To appear in the *Journal of Experimental & Theoretical Artificial Intelligence*
Vol. 00, No. 00, Month 20XX, 1–19

Formalization and logical properties of the Maximal Ideal Recursive Semantics for Weighted Defeasible Logic Programming

Teresa Alsinet[a] and Ramón Béjar[a] and 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

(*Received 00 Month 20XX; final version received 00 Month 20XX*)

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. 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 weighted facts associated with warranted conclusions. Given the recursive based semantics of RP-DeLP, we have also implemented an argumentation framework for RP-DeLP that is able to compute not only the output of warranted and blocked conclusions, but also explain the reasons behind the status of each conclusion. We have developed this framework as a stand-alone application with a simple text-based input/output interface to be able to use it as part of other AI systems.

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 (55). 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 (22).

Defeasible Logic Programming (DeLP) (42) 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 (41) (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) (4) 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 (28; 55), 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 (2), 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 (5), 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 (53), 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 (2) 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.

Our final aim is to be able to develop an argumentation framework that is useful for different application domains. For example, recently argumentation tools have been proposed to analyze and understand the information obtained from different Social Web applications, like Twitter or Debatepedia (25; 45), but also specific web applications to allow users to develop debates about specific issues (like policy proposal by governments), and then finally allow to policy makers to understand the relevant opinions on the ongoing debates. A relevant example of this last kind of tools is Parmenides (14), that follows an structured approach to propose arguments in a debate and uses valued abstract argumentation frameworks (15) for analyzing the results of the debate. However, as a first step, it is important to understand the characteristics of our argumentation framework when compared with other existing ones.

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. Finally, considering our final aim of using RP-DeLP in argumentation applications like the ones related to the Social Web, we briefly explain the characteristics of an implementation of an argumentation framework for RP-DeLP we have recently developed as a command-line application that can be easily used by third-party final applications or services.

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; 4).

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 (28; 55; 57). 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 knowledge). 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. (28; 55)), 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.

[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.

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. Dialectical analysis based 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 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 $\mathcal{P} \mid\!\sim^w \langle \mathcal{A}, Q, \alpha\rangle$ 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 (26) 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 $\Pi \cup \mathcal{C}^w_{\mathrm{DT}}(\mathcal{P}) \not\vdash \bot$.

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

[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.

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 pair-wisely generate a conflict since $\Pi \cup \{a, b\} \not\vdash \bot$, $\Pi \cup \{a, c\} \not\vdash \bot$ and $\Pi \cup \{b, c\} \not\vdash \bot$. However, these conclusions are collectively conflicting w.r.t. the strict part of program Π since $\Pi \cup \{a, b, c\} \vdash \bot$, and thus, the warrant semantics of P-DeLP does not satisfy the indirect consistency postulate.

In order to characterize such situations we proposed in (2) 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. Recursive 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 $\Pi \cup W \cup \{Q_1, \ldots, Q_k\} \vdash \bot$; 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 $\Pi \cup W \cup S \not\vdash \bot$.

This general notion of conflict is used to define an *output of an RP-DeLP program* $\mathcal{P} = (\Pi, \Delta, \preceq)$ as a pair $(\mathit{Warr}, \mathit{Block})$ of subsets of $\mathcal{L}$ of warranted and blocked conclusions 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: $\mathit{Warr} = \mathit{Warr}(1) \cup \{\cup_{i=1,p} \mathit{Warr}(\alpha_i)\}$ and $\mathit{Block} = \cup_{i=1,p} \mathit{Block}(\alpha_i)$, where $\mathit{Warr}(1) = \{Q \mid \Pi \vdash Q\}$, and $\mathit{Warr}(\alpha_i)$ and $\mathit{Block}(\alpha_i)$ 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 both blocked conclusions of strength $> \alpha_i$ and the strict knowledge extended with warranted 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 (2) 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 $(\mathit{Warr}, \mathit{Block})$ 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, $\mathit{Warr}(1) = \{y\}$ 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 $\Pi \cup \{p, r\} \vdash \bot$ and $\Pi \cup \{q, s\} \vdash \bot$, 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: $(\mathit{Warr}_1, \mathit{Block}_1)$ and $(\mathit{Warr}_2, \mathit{Block}_2)$, where $\mathit{Warr}_1 = \{y, p\}, \mathit{Block}_1 = \{q, s\}$ and $\mathit{Warr}_2 = \{y, q\}, \mathit{Block}_2 = \{p, r\}$. 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).

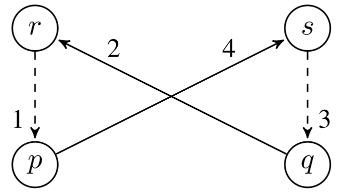

Figure 1. Circular dependences between arguments.

In (2) 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 (53), 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 $\mathcal{P}$: $(\mathit{Warr}_{\max}, \mathit{Block}_{\max})$, where $\mathit{Warr}_{\max} = \{y\}$ and $\mathit{Block}_{\max} = \{p, q\}$.

5. Dialectical analysis and recursive warrant semantics

In (2) 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 $\mathcal{C}^w_{\mathrm{DT}}(\mathcal{P})$ of warranted conclusions based on dialectical trees, $\mathcal{C}^w_{\mathrm{DT}}(\mathcal{P})$ contains each warranted conclusion in the maximal ideal output of $\mathcal{P}$.

Proposition 5.1: *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$(1) = \{Q \mid \mathcal{P} \mid\!\sim^w \langle A, Q, 1\rangle\}$.*

Proof. If $Q \in \mathit{Warr}(\alpha_i)$, there exits an argument $\langle A, Q\rangle$ of strength α_i such that (i) it is based on warranted conclusions; (ii) there does not exist an argument for Q of strength $> \alpha_i$ based on warranted conclusions; (iii) Q is consistent with both blocked conclusions of strength $> \alpha_i$ and the strict knowledge extended with warranted conclusions of strength $> \alpha_i$; (iv) it does not lead to any conflict among arguments of strength α_i and; (v) it is not involved in any circular definition of conflict between arguments of strength α_i. Hence, every acceptable argumentation line w.r.t. $\mathcal{P}$ starting in $\langle A, Q\rangle$ has an odd number of arguments and there is no other argument of the form $\langle B, Q\rangle$ w.r.t. $\mathcal{P}$, with $s(\langle B, Q\rangle) > \alpha_i$, satisfying the above condition, and thus, $\mathcal{P} \mid\!\sim^w \langle A, Q, \alpha_i\rangle$. □

Notice that the inverse of Prop. 5.1 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 dialectical analysis based semantics of P-DeLP and the maximal ideal recursive semantics of RP-DeLP, we have to extend the P-DeLP framework with some mechanism ensuring this property.

