldots, P_k\} \vdash Q$. A set of clauses Γ will be deemed as *contradictory*, denoted $\Gamma \vdash \bot$, if , for some atom q, $\Gamma \vdash q$ and $\Gamma \vdash \sim q$.

In both frameworks a *program* $\mathcal{P}$ is a tuple $\mathcal{P} = (\Pi, \Delta, \preceq)$ over the logic $(\mathcal{L}, \vdash)$, where $\Pi, \Delta \subseteq \mathcal{L}$, and $\Pi \not\vdash \bot$. Π is a finite set of clauses representing strict knowledge (information we take for granted they hold true), Δ is another finite set of clauses representing the defeasible knowledge (formulas for which we have reasons to believe they are true). Finally, $\preceq$ is a total pre-order on $\Pi \cup \Delta$ representing levels of defeasibility: $\varphi \prec \psi$ means that φ is more defeasible than ψ. Actually, since formulas in Π are not defeasible, $\preceq$ is such that all formulas in Π are at the top of the ordering. For the sake of a simpler notation we will often refer in the paper to numerical levels for defeasible clauses and arguments rather than to the pre-ordering $\preceq$, so we will assume a mapping $N: \Pi \cup \Delta \rightarrow [0, 1]$ such that $N(\varphi) = 1$ for all $\varphi \in \Pi$ and $N(\varphi) < N(\psi)$ iff $\varphi \prec \psi$. [1]

The notion of *argument* is the usual one inherited from similar definitions in the argumentation literature [11,10,6]. Given a program $\mathcal{P}$, an argument for a literal (conclusion) Q of $\mathcal{L}$ is a pair $\mathcal{A} = \langle A, Q \rangle$, with $A \subseteq \Delta$ such that $\Pi \cup A \not\vdash \bot$, and A is minimal (w.r.t. set inclusion) such that $\Pi \cup A \vdash Q$. 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). We define the *strength of an argument* $\langle A, Q \rangle$, written $s(\langle A, Q \rangle)$, as follows: $s(\langle A, Q \rangle) = 1$ if $A = \emptyset$, and $s(\langle A, Q \rangle) = \min\{N(\psi) \mid \psi \in A\}$, otherwise.

The notion of *subargument* is referred to d-arguments and expresses an incremental proof relationship between arguments which is defined as follows. Let $\langle B, Q \rangle$ and $\langle A, P \rangle$ be two d-arguments such that the minimal sets (w.r.t. set inclusion) $\Pi_Q \subseteq \Pi$ and $\Pi_P \subseteq \Pi$ such that $\Pi_Q \cup B \vdash Q$ and $\Pi_P \cup A \vdash P$ verify that $\Pi_Q \subseteq \Pi_P$. Then, $\langle B, Q \rangle$ is a *subargument* of $\langle A, P \rangle$, written $\langle B, Q \rangle \sqsubset \langle A, P \rangle$, when either $B \subset A$ (strict inclusion for defeasible knowledge), or $B = A$ and $\Pi_Q \subset \Pi_A$ (strict inclusion for strict knowl-

---

[1]Actually, a same pre-order $\preceq$ can be represented by many mappings, but we can take any of them to since only the relative ordering is what actually matters.

edge). A literal Q of $\mathcal{L}$ is called *justifiable conclusion* w.r.t. $\mathcal{P}$ if there exists an argument for Q, i.e. there exists $A \subseteq \Delta$ such that $\langle A, Q \rangle$ is an argument.

As in most argumentation formalisms (see e.g. [10,6]), in P-DeLP and RP-DeLP frameworks it can be the case that there exist arguments supporting contradictory literals, and thus, there exist sets of *conflicting* arguments. Since arguments can rely on defeasible information, conflicts among arguments may be resolved in both frameworks by comparing their strength. In this sense the aim of both frameworks is to provide a useful warrant procedure in order to determine which conclusions are ultimately accepted (or *warranted*) on the basis of a given program. The difference between the two frameworks lies in the way in which this procedure is defined and the type of conflicts are handled.

In P-DeLP warranted conclusions are justifiable conclusions which remain undefeated after an exhaustive dialectical analysis of all possible arguments for an against and only *binary* attacks or defeat relations are considered. In RP-DeLP semantics for warranted conclusions is based on a *collective* (non-binary) notion of conflict between arguments and if an argument is excluded from an output, then all the arguments built on top of it are excluded from that output. In the following sections we describe both mechanisms.

