2.13 On the Computation of Warranted Arguments within a Possibilistic Logic Framework with Fuzzy Unification

On the Computation of Warranted Arguments within a Possibilistic Logic Framework with Fuzzy Unification *

Teresa Alsinet Carlos Chesñevar Dept. of Computer Science, University of Lleida, Lleida, SPAIN {tracy,cic}@eps.udl.es

Lluís Godo Sandra Sandri AI Research Institute (IIIA-CSIC), Campus UAB, Bellaterra, SPAIN {godo,sandri}@iiia.csic.es

Guillermo Simari Dept. of Computer Science and Eng., Universidad Nacional del Sur, Bahía Blanca, ARGENTINA grs@cs.uns.edu.ar

Abstract

Possibilistic Defeasible Logic Programming (P-DeLP) is a logic programming language which combines features from argumentation theory and logic programming, incorporating the treatment of possibilistic uncertainty at object-language level. The aim of this paper is twofold: first to present an approach towards extending P-DeLP in order to incorporate fuzzy constants and fuzzy unification, and after to propose a way to handle conflicting arguments in the context of the extended framework.

Keywords: Possibilistic logic, fuzzy constants, fuzzy unification, defeasible argumentation.

Introduction

In the last decade, defeasible argumentation has emerged as a very powerful paradigm to model commonsense reasoning in the presence of incomplete and potentially inconsistent information (Chesñevar, Maguitman, & Loui 2000). Recent developments have been oriented towards integrating argumentation as part of logic programming languages. In this context, Possibilistic Defeasible Logic Programming (P-DeLP) (Chesñevar *et al.* 2004) is a logic programming language which combines features from argumentation theory and logic programming, incorporating the treatment of possibilistic uncertainty at object-language level. Roughly speaking, in P-DeLP degrees of uncertainty help in determining which arguments prevail in case of conflict.

In spite of its expressive power, an important limitation in P-DeLP (as defined in (Chesñevar *et al.* 2004)) is that the treatment of imprecise, fuzzy information was not formalized. One interesting alternative for such formalization is the use of PGL$^+$, a Possibilistic logic over Gödel logic extended with fuzzy constants. Fuzzy constants in PGL$^+$ allow expressing imprecise information about the possibly unknown value of a variable (in the sense of magnitude) modeled as a (unary) predicate. For instance, an imprecise statement like "John's salary is low" can be expressed PGL$^+$ by the formula $John\_salary(low)$ where $John\_salary$ is a predicate and low a fuzzy constant, which will be mapped under a given PGL$^+$ interpretation to a fuzzy set rather to a single domain element as usually in predicate logics. Notice that this kind of statements express *disjunctive* knowledge (mutually exclusive), in the sense that in each interpretation it is natural to require that the predicate $John\_salary(x)$ be true for one and only one variable assignment to x, say u_0. Then, in such an interpretation it is also natural to evaluate to what extent $John\_salary(low)$ is true as the degree in which the salary u_0 is considered to be low. Hence, allowing fuzzy constants in the language leads to treat formulas in a many-valued logical setting (that of Gödel many-valued logic in our framework), as opposed to the bivalued setting within classical possibilistic logic, with the unit interval $[0, 1]$ as a set of truth-values.

The aim of this paper is twofold: first to define DePGL$^+$, a possibilistic defeasible logic programming language that extends P-DeLP through the use of PGL$^+$, instead of (classical) possibilistic logic, in order to incorporate fuzzy constants and fuzzy unification, and after to propose a way to handle conflicting arguments in the context of the extended framework. To this end, the rest of the paper is structured as follows. First, we present the fundamentals of PGL$^+$. Then we define the DePGL$^+$ programming language. The next two sections focus on the characterization of arguments in DePGL$^+$ and the analysis of the notion of conflict among arguments in the context of our proposal. Next we discuss some problematic situations that may arise when trying to define the notion of warranted arguments in DePGL$^+$, and propose some solutions. Finally we discuss some related work and present the main conclusions we have obtained.