In (26) 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 $\neg\phi_1 \vee \neg\phi_2 \vee \ldots \vee \neg\phi_n \vee \psi$. 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 (*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 (26). 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. $\mathcal{C}^w_{\mathrm{DT}}(\mathcal{P}) = \{p, \sim q, \sim s\}$. 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 $\mathcal{C}^w_{\mathrm{DT}}(\mathcal{P}) = \{y, a, b\}$ and $\mathcal{C}^w_{\mathrm{DT}}(\mathcal{P}^\star) = \{y, a, \sim b\}$.

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

Proposition 5.2: *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* $(\mathit{Warr}, \ \mathit{Block})$ *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* $\{Q \mid \mathcal{P}^\star \mid\!\sim^w \langle A, Q, \alpha_i\rangle$ *and* $\Pi \cup A \vdash Q\} \subseteq \mathit{Warr}(\alpha_i)$. *Obviously,* $\{Q \mid \mathcal{P}^\star \mid\!\sim^w \langle A, Q, 1\rangle$ *and* $\Pi \cup A \vdash Q\} = \{Q \mid \Pi \vdash Q\} = \mathit{Warr}(1)$.

Proof. We distinguish between two cases:

(i) $\mathcal{P}^\star \mid\!\sim^w \langle A, Q, \alpha_i\rangle$ and there is no defeater for $\langle A, Q\rangle$ w.r.t. $\mathcal{P}^\star$; i.e. $[\langle A, Q\rangle]$ is the only acceptable argumentation line starting in $\langle A, Q\rangle$, and thus, there does not exist any argument $\langle D, \sim Q\rangle$ w.r.t. $\mathcal{P}^\star$ such that $s(\langle D, \sim Q\rangle) \geq \alpha_i$. Moreover, if $\langle B, P\rangle \sqsubset \langle A, Q\rangle$, there is no defeater of $\langle B, P\rangle$ w.r.t. $\mathcal{P}^\star$, and thus, there does not exist any argument $\langle C, \sim P\rangle$ w.r.t. $\mathcal{P}^\star$ such that $s(\langle C, \sim P\rangle) \geq s(\langle B, P\rangle)$. Since $\mathcal{P}^\star$ contains the closure of strict rules $Cl_{tp}(\Pi)$, if $\Pi \cup A \vdash Q$ we have that $\langle A, Q\rangle$ of strength α_i is valid w.r.t. the recursive warrant semantics for $\mathcal{P}$. Moreover, $\langle A, Q\rangle$ is not involved in a conflict nor in a cycle among valid arguments of strength α_i and the set of warranted conclusions of strength greater than α_i, and thus, argument $\langle A, Q\rangle$ of strength α_i is warranted in the maximal ideal output of $\mathcal{P}$. Hence, if $\Pi \cup A \vdash Q$, $Q \in \mathit{Warr}(\alpha_i)$.

(ii) $\mathcal{P}^\star \mid \sim^w \langle A, Q, \alpha_i\rangle$ and $\langle A, Q\rangle$ has at least one defeater w.r.t. $\mathcal{P}^\star$; i.e. every acceptable argumentation line starting in an argument $\langle A, Q\rangle$ is of the form $\lambda = [\langle A, Q\rangle, \ldots, \mathcal{A}_{2n-1}, \mathcal{A}_{2n}]$ with $n \geq 1$. Again we distinguish between two cases:
(a) Argument $\mathcal{A}_{2n}$ is a blocking defeater for argument $\mathcal{A}_{2n-1}$. In this case, conclusions of arguments $\mathcal{A}_{2n}$ and $\mathcal{A}_{2n-1}$ are not warranted w.r.t. $\mathcal{P}^\star$ and the argumentation line $[\langle A, Q\rangle, \ldots, \mathcal{A}_{2n-1}, \mathcal{A}_{2n}]$ can be omitted in the sense that it is subsumed by the rest of acceptable argumentation lines starting in argument $\langle A, Q\rangle$ and containing argument $\mathcal{A}_{2n-1}$.
(b) Argument $\mathcal{A}_{2n}$ is a proper defeater for argument $\mathcal{A}_{2n-1}$. In this case, $\mathcal{A}_{2n}$ has no defeaters w.r.t. $\mathcal{P}^\star$. Then, if $\mathcal{A}_{2n} = \langle B, P\rangle$ and $\Pi \cup B \vdash P$, $P \in \mathit{Warr}(s(\langle B, P\rangle))$. Moreover, if $\mathcal{A}_{2n-1} = \langle C, R\rangle$, $R \notin \mathit{Warr}(\beta)$ for all $\beta \geq s(\langle C, R\rangle)$. Since the above reasoning can be applied recursively for every argument $A_{2(n-i)}$ in λ with $i = 0 \ldots n$, if $\Pi \cup A \vdash Q$, $Q \in \mathit{Warr}(\alpha_i)$.

□

6. Logical properties of the maximal ideal recursive semantics of RP-DeLP

Following the approach we made in (4) 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 $\mathcal{E}^w_{\mathrm{RS}}$ 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 $(\mathit{Warr}, \mathit{Block})$ be the maximal ideal output of $\mathcal{P}$. We define the operator $\mathcal{E}^w_{\mathrm{RS}}$ associated with $\mathcal{P}$ as follows: $\mathcal{E}^w_{\mathrm{RS}}(\mathcal{P}) = (\Pi \cup \mathit{Warr}(1), \Delta \cup (\cup_{i=1\ldots p}\mathit{Warr}(\alpha_i)), \preceq')$ and such that $N'(\varphi) = N(\varphi)$ for all $\varphi \in \Pi \cup \Delta$, $N'(\varphi) = 1$ for all $\varphi \in \mathit{Warr}(1)$, and $N(\varphi) = \alpha_i$ for all $\varphi \in \mathit{Warr}(\alpha_i)$, $i = 1 \ldots p$.

Notice that by definition operator $\mathcal{E}^w_{\mathrm{RS}}$ is *well-defined* (i.e., given an RP-DeLP program as input, the associated output is also an RP-DeLP program). Moreover, $\mathcal{E}^w_{\mathrm{RS}}$ satisfies *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 $(\mathit{Warr}, \mathit{Block})$, $\Pi \subseteq \Pi \cup \mathit{Warr}(1)$, $\Delta \subseteq \Delta \cup (\cup_{i=1\ldots p}\mathit{Warr}(\alpha_i))$ 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. Moreover, given a different RP-DeLP program $\mathcal{P}'$, if $\varphi \in \mathcal{P}$ in defeasibility level α, we will write $\mathcal{P}' \cup \{\varphi\}$ to denote that program $\mathcal{P}'$ is extended with the clause φ in defeasibility level α.