3. Warrant semantics of P-DeLP

Let $\mathcal{P}$ be a P-DeLP program, and let $\langle A_1, Q_1 \rangle$ and $\langle A_2, Q_2 \rangle$ be two arguments w.r.t. $\mathcal{P}$. $\langle A_1, Q_1 \rangle$ *defeats* $\langle A_2, Q_2 \rangle$[2] iff $Q_1 = \sim Q_2$ and $s(\langle A_1, Q_1 \rangle) \geq s(\langle A_2, Q_2 \rangle)$, or $\langle A, Q \rangle \sqsubset \langle A_2, Q_2 \rangle$ and $Q_1 = \sim Q$ and $s(\langle A_1, Q_1 \rangle) \geq s(\langle A, Q \rangle)$. Moreover, if $\langle A_1, Q_1 \rangle$ defeats $\langle A_2, Q_2 \rangle$ with strict relation $>$ we say that $\langle A_1, Q_1 \rangle$ is a *proper defeater* for $\langle A_2, Q_2 \rangle$, otherwise we say that $\langle A_1, Q_1 \rangle$ is a *blocking defeater* for $\langle A_2, Q_2 \rangle$.

In P-DeLP *warranted* conclusions are 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 $\langle A_0, Q_0 \rangle$ is a sequence of arguments $\lambda = [\langle A_0, Q_0 \rangle, \langle A_1, Q_1 \rangle, \ldots, \langle A_n, Q_n \rangle, \ldots]$ such that each $\langle A_i, Q_i \rangle$ defeats the previous argument $\langle A_{i-1}, Q_{i-1} \rangle$ 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* w.r.t. $\mathcal{P}$. [3]
2. **Progressive argumentation:** (i) every blocking defeater $\langle A_i, Q_i \rangle$ in λ with $i > 0$ is defeated by a proper defeater[4] $\langle A_{i+1}, Q_{i+1} \rangle$ in λ; and (ii) each argument $\langle A_i, Q_i \rangle$ in λ, with $i \geq 2$, is such that $Q_i \neq \sim Q_{i-1}$.

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

---

[2]In 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 $Q \equiv \sim q$.

[3]Non-contradiction for a set of arguments is defined as follows: a set $S = \bigcup_{i=1}^{n} \{\langle A_i, Q_i \rangle\}$ is *contradictory* w.r.t. $\mathcal{P}$ iff $\Pi \cup \bigcup_{i=1}^{n} A_i$ is contradictory.

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

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 $\mathcal{P}$ and a goal Q, Q is *warranted* w.r.t. $\mathcal{P}$ with *maximum strength* α iff there exists an argument $\langle A, Q \rangle$ with $s(\langle A, Q \rangle) = \alpha$ such that: i) every acceptable argumentation line starting with $\langle A, Q \rangle$ has an odd number of arguments; and ii) there is no other argument of the form $\langle B, Q \rangle$, with $s(\langle B, Q \rangle) > \alpha$, satisfying the above condition. In the rest of the paper we will write $\mathcal{P} \mid\sim^w \langle A, Q, \alpha \rangle$ to denote this fact and we will write $\mathcal{C}^w_{\mathrm{DT}}(\mathcal{P})$ to denote the set of warranted conclusions of $\mathcal{P}$ based on dialectical trees, i.e. $\mathcal{C}^w_{\mathrm{DT}}(\mathcal{P}) = \{Q \mid \mathcal{P} \mid\sim^w \langle A, Q, \alpha \rangle\}$.

In [5] Caminada and Amgoud proposed three *rationality postulates* which every rule-based argumentation system should satisfy. One of such postulates (called *Indirect Consistency*) requires that the set of warranted conclusions must be consistent (w.r.t. the underlying logic) with the set of strict clauses. This means that the warrant semantics of P-DeLP satisfies the indirect consistency postulate iff given a program $\mathcal{P} = (\Pi, \Delta, \preceq)$ its set of warranted conclusions $\mathcal{C}^w_{\mathrm{DT}}(\mathcal{P})$ is such that $\Pi \cup \mathcal{C}^w_{\mathrm{DT}}(\mathcal{P}) \not\vdash \perp$.