PGL$^+$: Overview

Possibilistic logic (Dubois, Lang, & Prade 1994) is a logic of uncertainty where a certainty degree between 0 and 1, interpreted as a lower bound of a necessity measure, is attached to each classical formula. In the propositional version, possibilistic formulas are pairs (φ, α) where φ is a proposition of classical logic and interpreted as specifying a constraint $N(\varphi) \geq \alpha$ on the necessity measure of φ. Possibilistic models are possibility distributions $\pi : \Omega \rightarrow [0, 1]$ on the set of classical (bivalued) interpretations Ω which rank them in terms of plausibility: w is at least as plausible as w' when $\pi(w) \geq \pi(w')$. If $\pi(w) = 1$ then w is considered as fully plausible, while if $\pi(w) = 0$ w is

*This is a proper extension of the paper "Modeling Defeasible Argumentation within a Possibilistic Logic Framework with Fuzzy Unification" to appear in the 11th IPMU International Conference 2006 (Paris, France).

considered as totally impossible. Then (φ, α) is satisfied by π, written $\pi \models (\varphi, \alpha)$ whenever $N_\pi(\varphi) \geq \alpha$, where $N_\pi(\varphi) = \inf\{1 - \pi(w) \mid w(\varphi) = 0\}$.

In (Alsinet & Godo 2000; 2001) the authors introduce PGL$^+$, an extension of possibilistic logic allowing to deal with some form of fuzzy knowledge and with an efficient and complete proof procedure for atomic deduction when clauses fulfill two kinds of constraints. Technically speaking, PGL$^+$ is a possibilistic logic defined on top of (a fragment of) Gödel infinitely-valued logic, allowing uncertainty qualification of predicates with imprecise, fuzzy constants, and allowing as well a form of graded unification between them. Next we provide some details.

The *basic components* of PGL$^+$ formulas are: a set of primitive propositions (fuzzy propositional variables) *Var*; a set S of *sorts* of constants; a set $\mathcal{C}$ of object *constants*, each having its sort; a set *Pred* of unary regular predicates, each one having a *type*; and *connectives* $\wedge, \rightarrow$. An *atomic formula* is either a primitive proposition from *Var* or of the form $p(A)$, where p is a predicate symbol from *Pred*, A is an object constant from $\mathcal{C}$ and the sort of A corresponds to the type of p. *Formulas* are Horn-rules of the form $p_1 \wedge \cdots \wedge p_k \rightarrow q$ with $k \geq 0$, where $p_1, \ldots, p_k, q$ are atomic formulas. A (weighted) *clause* is a pair of the form (φ, α), where φ is a Horn-rule and $\alpha \in [0, 1]$.

Remark *Since variables, quantifiers and function symbols are not allowed, the language of PGL$^+$ so defined remains in fact propositional. This allows us to consider only unary predicates since statements involving multiple (fuzzy) properties can be always represented in PGL$^+$ as a conjunction of atomic formulas. For instance, the statement "Mary is young and tall" can be represented in PGL$^+$ as $age\_Mary(young) \wedge height\_Mary(tall)$ instead of using a binary predicate involving two fuzzy constants like $age\&height\_Mary(young, tall)$.*

A many-valued *interpretation* for the language is a structure $w = (U, i, m)$, where: $U = \cup_{\sigma \in \mathcal{S}} U_\sigma$ is a collection of non-empty domains U_σ, one for each basic sort $\sigma \in \mathcal{S}$; $i = (i_{prop}, i_{pred})$, where $i_{prop} : Var \rightarrow [0, 1]$ maps each primitive proposition q into a value $i_{prop}(q) \in [0, 1]$ and $i_{pred} : Pred \rightarrow U$ maps a predicate p of type (σ) into a value $i_{pred}(p) \in U_\sigma$; and $m : \mathcal{C} \rightarrow [0, 1]^U$ maps an object constant A of sort σ into a normalized fuzzy set $m(A)$ on U_σ, with membership function $\mu_{m(A)} : U_\sigma \rightarrow [0, 1]$. [1]