Besides, *monotonicity* does not hold for $\mathcal{E}^w_{\mathrm{RS}}$, 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, $\mathit{Warr}(1) = \{q\}$ and $\mathit{Warr}(\alpha) = \{p\}$, and thus,

$\{q, p\} \subseteq \mathcal{E}^{w}_{\text{RS}}(\mathcal{P})$. 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, $\mathit{Warr}(1) = \{q, \sim p\}$ and $\mathit{Warr}(\alpha) = \emptyset$ in the maximal ideal output of $\mathcal{P}'$. Hence, $p \not\in \mathcal{E}^{w}_{\text{RS}}(\mathcal{P}')$ but $p \in \mathcal{E}^{w}_{\text{RS}}(\mathcal{P})$.

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 $\mathcal{E}^{w}_{\text{RS}}$, 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, $\mathit{Warr}(1) = \{q\}$, $\mathit{Warr}(\alpha) = \emptyset$ and $\mathit{Block}(\alpha) = \{p, \sim p\}$ for the maximal ideal output of $\mathcal{P}'$. Hence, $p \not\in \mathcal{E}^{w}_{\text{RS}}(\mathcal{P}')$ but $p \in \mathcal{E}^{w}_{\text{RS}}(\mathcal{P})$.

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

Proposition 6.1 (Idempotence, cummulativity and supraclassicality)**:** *Let $\mathcal{P} = (\Pi, \Delta, \preceq)$ be an RP-DeLP program.*

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

Proof. If $1 > \alpha_1 > \ldots > \alpha_p > 0$ are the defeasibility levels of $\mathcal{P}$ and $(\mathit{Warr}, \mathit{Block})$ is the maximal ideal output of $\mathcal{P}$, by definition $\mathcal{E}^{w}_{\text{RS}}(\mathcal{P}) = (\Pi \cup \mathit{Warr}(1), \Delta \cup (\cup_{i=1\ldots p}\mathit{Warr}(\alpha_i)), \preceq')$, where $N'(\varphi) = N(\varphi)$ for all $\varphi \in \Pi \cup \Delta$, $N'(\varphi) = 1$ for all $\varphi \in \mathit{Warr}(1)$, and $N(\varphi) = \alpha_i$ for all $\varphi \in \mathit{Warr}(\alpha_i)$, $i = 1 \ldots p$. Hence, by definition $\mathcal{E}^{w}_{\text{RS}}(\mathcal{P}) \subseteq \mathcal{E}^{w}_{\text{RS}}(\mathcal{E}^{w}_{\text{RS}}(\mathcal{P}))$. Then, if $(\mathit{Warr}', \mathit{Block}')$ is the maximal ideal output of $\mathcal{P}' = \mathcal{E}^{w}_{\text{RS}}(\mathcal{P})$, we have to show that $\mathit{Warr}' \subseteq \mathit{Warr}$. Suppose that $\mathit{Warr}' \not\subseteq \mathit{Warr}$, then there should exist an argument involved in some conflict or some cycle w.r.t. $\mathcal{P}$ and not involved in any conflict and any cycle w.r.t. $\mathcal{P}'$, and thus, there should exist at least an argument $\mathcal{A}$ valid w.r.t. $\mathcal{P}$ but not valid w.r.t. $\mathcal{P}'$ and such that all subarguments of $\mathcal{A}$ are warranted w.r.t. both $\mathcal{P}$ and $\mathcal{P}'$. Therefore, it must be the case that $\mathcal{A}$ is inconsistent with either blocked conclusions of strength greater than $\mathcal{A}$ or the strict knowledge extended with warranted conclusions of strength greater than $\mathcal{A}$. However, as the sets of blocked and warranted conclusions of strength greater than $\mathcal{A}$ are the same in both cases, we have that every valid argument w.r.t. $\mathcal{P}$ is also valid w.r.t. $\mathcal{P}'$, and thus, $\mathit{Warr}' \subseteq \mathit{Warr}$.

Cummulativity property follows directly from idempotence. Finally, as $\{Q \mid \Pi \vdash Q\} = \mathit{Warr}(1) \subseteq \mathcal{E}^{w}_{\text{RS}}(\mathcal{P})$, (Horn) supraclassicality holds for RP-DeLP programs. □

Finally, the operator $\mathcal{E}^{w}_{\text{RS}}$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 $\{P_1, \ldots, P_k\} \subseteq \mathcal{E}^{w}_{\text{RS}}(\mathcal{P})$, then $Q \in \mathcal{E}^{w}_{\text{RS}}(\mathcal{P})$.

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 $\{\varphi, \psi\} \vdash \bot$.

The following example shows the propagation mechanism between defeasibility levels for the maximal ideal recursive semantics. Consider the RP-DeLP program $\mathcal{P} = (\Pi, \Delta, \preceq)$ with

$$\Pi = \{\sim s \leftarrow q, \sim r \leftarrow h\} \text{ and } \Delta = \{q \leftarrow r, h \leftarrow s, r, s, a, b, q \leftarrow a, h \leftarrow b\},$$

and two defeasibility levels for Δ: α_1 and α_2 with $1 > \alpha_1 > \alpha_2 > 0$. Consider that Δ is stratified as follows:

$$\text{level } \alpha_1\text{: } \{q \leftarrow r, h \leftarrow s, r, s, a, b\} \quad \text{level } \alpha_2\text{: } \{q \leftarrow a, h \leftarrow b\}.$$

Obviously, *Warr*$(1) = \emptyset$. Then, at level α_1, we have four valid arguments:

$$\begin{array}{ll} \mathcal{H}_1 = \langle \{r\}, r \rangle, & \mathcal{H}_2 = \langle \{s\}, s \rangle, \\ \mathcal{H}_3 = \langle \{a\}, a \rangle, & \mathcal{H}_4 = \langle \{b\}, b \rangle. \end{array}$$

and four almost valid arguments: [5]

$$\begin{array}{ll} \mathcal{F}_1 = \langle \{r, q \leftarrow r\}, q \rangle, & \mathcal{F}_2 = \langle \{s, h \leftarrow s\}, h \rangle, \\ \mathcal{F}_3 = \langle \{r, q \leftarrow r\}, \sim s \rangle, & \mathcal{F}_4 = \langle \{s, h \leftarrow s\}, \sim r \rangle. \end{array}$$

Figure 2 shows a graphical representation of support and conflict dependences between $\{\mathcal{H}_1, \mathcal{H}_2, \mathcal{H}_3, \mathcal{H}_4\}$ and $\{\mathcal{F}_1, \mathcal{F}_2, \mathcal{F}_3, \mathcal{F}_4\}$. Obviously, a and b can be warranted since they are not involved in any conflict nor in any cycle. The cycles express that either r or s can be warranted, but not both. Since valid arguments and almost valid arguments involved in cycles must be blocked and rejected, respectively, the maximal ideal output at level α_1 is:

$$\textit{Warr}(\alpha_1) = \{a, b\}, \quad \textit{Block}(\alpha_1) = \{r, s\}.$$