The defeat relation in P-DeLP, as occurs in most rule-based argumentation systems, is binary and, in some cases, the conflict relation among arguments is hardly representable as a binary relation when we compare them with the strict part of a program. For instance, consider the following program $\mathcal{P} = (\Pi, \Delta, \preceq)$ with $\Pi = \{p, \neg p \leftarrow a \wedge b \wedge c\}$, $\Delta = \{a, b, c\}$ and a single defeasibility level α for Δ. Clearly, $\mathcal{A}_1 = \langle \{a\}, a \rangle$, $\mathcal{A}_2 = \langle \{b\}, b \rangle$ and $\mathcal{A}_3 = \langle \{c\}, c \rangle$ are arguments that justify conclusions a, b and c respectively, and $\mathcal{A}_1$, $\mathcal{A}_2$ and $\mathcal{A}_3$ have no defeaters, and thus, $\{a, b, c\}$ are warranted w.r.t. the P-DeLP program $\mathcal{P}$. Indeed, conclusions a, b and c do not pairwisely generate a conflict since $\Pi \cup \{a, b\} \not\vdash \perp$, $\Pi \cup \{a, c\} \not\vdash \perp$ and $\Pi \cup \{b, c\} \not\vdash \perp$. However, these conclusions are collectively conflicting w.r.t. the strict part of program Π since $\Pi \cup \{a, b, c\} \vdash \perp$, and thus, the warrant semantics of P-DeLP does not satisfy the indirect consistency postulate.

In order to characterize such situations we proposed in [1] the RP-DeLP framework, a new warrant semantics for P-DeLP based on a general notion of collective (non-binary) conflict among arguments ensuring the three rationality postulates defined by Caminada and Amgoud.

4. Warrant semantics of RP-DeLP

The warrant recursive semantics of RP-DeLP is based on the following general notion of collective conflict in a set of arguments which captures the idea of an inconsistency arising from a consistent set of justifiable conclusions W together with the strict part of a program and the set of conclusions of those arguments. Let $\mathcal{P} = (\Pi, \Delta, \preceq)$ be a program and let $W \subseteq \mathcal{L}$ be a set of conclusions. We say that a set of arguments $\{\langle A_1, Q_1 \rangle, \ldots, \langle A_k, Q_k \rangle\}$ *minimally conflicts* with respect to W iff the two following conditions hold: (i) the set of argument conclusions $\{Q_1, \ldots, Q_k\}$ is contradictory with respect to W, i.e. it holds that $\Pi \cup W \cup \{Q_1, \ldots, Q_k\} \vdash \perp$; and (ii) the set $\{\langle A_1, Q_1 \rangle, \ldots, \langle A_k, Q_k \rangle\}$ is minimal with respect to set inclusion satisfying (i), i.e. if $S \subsetneq \{Q_1, \ldots, Q_k\}$, then $\Pi \cup W \cup S \not\vdash \perp$.

This general notion of conflict is used to define an *output of an RP-DeLP program* $\mathcal{P} = (\Pi, \Delta, \preceq)$ as a pair (*Warr*, *Block*) of subsets of $\mathcal{L}$ of warranted and blocked conclu-

sions respectively. 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 $\mathcal{P}$, we define: $Warr = Warr(1) \cup \{\cup_{i=1,p} Warr(\alpha_i)\}$ and $Block = \cup_{i=1,p} Block(\alpha_i)$, where $Warr(1) = \{Q \mid \Pi \vdash Q\}$, and $Warr(\alpha_i)$ and $Block(\alpha_i)$ are respectively the sets of the warranted and blocked justifiable conclusions of strength α_i. Intuitively, an argument $\langle A, Q \rangle$ of strength α_i is *valid* whenever (i) it is based on warranted conclusions; (ii) there does not exist a valid argument for Q with strength $> \alpha_i$; and (iii) Q is consistent with warranted and blocked conclusions of strength $> \alpha_i$. Then, a *valid* argument $\langle A, Q \rangle$ becomes *blocked* as soon as it leads to some conflict among valid arguments of same strength and the set of already warranted conclusions, otherwise, it is *warranted*.