The *truth value* of an atomic formula φ under an interpretation $w = (U, i, m)$, denoted by $w(\varphi) \in [0, 1]$, is defined as $w(q) = i_{prop}(q)$ for primitive propositions, and $w(p(A)) = \mu_{m(A)}(i_{pred}(p))$ for atomic predicates. The truth evaluation is extended to rules by means of interpreting the $\wedge$ connective by the min-conjunction and the $\rightarrow$ connective by the so-called Gödel's many-valued implication: $w(p_1 \wedge \cdots \wedge p_k \rightarrow q) = 1$ if $\min(w(p_1), \ldots, w(p_k)) \leq w(q)$, and $w(p_1 \wedge \cdots \wedge p_k \rightarrow q) = w(q)$ otherwise.

Note that the truth value $w(\varphi)$ will depend not only on the interpretation i_{pred} of predicate symbols that φ may contain, but also on the fuzzy sets assigned to fuzzy constants by m. Then, in order to define the possibilistic semantics, we need to fix a meaning for the fuzzy constants and to consider some extension of the standard notion of necessity measure for fuzzy events. The first is achieved by fixing a *context*. Basically a context is the set of interpretations sharing a common domain U and an interpretation of object constants m. So, given U and m, its associated context is just the set of interpretations $\mathcal{I}_{U,m} = \{w \mid w = (U, i, m)\}$ and, once fixed the context, $[\varphi]$ denotes the fuzzy set of models for a formula φ defining $\mu_{[\varphi]}(w) = w(\varphi)$, for all $w \in \mathcal{I}_{U,m}$.

Now, in a fixed context $\mathcal{I}_{U,m}$, a belief state (or *possibilistic model*) is determined by a normalized possibility distribution on $\mathcal{I}_{U,m}$, $\pi : \mathcal{I}_{U,m} \rightarrow [0, 1]$. Then, we say that π *satisfies* a clause (φ, α), written $\pi \models (\varphi, \alpha)$, iff the (suitable) necessity measure of the fuzzy set of models of φ with respect to π, denoted $N([\varphi] \mid \pi)$, is indeed at least α. Here, for the sake of soundness preservation, we take

$$N([\varphi] \mid \pi) = \inf_{w \in \mathcal{I}_{U,m}} \pi(w) \Rightarrow \mu_{[\varphi]}(w),$$

where $\Rightarrow$ is the reciprocal of Gödel's many-valued implication, defined as $x \Rightarrow y = 1$ if $x \leq y$ and $x \Rightarrow y = 1 - x$, otherwise. This necessity measure for fuzzy sets was proposed and discussed by Dubois and Prade (cf. (Dubois, Lang, & Prade 1994)). For example, according to this semantics, given a context $\mathcal{I}_{U,m}$ the formula

$$(age\_Peter(about\_35), 0.9)$$

is to be interpreted in PGL$^+$ as the following set of clauses with imprecise but non-fuzzy constants

$$\{(age\_Peter([about\_35]_\beta), \min(0.9, 1 - \beta)) : \beta \in [0, 1]\},$$

where $[about\_35]_\beta$ denotes the β-cut of the fuzzy set $m(about\_35)$. As usual, a set of clauses P is said to *entail* another clause (φ, α), written $P \models (\varphi, \alpha)$, iff every possibilistic model π satisfying all the clauses in P also satisfies (φ, α), and we say that a set of clauses P is *satisfiable* in the context determined by U and m if there exists a normalized possibility distribution $\pi : \mathcal{I}_{U,m} \rightarrow [0, 1]$ that satisfies all the clauses in P. Satisfiable clauses enjoy the following result (Alsinet 2003): If P is satisfiable and $P \models (\varphi, \alpha)$, with $\alpha > 0$, there exists at least an interpretation $w \in \mathcal{I}_{U,m}$ such that $w(\varphi) = 1$.

Finally, still in a context $\mathcal{I}_{U,m}$, the *degree of possibilistic entailment* of an atomic formula (or goal) φ by a set of clauses P, denoted by $\|\varphi\|_P$, is the greatest $\alpha \in [0, 1]$ such that $P \models (\varphi, \alpha)$. In (Alsinet 2003), it is proved that $\|\varphi\|_P = \inf\{N([\varphi] \mid \pi) \mid \pi \models P\}$.