Then, at level α_2 we have that arguments

$$\langle \{q \leftarrow a, a\}, q \rangle \text{ and } \langle \{h \leftarrow b, b\}, h \rangle$$

are valid and they are not involved in any cycle nor in any conflict, and thus, q and h are warranted conclusions at level α_2 (i.e. $\{q, h\} \subseteq \textit{Warr}(\alpha_2)$). Finally, although arguments

$$\langle \{q \leftarrow a, a\}, \sim s \rangle \text{ and } \langle \{h \leftarrow b, b\}, \sim r \rangle$$

are recursively based on warranted conclusions, both are inconsistent with blocked conclusions at level α_1, and thus, s and r are rejected for the maximal ideal output:

$$\textit{Warr}(\alpha_2) = \{q, h\}, \quad \textit{Block}(\alpha_2) = \emptyset.$$

Hence, the maximal ideal output for $\mathcal{P}$ is:

$$\textit{Warr} = \{a, b, q, h\}, \quad \textit{Block} = \{r, s\}.$$

Therefore, due to the fact that the set of conclusions that are warranted and blocked at each level determines which arguments are valid at lower levels, we get that $\Pi \cup \textit{Warr} \vdash \sim s$ and $\Pi \cup \textit{Warr} \vdash \sim r$, but $\sim s, \sim r \notin \textit{Warr}$ since $s, r \in \textit{Block}(\alpha_1)$.

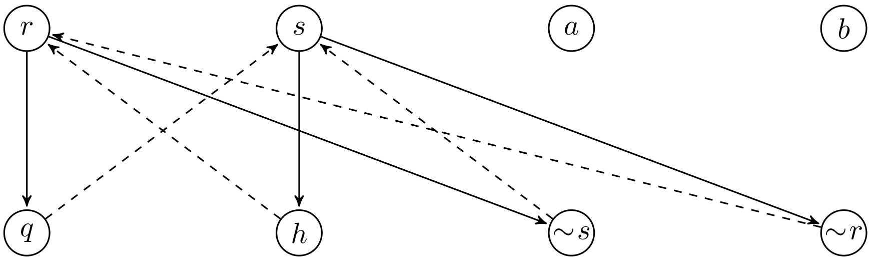

Figure 2. Support and conflict dependences between valid and almost valid arguments.

[5] Intuitively, an almost valid argument is an argument based on valid arguments and which status is warranted (not rejected) whenever these subarguments are warranted, and rejected, otherwise.

Then, for the general case we have the following *right weakening* logical property for operator $\mathcal{E}^{w}_{\text{RS}}$.

Proposition 6.2 (Right weakening)**:** *Let $\mathcal{P} = (\Pi, \Delta, \preceq)$ be an RP-DeLP program with defeasibility levels $1 > \alpha_1 > \ldots > \alpha_p > 0$, and let $(\mathit{Warr}, \mathit{Block})$ be the maximal ideal output of $\mathcal{P}$. If $Q \leftarrow P_1 \wedge \ldots \wedge P_k \in \Pi$ and $\{P_1, \ldots, P_k\} \subseteq \mathcal{E}^{w}_{\text{RS}}(\mathcal{P})$, then either $Q \in \mathcal{E}^{w}_{\text{RS}}(\mathcal{P})$ and $N'(Q) \geq \min\{N'(P_i) \mid P_i \in \{P_1, \ldots, P_k\}\}$, or Q, or $\sim Q \in \mathit{Block}(\beta)$ for some $\beta > \min\{N'(P_i) \mid P_i \in \{P_1, \ldots, P_k\}\}$.*

Proof. We prove that if $\Pi \cup \mathit{Warr}(\geq \alpha_i) \vdash Q$ and $\Pi \cup \mathit{Warr}(> \alpha_i) \not\vdash Q$, then either $Q \in \mathit{Warr}(\alpha_i)$, or $Q \in \mathit{Block}(> \alpha_i)$, or $\sim Q \in \mathit{Block}(> \alpha_i)$. [6]

Suppose that for some α_i, $\Pi \cup \mathit{Warr}(\geq \alpha_i) \vdash Q$, $\Pi \cup \mathit{Warr}(> \alpha_i) \not\vdash Q$, $Q \notin \mathit{Warr}(\alpha_i)$, and $Q, \sim Q \notin \mathit{Block}(> \alpha_i)$. Then, since $\Pi \cup \mathit{Warr} \not\vdash \bot$, $\Pi \cup \mathit{Warr}(\geq \alpha_i) \cup \{Q\} \not\vdash \bot$, there exists a valid argument $\langle A, Q\rangle$ for Q of strength α_i. Now, since $Q \notin \mathit{Warr}(\alpha_i)$ we get two possible cases:

Case 1 There exists a conflict between $\langle A, Q\rangle$ and a set of valid arguments $\mathbb{G}$, with $\langle A, Q\rangle \not\sqsubset \mathbb{G}$, w.r.t. $W = \mathit{Warr}(> \alpha_i) \cup \{P \mid \langle B, P\rangle \sqsubset \mathbb{G} \cup \{\langle A, Q\rangle\}\}$. Thus $\Pi \cup W \cup \{Q\} \cup \{P \mid \langle B, P\rangle \in \mathbb{G}\} \vdash \bot$ and $\Pi \cup W \cup S \not\vdash \bot$, for all $S \subset \{Q\} \cup \{P \mid \langle B, P\rangle \in \mathbb{G}\}$. Consider now $W' = \{R \mid \langle B, R\rangle \sqsubset \langle A, Q\rangle\}$. Then, as $W' \subseteq W$ and $\Pi \cup W' \vdash Q$, if $\Pi \cup W \cup \{Q\} \cup \{P \mid \langle B, P\rangle \in G\} \vdash \bot$, then $\Pi \cup W \cup \{P \mid \langle B, P\rangle \in \mathbb{G}\} \vdash \bot$, and thus, either Q is warranted at level α_i or Q is rejected at level α_i because Q or $\sim Q$ are blocked at a level β with $\beta > \alpha_i$. In other words, either $Q \in \mathit{Warr}(\alpha_i)$, or $Q \in \mathit{Block}(> \alpha_i)$, or $\sim Q \in \mathit{Block}(> \alpha_i)$.