In [1] we show that, in case of some circular dependences among arguments, the output of an RP-DeLP program may be not unique, that is, there may exist several pairs (*Warr*, *Block*) satisfying the above conditions for a given RP-DeLP program. The following example shows a circular relation among arguments involving strict knowledge. Consider the RP-DeLP program $\mathcal{P} = (\Pi, \Delta, \preceq)$ with $\Pi = \{y, \sim y \leftarrow p \wedge r, \sim y \leftarrow q \wedge s\}$, $\Delta = \{p, q, r \leftarrow q, s \leftarrow p\}$ and a single defeasibility level α for Δ. Then, $Warr(1) = \{y\}$ and $\mathcal{A}_1 = \langle \{p\}, p \rangle$ and $\mathcal{A}_2 = \langle \{q\}, q \rangle$ are valid arguments for conclusions p and q, respectively, and thus, conclusions p and q may be warranted or blocked but not rejected. Moreover, since arguments $\mathcal{B}_1 = \langle \{q, r \leftarrow q\}, r \rangle$ and $\mathcal{B}_2 = \langle \{p, s \leftarrow p\}, s \rangle$ are valid whenever q and p are warranted, respectively, and $\Pi \cup \{p, r\} \vdash \perp$ and $\Pi \cup \{q, s\} \vdash \perp$, we get that p can be warranted iff q is blocked and that q can be warranted iff p is blocked. Hence, in that case we have two possible outputs: $(Warr_1, Block_1)$ and $(Warr_2, Block_2)$, where $Warr_1 = \{y, p\}, Block_1 = \{q, s\}$ and $Warr_2 = \{y, q\}, Block_2 = \{p, r\}$. Figure 1 shows the circular dependences among $\{\mathcal{A}_1, \mathcal{A}_2\}$ and $\{\mathcal{B}_1, \mathcal{B}_2\}$. Conflict and support dependencies among arguments 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 r; (2) the support of r depends on q (i.e., r is valid whenever q is warranted); (3) the warranty of q depends on a (possible) conflict with s; and (4) the support of s depends on p (i.e., s is valid whenever p is warranted).
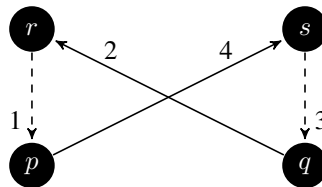

Figure 1. Circular dependences for program $\mathcal{P}$.

In [1] we analyze the problem of deciding the set of conclusions that can be ultimately warranted in RP-DeLP programs with multiple outputs. The usual skeptical approach would be to adopt the intersection of all possible outputs. However, in addition to the computational limitation, as stated in [9], adopting the intersection of all outputs

may lead to an inconsistent output in the sense of violating the base of the underlying recursive warrant semantics, claiming that if an argument is excluded from an output, then all the arguments built on top of it should also be excluded from that output. Intuitively, for a conclusion, to be in the intersection does not guarantee the existence of an argument for it that is recursively based on ultimately warranted conclusions. Instead, the set of ultimately warranted conclusions we are interested in for RP-DeLP programs is characterized by means of a recursive level-wise definition considering at each level the maximum set of conclusions based on warranted information and not involved in neither a conflict nor a circular definition of warranty. We refer to this output as *maximal ideal output* of an RP-DeLP program.

Intuitively, a valid argument $\langle A, Q \rangle$ becomes blocked in the *maximal ideal output*, as soon as (i) it leads to some conflict among valid arguments of same strength and the set of already warranted conclusions or (ii) the warranty of $\langle A, Q \rangle$ depends on some circular definition of conflict between arguments of same strength; otherwise, it is warranted. Consider again the previous program $\mathcal{P}$. According to Figure 1, valid arguments for conclusions p and q are involved in a circular circular definition of conflict, and thus, conclusions p and q must be blocked in the maximal ideal output of $\mathcal{P}$ and arguments for conclusions r and s are rejected. Hence, in that case, we have the following maximal ideal output of $\mathcal{P}$: $(Warr_{\max}, Block_{\max})$, where $Warr_{\max} = \{y\}$, $Block_{\max} = \{p, q\}$.

5. Dialectical analysis and maximal ideal output

In [1] we prove that the maximal ideal output of an RP-DeLP program is unique and satisfies the indirect consistency property defined by Caminada and Amgoud with respect to the strict knowledge.