The calculus for PGL$^+$ in a given context $\mathcal{I}_{U,m}$ is defined by the following set of inference rules:

Generalized resolution:

$$\frac{\begin{array}{c}(p \wedge s \rightarrow q(A), \alpha),\\ (q(B) \wedge t \rightarrow r, \beta)\end{array}}{(p \wedge s \wedge t \rightarrow r, \min(\alpha, \beta))} \text{ [GR], if } A \subseteq B$$

Fusion:

$$\frac{\begin{array}{c}(p(A) \wedge s \rightarrow q(D), \alpha),\\ (p(B) \wedge t \rightarrow q(E), \beta)\end{array}}{(p(A \cup B) \wedge s \wedge t \rightarrow q(D \cup E), \min(\alpha, \beta))} \text{ [FU]}$$

[1]Note that for each predicate symbol p, $i_{pred}(p)$ is the one and only value of the domain which satisfies p in that interpretation and that m prescribes for each constant A at least one value u_0 of the domain U_σ as fully compatible, i.e. such that $\mu_{m(A)}(u_0) = 1$.

Intersection:

$$\frac{(p(A),\alpha),(p(B),\beta)}{(p(A\cap B),\min(\alpha,\beta))}\ \text{[IN]}$$

Resolving uncertainty:

$$\frac{(p(A),\alpha)}{(p(A'),1)}\ \text{[UN]},\ \text{where } A'=\max(1-\alpha,A)$$

Semantical unification:

$$\frac{(p(A),\alpha)}{(p(B),\min(\alpha,N(B\mid A)))}\ \text{[SU]}$$

For each context $\mathcal{I}_{U,m}$, the above GR, FU, SU, IN and UN inference rules can be proved to be *sound* with respect to the possibilistic entailment of clauses. Moreover we shall also refer to the following weighted **modus ponens** rule, which can be seen as a particular case of the GR rule

$$\frac{\begin{array}{c}(p_1\wedge\ldots\wedge p_n\rightarrow q,\alpha),\\ (p_1,\beta_1),\ldots,(p_n,\beta_n)\end{array}}{(q,\min(\alpha,\beta_1,\ldots,\beta_n))}\ \text{[MP]}$$

The notion of *proof* in PGL$^+$, denoted by $\vdash$, is that of deduction by means of the triviality axiom and the PGL$^+$ inference rules. Given a context $\mathcal{I}_{U,m}$, the *degree of deduction* of a goal φ from a set of clauses P, denoted $|\varphi|_P$, is the greatest $\alpha\in[0,1]$ for which $P\vdash(\varphi,\alpha)$. Actually this notion of proof is complete for determining the degree of possibilistic entailment of a goal, i.e. $|\varphi|_P=\|\varphi\|_P$, for non-recursive and satisfiable programs P, called PGL$^+$ programs, under certain further conditions. Details can be found in (Alsinet & Godo 2001; Alsinet 2003).

The DePGL$^+$ programming language

As already pointed out our objective is to extend the P-DeLP programming language through the use of PGL$^+$ in order to incorporate fuzzy constants and fuzzy propositional variables; we will refer to this extension as *Defeasible* PGL$^+$, DePGL$^+$ for short. To this end, the base language of P-DeLP (Chesñevar *et al.* 2004) will be extended with fuzzy constants and fuzzy propositional variables, and arguments will have an attached necessity measure associated with the supported conclusion.

The DePGL$^+$ language $\mathcal{L}$ is defined over PGL$^+$ atomic formulas together with the connectives $\{\sim,\wedge,\leftarrow\}$. The symbol $\sim$ stands for *negation*. A *literal* $L\in\mathcal{L}$ is a PGL$^+$ atomic formula or its negation. A *rule* in $\mathcal{L}$ is a formula of the form $Q\leftarrow L_1\wedge\ldots\wedge L_n$, where $Q,L_1,\ldots,L_n$ are literals in $\mathcal{L}$. When $n=0$, the formula $Q\leftarrow$ is called a *fact* and simply written as Q. In the following, capital and lower case letters will denote literals and atoms in $\mathcal{L}$, respectively.