Case 2 There is a cycle in a warrant dependency graph at level α_i and either $\langle A, Q\rangle$ is a vertex of the cycle or there exists a path from some vertex of the cycle to $\langle A, Q\rangle$. If $\mathbb{H}$ is the set of valid arguments of the cycle and $\mathbb{F}$ the set of almost valid arguments w.r.t. $\mathbb{H} \cup \{\langle A, Q\rangle\}$, there is an almost valid argument for conclusion $\sim Q$ in $\mathbb{F}$ or an strict rule $\sim Q \leftarrow L_1 \wedge \ldots \wedge L_m \in \Pi$ such that $\{L_1, \ldots, L_m\} \subseteq \mathit{Warr}(\geq \alpha_i) \cup \{H \mid \langle E, H\rangle \in \mathbb{H}\} \cup \{F \mid \langle J, F\rangle \in \mathbb{F}\}$. Hence, there is an almost valid argument $\langle D, \sim Q\rangle$ for conclusion $\sim Q$ in $\mathbb{F}$, and an edge from the vertex of $\sim Q$ to the vertex of Q. Now, since $\Pi \cup \mathit{Warr}(\geq \alpha_i) \vdash Q$ and $Q \notin \mathit{Warr}(\geq \alpha_i)$, there exists a strict rule $Q \leftarrow L'_1 \wedge \ldots \wedge L'_p \in \Pi$ with all the L'_j's in $\mathit{Warr}(\geq \alpha_i)$. Moreover, as $\Pi \cup \mathit{Warr}(> \alpha_i) \not\vdash Q$, there is at least one literal $L' \in \{L'_1, \ldots, L'_p\}$ such that $L' \in \mathit{Warr}(\alpha_i)$, and thus, there is a valid argument $\langle J, L'\rangle$ for L' of strength α_i and $\langle A, Q\rangle \not\sqsubset \langle J, L'\rangle$. Then, there is a cycle in the warrant dependency graph for $\mathbb{H} \cup \{\langle A, Q\rangle\} \cup \{\langle J, L'\rangle\}$ and $\mathbb{F}$ and an edge from the vertex of $\sim Q$ to the vertex of L', and thus, $L' \notin \mathit{Warr}(\alpha_i)$. Hence, either $Q \in \mathit{Warr}(\alpha_i)$, or $Q \in \mathit{Block}(> \alpha_i)$, or $\sim Q \in \mathit{Block}(> \alpha_i)$. □

Let us briefly discuss the most relevant results provided by the properties presented before. When analyzing warranted conclusions under the operator $\mathcal{E}^{w}_{\text{RS}}$, idempotence shows us that adding warranted conclusions as new facts to a given program does not add any new warrant capabilities. Cummulativity shows us that any warranted conclusion obtained from a program $\mathcal{P}$ can be considered as a strict warranted conclusion since in some sense it is an intermediate proof (lemma) to be used in building more complex warranties. (Horn) supraclassicality indicates that every conclusion that follows via traditional SLD inference (involving only certain clauses) can be considered as a special form of argument (namely, an empty argument), whereas right weakening tells us that RP-DeLP preserves the usual semantics for Horn rules: the existence of an strict rule $X \leftarrow Y$ causes that the warranty of Y is also a warranty for X.

[6] We will write $\mathit{Warr}(\geq \alpha_i)$ and $\mathit{Warr}(> \alpha_i)$ to denote $\cup_{\beta \geq \alpha_i} \mathit{Warr}(\beta)$ and $\cup_{\beta > \alpha_i} \mathit{Warr}(\beta)$, respectively, and analogously for $\mathit{Block}(> \alpha_i)$, assuming $\mathit{Block}(> \alpha_1) = \emptyset$.

7. Implementation of the RP-DeLP framework

As we have said at the introduction, our final aim is to investigate the use of an argumentation framework based on RP-DeLP in application domains such that debates in social networks. As a first step, we have implemented the algorithm for computation of the maximal ideal output (2) and also the algorithm for computation of the multiple outputs semantics (1). Both algorithms are based on solving a sequence of queries, with complexity within NP, related to the discovery of valid arguments and conflicts between sets of arguments. We have made available both algorithms in a single system, and the implementation of the main code of both algorithms has been done with the programming language `python`. However, in order to have a good scaling behavior when solving RP-DeLP programs of big size, the NP queries are solved trough reductions to SAT encodings or ASP encodings. Our system is freely available as a command-line application at the GitHub repository [7] and can be installed on Linux and Mac OSX systems, but we have also available a web based version in our web server http://arinf.udl.cat/rp-delp. In order to install and use the command-line version in your our system, apart of having python 2.7 installed, one needs to have also installed the SAT solver minisat (34) [8] and the ASP solver clingo (43) [9]. The web version can be used trough the html interface provided, but final applications can also use it trough the http POST method.

One important feature of our system is that due to the recursive nature of our semantics, the computation of the maximal ideal output allows not only to provide warranted and blocked conclusions, but also easily explain the reasons for a warranted or blocked conclusion. Remember that for the recursive semantics from Section 4 the main notion for characterizing warranted and blocked conclusions is the one of valid argument. The recursive definition for valid argument indicates that an argument is valid if it is based on warranted subarguments and it does not have conflicts with warranted conclusions and blocked conclusions of greater strength.

In practical terms, a valid argument for a conclusion Q is simply explained in our system by a rule $Q \leftarrow W_1 \wedge \ldots \wedge W_n$, where each W_i is a warranted conclusion, and the strength α with which Q has been found valid. To fully understand the argument for Q, the user has simply to recursively follow the arguments provided for each W_i. Then, if our system finds a conflict between Q and other valid conclusions of the same strength, it blocks Q and gives that set of conflicting valid conclusions as the explanation for blocking Q. More complex is the situation of blocked conclusion due to a circular definition of conflict between valid arguments. In (2) this situation was characterized through the definition of dependency graphs, like the ones shown in figures 1 and 2. However, for obtaining a more efficient implementation we have followed an approach that avoids the explicit computation of dependency graphs. Instead, we follow an alternative characterization of circular conflicts based on the detection of sets of valid arguments that after two iterations of the main loop of our algorithm cannot be decided to be either warranted or blocked because of dependencies that cannot be resolved. In that case, our system identifies such valid conclusions also as blocked, but due to a circular conflict, and inform of such set of valid conclusions as being part of a circular conflict. The problem of choosing this, more indirect, way of detecting circular conflicts is that the information provided to the user is not as clear as in the case of the information for valid and warranted conclusions. But we believe that it is a good trade-off between the computation cost and the quality of the information provided to the user to understand the output of the argumentation system.