Next we show that given a program $\mathcal{P}$ and its set $\mathcal{C}_{\mathrm{DT}}^{w}(\mathcal{P})$ of warranted conclusions based on dialectical trees, $\mathcal{C}_{\mathrm{DT}}^{w}(\mathcal{P})$ contains each warranted conclusion in the maximal ideal output of $\mathcal{P}$. [5]

Proposition 1 *Let* $\mathcal{P} = (\Pi, \Delta, \preceq)$ *be a program with levels of defeasibility* $1 > \alpha_1 > \ldots > \alpha_p \geq 0$. *If* $(Warr, Block)$ *is the maximal ideal output of* $\mathcal{P}$, *for each level* α_i *it holds that* $Warr(\alpha_i) \subseteq \{Q \mid \mathcal{P} \mid\sim^w \langle A, Q, \alpha_i \rangle\}$. *Obviously,* $Warr(1) = \{Q \mid \mathcal{P} \mid\sim^w \langle A, Q, 1 \rangle\}$.

Notice that the inverse of Prop. 1 does not hold since the dialectical analysis based semantics of P-DeLP does not satisfy the indirect consistency property defined by Caminada and Amgoud with respect to the strict knowledge. Because we are interested in exploring the relationship between the warrant semantics of P-DeLP and the maximal ideal output of RP-DeLP, we have to extend the P-DeLP framework with some mechanism ensuring this property.

In [5] Caminada and Amgoud propose 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 $r = \phi_1, \phi_2, \ldots, \phi_n \rightarrow \psi$ as a material implication in propositional logic which is equivalent to the disjunction $\phi_1 \vee \phi_2 \vee \ldots, \phi_n \vee \neg\psi$. From that disjunction different rules of the form $\phi_1, \ldots, \phi_{i-1}, \neg\psi, \phi_{i+1}, \ldots, \phi_n \rightarrow \neg\phi_i$ can be obtained

---

[5]Proofs are not included in the paper for space reasons.

(*transpositions* of r). If $\mathcal{S}$ is a set of strict rules, Cl_{tp} is the minimal set such that (i) $\mathcal{S} \subseteq Cl_{tp}(\mathcal{S})$ and (ii) If $s \in Cl_{tp}(\mathcal{S})$ and t is a transposition of s, then $t \in Cl_{tp}(\mathcal{S})$.

Computing the closure under transposition of strict rules allows the indirect consistency property to be satisfied in the case of rule-based argumentation systems like DeLP or P-DeLP as it was proved in [5]. In fact, in some sense, it allows to perform *forward reasoning* from warranted conclusions, and thus, to evaluate collective conflicts among arguments. However, P-DeLP is a *Horn*-based system, 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 $\mathcal{P} = (\Pi, \Delta, \preceq)$ with $\Pi = \{q \leftarrow p \wedge r, s \leftarrow \sim r, p, \sim q, \sim s\}$, $\Delta = \emptyset$. In P-DeLP, $p, \sim q$ and $\sim s$ would be warranted conclusions, i.e. $\mathcal{C}_{\text{DT}}^{w}(\mathcal{P}) = \{p, \sim q, \sim s\}$. However, the closure under transposition $Cl_{tp}(\Pi)$ would include the rule $\sim r \leftarrow p \wedge \sim q$, resulting in inconsistency since both s and $\sim s$ can be derived, so that the whole program would be deemed as invalid.

Apart from the above limitation, when extending a P-DeLP program with all possible transpositions of every strict rule, the system can possibly establish as warranted goals conclusions which are not explicitly expressed in the original program. Consider e.g. the program $\mathcal{P} = (\Pi, \Delta, \preceq)$ with $\Pi = \{\sim y \leftarrow a \wedge b, y\}$, $\Delta = \{a, b\}$ and two levels of defeasibility for Δ as follows: $\{b\} \prec \{a\}$. Assume α_1 is the level of $\{a\}$ and α_2 is the level of $\{b\}$, with $1 > \alpha_1 > \alpha_2 > 0$. Transpositions of the strict rule $\sim y \leftarrow a \wedge b$ are $\sim a \leftarrow y \wedge b$ and $\sim b \leftarrow y \wedge a$. Then, the argument $\mathcal{A} = \langle\{\sim b \leftarrow a \wedge y, a\}, \sim b\rangle$ with strength α_1 justifies conclusion $\sim b$. Moreover, as there is neither a proper nor a blocking defeater of $\mathcal{A}$, we conclude that $\sim b$ is warranted w.r.t. $\mathcal{P}^{\star} = (\Pi \cup Cl_{tp}(\Pi), \Delta, \preceq)$, although no explicit information is given for literal $\sim b$ in $\mathcal{P}$. Moreover, notice that $\mathcal{C}_{\text{DT}}^{w}(\mathcal{P}) = \{y, a, b\}$ and $\mathcal{C}_{\text{DT}}^{w}(\mathcal{P}^{\star}) = \{y, a, \sim b\}$.