Consider the following example of application of RP-DeLP to obtain the arguments that are consistently supported by all the participants in a political debate. The debate is about possible policies to increase the gross domestic product (represented by literal G). There are different policies being discussed: increase the public infrastructures expenditure (A_1), reduce the public

[7] At the URL https://github.com/f-guitart/RP-DeLP_solver/

[8] We have tested the correct use of minisat version 2.2, available at http://minisat.se/MiniSat.html

[9] Our system needs clingo version 3.0.5. Available at http://sourceforge.net/projects/potassco/files/clingo/

debt (A_2), increase the taxes on the richer (A_3) and increase the education expenditure (A_4). All the participants agree that, no matter what policies are chosen, the public debt should be in any case decreased and that if we have increase in infrastructures and education expenditure then the public debt will not decrease. We represent these two arguments as the strict knowledge, because all the participants agree that this should hold in any possible scenario:

$$\Pi = \{\sim A_2 \leftarrow A_1 \wedge A_4,\ A_2\}$$

For the rest of arguments, not all the participants agree with the same strength, so we turn out to have two defeasible levels. At the stronger one (more participants support this opinion), we have the belief that increase the education expenditure should be performed, so we have:

$$\Delta_1 = \{A_4\}$$

At the next defeasible level, we have other opinions that have less support from the participants. Participants believe that increase the public infrastructures expenditure and increase the taxes on the richer should be performed. Also, they believe that the GDP would be increased if there is reduction in public debt and taxes on the richer are not increased, but another equally strong opinion is that the GDP would be increased if there is increase of taxes on the richer and increase on the public infrastructures expenditure. Finally, another opinion is that if the reduction of the public debt is going to happen, then necessarily this cannot be done without not increasing the taxes on the richer. So, at the second defeasible level we have this knowledge:

$$\Delta_2 = \{A_1,\ A_3,\ G \leftarrow A_2 \wedge \sim A_3,\ G \leftarrow A_3 \wedge A_1,\ \sim A_3 \leftarrow A_2\}$$

Our argumentation system will give the following output for this program:

- Warranted conclusions: `[(A2, [A2], strict), (A4, [A4], 1)]`. Meaning that A_2 is warranted at the strict level with the fact A_2 and that A_4 is warranted at the first defeasible level with the fact A_4.
- Blocked conclusions: `[(A3, conflict, [A3, ∼ A3], [A3], 2), (∼ A3, conflict, [A3, ∼ A3], [A2], 2)]`. Meaning that the valid argument for A_3 given by the fact A_3, has a conflict with a valid argument for $\sim A_3$ and that the valid argument for $\sim A_3$ given by the rule $\sim A_3 \leftarrow A_2$ and the warranted fact A_2, has a conflict with a valid argument for A_3, at the second defeasible level.

This information is discovered in an iterative process, starting from simpler arguments and moving towards more complex ones. First, A_2 and A_4 are found to be warranted (no conflicts with Π up to level Δ_1. Then, at level Δ_2 A_1 is found to be not valid, as it produces a conflict with Π and the current set of warrants. Finally, valid arguments for A_3 and $\sim A_3$ are found, but with the same strength, so they are blocked because they generate a conflict.

The command-line version of our argumentation system gives this output formatted in a JSON object [10], so that web based and similar applications can easily process the output. Regarding the input format for programs, it is a simple ASCII file format that follows a syntax similar to the one used by standard Prolog interpreters, but with labels that distinguish the strict and the different defeasible parts of the program.

8. Related work

RP-DeLP, as well as its predecessor P-DeLP (3; 4), builds on top of Defeasible Logic Programming argumentative system (DeLP) (42), and extends it by introducing different levels of preference or priority at the object language level by means of weights. In this approach, arguments are sets of weighted formulas that support a goal, and weights are used to compute the strength of an argument and then, to resolve conflicts among contradictory conclusions.

Actually, introducing preferences in argumentation frameworks goes back to Simari and

[10] A popular data-interchange format used nowadays by many web applications.

Liou (57). In that work the authors defined an argumentation framework in which arguments are built from a propositional knowledge base. The arguments grounded on specific information are considered stronger than the ones built from more general information. This preference relation is used then to solve conflicts between a pair of conflicting arguments. Nonetheless, the way RP-DeLP (and P-DeLP) makes use of preferences, to define the strength of arguments by stratifying the formulas in a program (or knowledge base), directly stems from Brewka's preferred subtheories based approach (23) to nonmonotonic reasoning, where the different levels in which default theories are stratified represent different degrees of reliability. This idea has been used in many other approaches to reasoning with inconsistent information, mainly in those related to possibilistic logic (17; 18; 19; 20). In particular, the notion of argumentative inference introduced in (18) is based on a measure of strength of arguments that is in fact the one used in RP-DeLP. On the other hand, Prakken and Sartor (54) formalized the role of preferences in the underlying logical formalisms that instantiate Dung's seminal theory of argumentation (29).

Other approaches have formalized the role of preferences at an abstract level. In Amgoud and Cayrol's preference-based argumentation frameworks (PAFs) (8; 9), Dung's framework is augmented with a preference ordering on the set of arguments, so that an attack by an argument X on an argument Y is successful only if Y is not preferred to X. In Bench-Capon's value-based Argumentation Frameworks (16), Dung's framework is augmented with values and value orderings, so that an attack by X on Y is successful only if the value promoted by Y is not ranked higher than the value promoted by X according to a given ordering on values. Recently, Kaci (48) and Amgoud and Vesic (10; 11; 12; 13) have addressed the issue of how consistency postulates (26) can be ensured for instantiations of PAFs. They all argue that instantiations of standard PAFs have problems with unsuccessful asymmetric binary attacks. Kaci (48) argues that all attacks should therefore be symmetric. However, Amgoud and Besnard (6; 7) show that for logic-based argumentation systems this would still lead to inconsistency problems, and they show that in order to satisfy the consistency postulates an attack relation should be valid, in the sense that when two arguments have jointly inconsistent premises, they should attack each other.

The output for an RP-DeLP program is a rank-ordered set of warranted and blocked conclusions which satisfy the consistency postulates (26). In contrast to DeLP and other argument-based approaches, the RP-DeLP semantics is based on a (not necessarily binary) general notion of collective conflict among arguments and on the fact that if an argument is warranted it must be that all its subarguments also are warranted.

Collective conflicts has also been considered in several papers, e.g. in (49), while in (6; 7) discuss to some extent the problems binary attacks can cause. On the other hand, the idea of defining a warrant semantics on the basis of conflicting sets of arguments was proposed in (59) and (49). The difference between these approaches and our notion of collective conflict is that in (59) the notion of conflict is not relative to a set of already warranted conclusions and (49) defines a generalization of Dung's abstract framework with sets of attacking arguments not relative to the strict part of the knowledge base. Although the RP-DeLP semantics for warranted conclusions is skeptical, circular definitions of conflict between sets of arguments can lead to situations in which multiple evaluation orders exist, giving rise to different outputs of warranted and blocked conclusions. Following Pollock's recursive semantics for defeasible argumentation (53), circular definitions of conflict between sets of arguments have been characterized by means of dependency graphs representing support and collective conflict relations between the conclusions of arguments and the strict part of the knowledge base.