Next we show that if (*Warr*, *Block*) is the maximal ideal output of a program $\mathcal{P} = (\Pi, \Delta, \preceq)$, the set *Warr* of warranted conclusions contains indeed each literal Q satisfying that $\mathcal{P}^{\star} \mid\sim^{w} \langle A, Q, \alpha\rangle$ and $\Pi \cup A \vdash Q$, with $\mathcal{P}^{\star} = (\Pi \cup Cl_{tp}(\Pi), \Delta, \preceq)$ and whenever $\Pi \cup Cl_{tp}(\Pi) \not\vdash \bot$.

Proposition 2 *Let* $\mathcal{P} = (\Pi, \Delta, \preceq)$ *be a program with levels of defeasibility* $1 > \alpha_1 > \ldots > \alpha_p \geq 0$ *and such that* $\Pi \cup Cl_{tp}(\Pi) \not\vdash \bot$. *If* (*Warr, Block*) *is the maximal ideal output of* $\mathcal{P}$ *and* $\mathcal{P}^{\star} = (\Pi \cup Cl_{tp}(\Pi), \Delta, \preceq)$, *for each level* α_i *it holds that* $Warr(\alpha_i) = \{Q \mid \mathcal{P}^{\star} \mid\sim^{w} \langle A, Q, \alpha_i\rangle$ *and* $\Pi \cup A \vdash Q\}$. *Obviously,* $Warr(1) = \{Q \mid \Pi \vdash Q\} = \{Q \mid \mathcal{P}^{\star} \mid\sim^{w} \langle A, Q, 1\rangle$ *and* $\Pi \cup A \vdash Q\}$.

Following the approach we made in [2] for dialectical semantics, next we study the behavior of the maximal ideal output of an RP-DeLP program in the context of non-monotonic inference relationships. In order to do this, we define an inference operator $\mathcal{E}_{\text{RS}}^{w}$ that computes the expansion of a program including all new facts which correspond to warranted conclusions in the maximal ideal output.

Formally: Let $\mathcal{P} = (\Pi, \Delta, \preceq)$ be an RP-DeLP program with levels of defeasibility $1 > \alpha_1 > \ldots > \alpha_p \geq 0$ and let (*Warr*, *Block*) be the maximal ideal output of $\mathcal{P}$. We define the operator $\mathcal{E}_{\text{RS}}^{w}$ associated with $\mathcal{P}$ as follows: $\mathcal{E}_{\text{RS}}^{w}(\mathcal{P}) = (\Pi \cup Warr(1), \Delta \cup (\cup_{i=1\ldots p} Warr(\alpha_i)), \preceq')$ and such that $N'(\varphi) = N(\varphi)$ for all $\varphi \in \Pi \cup \Delta$, $N'(\varphi) = 1$ for all $\varphi \in Warr(1)$, and $N(\varphi) = \alpha_i$ for all $\varphi \in Warr(\alpha_i)$, $i = 1 \ldots p$.

Notice that by definition operator $\mathcal{E}_{\text{RS}}^{w}$ is *well-defined* (i.e., given an RP-DeLP program as input, the associated output is also an RP-DeLP program). Moreover, $\mathcal{E}_{\text{RS}}^{w}$ sat-

isfies *inclusion*: given an RP-DeLP program $\mathcal{P} = (\Pi, \Delta, \preceq)$ with levels of defeasibility $1 > \alpha_1 > \ldots > \alpha_p \geq 0$ and maximal ideal output (*Warr, Block*), $\Pi \subseteq \Pi \cup Warr(1)$, $\Delta \subseteq \Delta \cup (\cup_{i=1\ldots p} Warr(\alpha_i))$ and $\preceq'$ preserves the total pre-order $\preceq$ on $\Pi \cup \Delta$.