RP-DeLP recursive semantics draws from the so-called "ideal semantics" promoted by Dung, Mancarella and Toni (30; 31) as an alternative basis for skeptical reasoning within abstract argumentation settings. Informally, ideal acceptance not only requires an argument to be skeptically accepted in the traditional sense but further insists that the argument is in an admissible set all of whose arguments are also skeptically accepted. While the original proposal was couched in terms of the so-called preferred semantics for abstract argumentation, in (33) the notion of "ideal acceptability" has been extended to arbitrary semantics, showing that standard properties of classical ideal semantics, e.g. unique status, continue to hold in some extension-based semantics (see also (32) for an analysis of the computational complexity of the ideal semantics within abstract argumentation frameworks and assumption-based argumentation frameworks). In RP-DeLP, the maximal ideal output for an RP-DeLP program is defined in terms of the maximum rank-ordered set of warranted and blocked conclusions recursively based on warranted information and not involved in neither a conflict nor a circular definition of conflict. The idea is that if a conclusion is warranted at a given level β, so it could also be at any higher level. A different approach could have been to consider that blocked conclusions at one level are not propagated to lower levels. In such a case, an alternative semantics for our system could therefore be defined following a similar line to the one in (44).

Research in logical properties for defeasible argumentation can be traced back to Benferhat et al. (18; 21) and Vreeswijk (59; 60). In the context of his abstract argumentation systems, Vreeswijk showed that many logical properties for non-monotonic inference relationships turned out to be counter-intuitive for argument-based systems. Benferhat et al. (18) were the first who studied argumentative inference in uncertain and inconsistent knowledge bases. They defined an argumentative consequence relationship taking into account the existence of arguments favoring a given conclusion against the absence of arguments in favor of its contrary. In (4) we defined a non-monotonic expansion operator for P-DeLP which modeled the expansion of a program by adding new weighed facts associated with warranted literals according with the dialectic analysis based semantics of P-DeLP and we showed that right-weakening do not hold for P-DeLP.

Challenges in the field of argumentation has been recently addressed towards the implementation of argumentation frameworks. The system ASPARTIX (35; 36) is a tool for computing acceptable extensions for some formalizations of Dung's abstract argumentation framework and relies on a fixed disjunctive datalog program which takes an instance of an argumentation framework as input, and uses an ASP solver for computing the extension specified by the user. The work of Nieves et al. (50) also suggest to use ASP for computing extensions of argumentation frameworks. Dungine (58) is a Java reasoner capable of reasoning with grounded and preferred extensions. CASAPI (39) is a Prolog implementation that combines abstract and assumption-based argumentation. ArguLab (52) is a library of components that provide basic functionalities for agent-related argumentation tasks and DIAMOND (37) uses ASP encodings to compute interpretations of Brewka and Woltran's abstract dialectical frameworks (24).

In our case, we have implemented the algorithm for computation of the maximal ideal output (2) and also the algorithm for computation of the multiple outputs semantics (1). We have made available both algorithms in a single system and the implementation is based on reductions to SAT encodings and ASP encodings.

9. 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. In this sense we have defined a non-monotonic expansion operator for RP-DeLP which modeled the expansion of a program by adding new weighed facts associated with warranted literals and we showed that idempotence, cummulativity and right-weakening properties hold for RP-DeLP. We have also developed an argumentation framework for RP-DeLP that is able to compute not only the output of warranted and blocked conclusions, but also explain the reasons behind the status of each conclusion. Our system is freely available as a command-line application and as a web based service in order to use it as part of other AI systems. Argumentation web services can also be used as part of bigger systems, like in the BDI system described in (56) where they use an available argumentation web system (40) for the DeLP argumentation framework (42).

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. For this goal, we plan to study reductions from ideal semantics for abstract argumentation frameworks, given that its complexity is well understood (32), or also reductions from the more recent framework of probabilistic abstract frameworks (38). Studying reductions from these systems to ours is not only useful for obtaining lower bound on complexity, but also to understand the relationship between the semantics of these argumentation systems with the maximal ideal semantics of RP-DeLP. On the other hand, we are concerned with developing a graphical representation framework of the maximal ideal recursive semantics. This representation could be used as a mechanism for refinement of strict and defeasible information. As more concrete application domains, we have already started to consider the use of our system to encourage users to domains such that debates in social networks (47) and discuss political actions (61), through the use of argumentation structures, following the line of an existing tool for that purpose that is based also on argumentation structures: the Parmenides system for deliberative democracy (14; 27). For using our system in other application domains like in legal reasoning, another interesting line of work would be the study of relationships between RP-DeLP and deontic logics, considering recent approaches to characterize deontic logics beyond classic deontic logic, like for example the alternative semantics (46) or the input/output logic (51).

Acknowledgments

This research was partially supported by the Spanish projects EdeTRI (TIN2012-39348-C02-01) and AT (CONSOLIDER- INGENIO 2010, CSD2007-00022).

References

[1] Teresa Alsinet, Ramón Béjar, Lluis Godo, and Francesc Guitart. On the implementation of a multiple output algorithm for defeasible argumentation. In *SUM* , pages 71–77, 2013.

[2] 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.

[3] Teresa Alsinet, Carlos I. Chesñevar, Lluis Godo, Sandra Sandri, and Guillermo R. Simari. Formalizing argumentative reasoning in a possibilistic logic programming setting with fuzzy unification. *Int. J. Approx. Reasoning*, 48(3):711–729, 2008.

[4] 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.

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

[6] Leila Amgoud and Philippe Besnard. Bridging the gap between abstract argumentation systems and logic. In *SUM*, pages 12–27, 2009.

[7] Leila Amgoud and Philippe Besnard. A formal analysis of logic-based argumentation systems. In *SUM*, pages 42–55, 2010.

[8] Leila Amgoud and Claudette Cayrol. Inferring from inconsistency in preference-based argumentation frameworks. *J. Autom. Reasoning*, 29(2):125–169, 2002.

[9] Leila Amgoud and Claudette Cayrol. A reasoning model based on the production of acceptable arguments. *Annals of Mathematics and Artificial Intelligence*, 34(1–3):197–215, 2002.

[10] Leila Amgoud and Srdjan Vesic. Repairing preference-based argumentation frameworks. In *IJCAI*, pages 665–670, 2009.

[11] Leila Amgoud and Srdjan Vesic. Generalizing stable semantics by preferences. In *COMMA*, pages 39–50, 2010.

[12] Leila Amgoud and Srdjan Vesic. Handling inconsistency with preference-based argumentation. In *SUM*, pages 56–69, 2010.

[13] Leila Amgoud and Srdjan Vesic. A new approach for preference-based argumentation frameworks. *Ann. Math. Artif. Intell.*, 63(2):149–183, 2011.

[14] Katie Atkinson, Trevor J. M. Bench-Capon, and Peter McBurney. PARMENIDES: facilitating deliberation in democracies. *Artif. Intell. Law*, 14(4):261–275, 2006.

[15] Trevor J. M. Bench-Capon. Persuasion in practical argument using value-based argumentation frameworks. *J. Log. Comput.*, 13(3):429–448, 2003.

[16] Trevor J. M. Bench-Capon. Persuasion in practical argument using value-based argumentation frameworks. *J. Log. Comput.*, 13(3):429–448, 2003.

[17] Salem Benferhat, Claudette Cayrol, Didier Dubois, Jérôme Lang, and Henri Prade. Inconsistency management and prioritized syntax-based entailment. In *IJCAI*, pages 640–647, 1993.

[18] Salem Benferhat, Didier Dubois, and Henri Prade. Argumentative inference in uncertain and inconsistent knowledge base. In *UAI*, pages 411–419, 1993.

[19] Salem Benferhat, Didier Dubois, and Henri Prade. How to infer from inconsisent beliefs without revising? In *IJCAI*, pages 1449–1457, 1995.

[20] Salem Benferhat, Didier Dubois, and Henri Prade. Reasoning in inconsistent stratified knowledge bases. In *ISMVL*, pages 184–189, 1996.

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

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

[23] Gerhard Brewka. Preferred subtheories: An extended logical framework for default reasoning. In *IJCAI*, pages 1043–1048, 1989.

[24] Gerhard Brewka and Stefan Woltran. Abstract dialectical frameworks. In *KR*, pages 1043–1048, 2010.

[25] Elena Cabrio and Serena Villata. Natural language arguments: A combined approach. In *ECAI*, pages 205–210, 2012.

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

[27] Dan Cartwright and Katie Atkinson. Using computational argumentation to support e-participation. *IEEE Intelligent Systems*, 24(5):42–52, 2009.

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

[29] Phan Minh Dung. On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games. *Artif. Intell.*, 77(2):321–358, 1995.

[30] Phan Minh Dung, Paolo Mancarella, and Francesca Toni. A dialectic procedure for sceptical, assumption-based argumentation. In *COMMA*, pages 145–156, 2006.

[31] Phan Minh Dung, Paolo Mancarella, and Francesca Toni. Computing ideal sceptical argumentation. *Artif. Intell.*, 171(10-15):642–674, 2007.

[32] Paul E. Dunne. The computational complexity of ideal semantics. *Artif. Intell.*, 173(18):1559–1591, 2009.

[33] Wolfgang Dvorák, Paul E. Dunne, and Stefan Woltran. Parametric properties of ideal semantics. In *IJCAI*, pages 851–856, 2011.

[34] Niklas Eén and Niklas Sörensson. An extensible sat-solver. In *SAT*, volume 2919 of *LNCS*, pages 502–518, 2003.

[35] Uwe Egly, Sarah Alice Gaggl, and Stefan Woltran. ASPARTIX: implementing argumentation frameworks using answer-set programming. In *ICLP*, pages 734–738, 2008.

[36] Uwe Egly, Sarah Alice Gaggl, and Stefan Woltran. Answer-set programming encodings for argumentation frameworks. *Argument & Computation*, 1(2):147–177, 2010.

[37] Stefan Ellmauthaler and Hannes Strass. The DIAMOND system for computing with abstract dialectical frameworks. In *COMMA*, pages 233–240, 2014.

[38] Bettina Fazzinga, Sergio Flesca, and Francesco Parisi. On the complexity of probabilistic abstract argumentation. In *IJCAI*, 2013.

[39] Dorian Gaertner and Francesca Toni. Hybrid argumentation and its properties. In *COMMA*, pages 183–195, 2008.

[40] A. J. García, N. D. Rotstein, M. Tucat, and G. R. Simari. An argumentative reasoning service for deliberative agents. In *2nd KSEM, LNCS*, volume 4798, pages 128–139, 2007.

[41] 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.

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

[43] M. Gebser, R. Kaminski, B. Kaufmann, M. Ostrowski, T. Schaub, and M. Schneider. Potassco: The Potsdam answer set solving collection. *AI Communications*, 24(2):105–124, 2011.

[44] Guido Governatori, Michael J. Maher, Grigoris Antoniou, and David Billington. Argumentation semantics for defeasible logic. *J. Log. Comput.*, 14(5):675–702, 2004.

[45] Kathrin Grosse, Carlos Iván Chesñevar, and Ana Gabriela Maguitman. An argument-based approach to mining opinions from twitter. In *AT*, pages 408–422, 2012.

[46] S.O. Hansson. Alternative semantics for deontic logic. In Dov Gabbay, John Horty, Xavier Parent, Ron van der Meyden, and Leon van der Torre, editors, *Handbook of Deontic Logic and Normative Systems*. College Publications, 2013.

[47] Stella Heras, Katie Atkinson, Vicente J. Botti, Floriana Grasso, Vicente Julián, and Peter McBurney. Research opportunities for argumentation in social networks. *Artif. Intell. Rev.*, 39(1):39–62, 2013.

[48] Souhila Kaci. Refined preference-based argumentation frameworks. In *COMMA*, pages 299–310, 2010.

[49] Søren Holbech Nielsen and Simon Parsons. A generalization of Dung's abstract framework for argumentation: Arguing with sets of attacking arguments. In *ArgMAS*, pages 54–73, 2006.

[50] Juan Carlos Nieves, Ulises Cortés, and Mauricio Osorio. Possibilistic-based argumentation: An answer set programming approach. In *ENC*, pages 249–260, 2008.

[51] X. Parent and L. van der Torre. Input/output logic. In Dov Gabbay, John Horty, Xavier Parent, Ron van der Meyden, and Leon van der Torre, editors, *Handbook of Deontic Logic and Normative Systems*. College Publications, 2013.

[52] Mikolaj Podlaszewski, Martin Caminada, and Gabriella Pigozzi. An implementation of basic argumentation components. In *AAMAS*, pages 1307–1308, 2011.

[53] 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.

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

[55] 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.

[56] Federico Schlesinger, Marcelo Errecalde, and Guillermo Aguirre. An approach to integrate web services and argumentation into a bdi system (extended abstract). In *AAMAS*, pages 1371–1372, 2010.

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

[58] Matthew South, Gerard Vreeswijk, and John Fox. Dungine: A java dung reasoner. In *COMMA*, pages 360–368, 2008.

[59] Gerard Vreeswijk. Abstract argumentation systems. *Artif. Intell.*, 90(1-2):225–279, 1997.

[60] Gerard A.W. Vreeswijk. *Studies in Defeasible Argumentation.* PhD thesis, Vrije University, Amsterdam (Holanda), 1993.

[61] Maya Wardeh, Adam Wyner, Katie Atkinson, and Trevor J. M. Bench-Capon. Argumentation based tools for policy-making. In *ICAIL*, pages 249–250, 2013.