In what follows, given an RP-DeLP program $\mathcal{P} = (\Pi, \Delta, \preceq)$, a clause φ and a set of clauses Γ, we will write $\varphi \in \mathcal{P}$ and $\Gamma \subseteq \mathcal{P}$ to denote that $\varphi \in \Pi \cup \Delta$ and $\Gamma \subseteq \Pi \cup \Delta$, respectively.

Besides, *monotonicity* does not hold for $\mathcal{E}_{RS}^w$, as expected. It is satisfied if all warranted conclusions from a given program are preserved when the program is augmented with new clauses. As a counterexample consider the program $\mathcal{P} = (\Pi, \Delta, \preceq)$ with $\Pi = \{q\}$, $\Delta = \{p \leftarrow q\}$ and a single level of defeasibility α for Δ. Then, $Warr(1) = \{q\}$ and $Warr(\alpha) = \{p\}$, and thus, $\{q, p\} \subseteq \mathcal{E}_{RS}^w(\mathcal{P})$. However, if we extend program $\mathcal{P}$ with the strict fact $\sim p$, we get the following program $\mathcal{P}' = (\Pi', \Delta, \preceq')$ with $\Pi' = \{q, \sim p\}$ and $N'(\sim p) = 1$. Then, $Warr(1) = \{q, \sim p\}$ and $Warr(\alpha) = \emptyset$ in the maximal ideal output of $\mathcal{P}'$. Hence, $p \notin \mathcal{E}_{RS}^w(\mathcal{P}')$ but $p \in \mathcal{E}_{RS}^w(\mathcal{P})$.

Semi-monotonicity is an interesting property for analyzing non-monotonic consequence relationships. It is satisfied if all defeasible warranted conclusions are preserved when the program is augmented with new *defeasible* clauses. *Semi-monotonicity* does not hold for $\mathcal{E}_{RS}^w$, as adding new defeasible clauses cannot invalidate already valid arguments, but it can enable new ones that were not present before, thus introducing new conflicts or new circular dependences among arguments. Arguments that were warranted may therefore no longer keep that status. Consider a variant of the previous counterexample: we consider the fact $\sim p$ as defeasible information, i.e. we define the following program $\mathcal{P}' = (\Pi, \Delta', \preceq')$ with $\Delta' = \{p \leftarrow q, \sim p\}$ $N'(\sim p) = N'(p \leftarrow q)$. Now, $Warr(1) = \{q\}$, $Warr(\alpha) = \emptyset$ and $Block(\alpha) = \{p, \sim p\}$ for the maximal ideal output of $\mathcal{P}'$. Hence, $p \notin \mathcal{E}_{RS}^w(\mathcal{P}')$ but $p \in \mathcal{E}_{RS}^w(\mathcal{P})$.

Next we define some relevant logical properties that operator $\mathcal{E}_{RS}^w$ satisfies.

- The operator $\mathcal{E}_{RS}^w$ satisfies *idempotence*: $\mathcal{E}_{RS}^w(\mathcal{P}) = \mathcal{E}_{RS}^w(\mathcal{E}_{RS}^w(\mathcal{P}))$.
- The operator $\mathcal{E}_{RS}^w$ satisfies *cummulativity*: if $Q \in \mathcal{E}_{RS}^w(\mathcal{P})$, then if $R \in \mathcal{E}_{RS}^w(\mathcal{P} \cup \{Q\})$ implies $R \in \mathcal{E}_{RS}^w(\mathcal{P})$.
- The operator $\mathcal{E}_{RS}^w$ satisfies (Horn) *supraclassicality*: $\Pi_\vdash \subseteq \mathcal{E}_{RS}^w(\mathcal{P})$, where $\Pi_\vdash = \{Q \mid \Pi \vdash Q\}$.

Finally, the operator $\mathcal{E}_{RS}^w$ satisfies (somewhat softened) *right weakening* with respect to the set of strict rules. Indeed, it is satisfied in the full sense for RP-DeLP programs with a single defeasibility level: let $\mathcal{P} = (\Pi, \Delta, \preceq)$ be an RP-DeLP program with a single defeasibility level for Δ, if $Q \leftarrow P_1 \wedge \ldots \wedge P_k \in \Pi$ and $\{P_1, \ldots, P_k\} \subseteq \mathcal{E}_{RS}^w(\mathcal{P})$, then $Q \in \mathcal{E}_{RS}^w(\mathcal{P})$.

The key point here is how warranted and blocked conclusions at higher levels of the maximal ideal output are taken into account in lower levels. In particular blocked conclusions play a key role in the propagation mechanism between defeasibility levels. In the RP-DeLP approach if a conclusion φ is blocked at level α, then for any lower level than α, not only the conclusion φ is rejected but also every conclusion ψ such that $\{\varphi, \psi\} \vdash \perp$. Then, for the general case we have the following *right weakening* logical property for operator $\mathcal{E}_{RS}^w$. Let $\mathcal{P} = (\Pi, \Delta, \preceq)$ be an RP-DeLP program with defeasibility levels $1 > \alpha_1 > \ldots > \alpha_p > 0$, and let (*Warr, Block*) be the maximal ideal output of $\mathcal{P}$. If $Q \leftarrow P_1 \wedge \ldots \wedge P_k \in \Pi$ and $\{P_1, \ldots, P_k\} \subseteq \mathcal{E}_{RS}^w(\mathcal{P})$, then either $Q \in \mathcal{E}_{RS}^w(\mathcal{P})$ and $N'(Q) \geq \min\{N'(P_i) \mid P_i \in \{P_1, \ldots, P_k\}\}$, or Q, or $\sim Q \in Block(\beta)$ for some $\beta > \min\{N'(P_i) \mid P_i \in \{P_1, \ldots, P_k\}\}$.

6. Conclusions and future work

In this paper we have analyzed the relationship between the exhaustive dialectical analysis based semantics of P-DeLP and the recursive based semantics of RP-DeLP and we have shown that the maximal ideal semantics of RP-DeLP provides a useful framework for making a formal analysis of logical properties of warrant in defeasible argumentation.

Our current research work in RP-DeLP will follow two main directions: on the one hand, we are concerned with characterizing a lower bound on complexity for computing the warranty status of arguments according to the maximal ideal recursive semantics. On the other hand, we are concerned with developing a graphic representation framework of the maximal ideal recursive semantics. This representation could be used as a mechanism for refinement of strict and defeasible information.

References

[1] Teresa Alsinet, Ramón Béjar, Lluis Godo, and Francesc Guitart. RP-DeLP: a weighted defeasible argumentation framework base on a recursive semantics. *Journal of Logic and Computation*, 2014. doi:10.1093/logcom/exu008.

[2] Teresa Alsinet, Carlos I. Chesñevar, Lluis Godo, and Guillermo R. Simari. A logic programming framework for possibilistic argumentation: Formalization and logical properties. *Fuzzy Sets and Systems*, 159(10):1208–1228, 2008.

[3] Leila Amgoud. Postulates for logic-based argumentation systems. In *Proceedings of the ECAI-2012 Workshop WL4AI*, pages 59–67, 2012.

[4] Philippe Besnard and Anthony Hunter. *Elements of Argumentation*. The MIT Press, 2008.

[5] Martin Caminada and Leila Amgoud. On the evaluation of argumentation formalisms. *Artif. Intell.*, 171(5-6):286–310, 2007.

[6] Carlos I. Chesñevar, Ana G. Maguitman, and Ronald P. Loui. Logical Models of Argument. *ACM Computing Surveys*, 32(4):337–383, December 2000.

[7] Alejandro J. García, Jürgen Dix, and Guillermo R. Simari. Argument-based logic programming. In Iyad Rahwan and Guillermo R. Simari, editors, *Argumentation in Artificial Intelligence*, chapter 8, pages 153–171. Springer, 2009.

[8] Alejandro J. García and Guilermo R. Simari. Defeasible Logic Programming: An Argumentative Approach. *Theory and Practice of Logic Programming*, 4(1):95–138, 2004.

[9] John 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.

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

[11] Guillermo R. Simari and Ronald P. Loui. A Mathematical Treatment of Defeasible Reasoning and its Implementation. *Artificial Intelligence*, 53:125–157, 1992.