In argumentation frameworks, the negation connective allows to represent conflicts among pieces of information. In the frame of DePGL$^+$, the handling of negation deserves some explanation. In what regards negated propositional variables $\sim p$, the negation connective $\sim$ will not be considered as a proper Gödel negation. Rather, $\sim p$ will be treated as another propositional variable p', with a particular status with respect to p, since it will be only used to detect contradictions at the syntactical level. On the other hand, negated literals of the form $\sim p(A)$, where A is a fuzzy constant, will be handled in the following way.

As previously mentioned, fuzzy constants are *disjunctively* interpreted in PGL$^+$. For instance, consider the formula $speed(low)$. In each interpretation $I=(U,i,m)$, the predicate $speed$ is assigned a *unique* element $i(speed)$ of the corresponding domain. If low denotes a crisp interval of rpm's, say $[0,2000]$, then $speed(low)$ will be true iff such element belongs to this interval, i.e. iff $i(speed)\in[0,2000]$. Now, since the negated formula $\sim speed(low)$ is to be interpreted as "$\neg[\exists x\in low$ such that the engine speed is $x]$", which (under PGL$^+$ interpretations) amounts to "$[\exists x\notin low$ such that the engine speed is $x]$", it turns out that $\sim speed(low)$ is true iff $speed(\neg low)$ is true, where $\neg low$ denotes the complement of the interval $[0,2000]$ in the corresponding domain. Then, given a context $\mathcal{I}_{U,m}$, this leads us to understand a negated literal $\sim p(A)$ as another positive literal $p(\neg A)$, where the fuzzy constant $\neg A$ denotes the (fuzzy) complement of A, that is, where $\mu_{m(\neg A)}(u)=n(\mu_{m(A)}(u))$, for some suitable negation function n (usually $n(x)=1-x$).

Therefore, given a context $\mathcal{I}_{U,m}$, using the above interpretations of the negation, and interpreting the DePGL$^+$ arrow $\leftarrow$ as the PGL$^+$ implication $\rightarrow$, we can actually transform a DePGL$^+$ program P into a PGL$^+$ program, denoted as $\tau(P)$, and then, we can apply the deduction machinery of PGL$^+$ on $\tau(P)$ for automated proof purposes. From now on and for the sake of a simpler notation, we shall write $\Gamma\vdash_\tau(\varphi,\alpha)$ to denote $\tau(\Gamma)\vdash\tau((\varphi,\alpha))$, being Γ and (φ,α) DePGL$^+$ clauses. Moreover, we shall consider that the negation function n is implicitly determined by each context $\mathcal{I}_{U,m}$, i.e. the function m will interpret both fuzzy constants A and their complement (negation) $\neg A$.

Arguments in DePGL$^+$

In the last sections we formalized the many-valued and the possibilistic semantics of the underlying logic of DePGL$^+$. In this section we formalize the procedural mechanism for building arguments in DePGL$^+$.

We distinguish between *certain* and *uncertain* DePGL$^+$ clauses. A DePGL$^+$ clause (φ,α) will be referred as certain when $\alpha=1$ and uncertain, otherwise. Given a context $\mathcal{I}_{U,m}$, a set of DePGL$^+$ clauses Γ will be deemed as *contradictory*, denoted $\Gamma\vdash_\tau\bot$, when

(i) either $\Gamma\vdash_\tau(q,\alpha)$ and $\Gamma\vdash_\tau(\sim q,\beta)$, with $\alpha>0$ and $\beta>0$, for some atom q in $\mathcal{L}$,

(ii) or $\Gamma\vdash_\tau(p(A),\alpha)$ with $\alpha>0$, for some predicate p and some fuzzy constant A such that $m(A)$ is non-normalized.

Notice that in the latter case, $\tau(\Gamma)$ is not satisfiable and there exist $\Gamma_1\subset\tau(\Gamma)$ and $\Gamma_2\subset\tau(\Gamma)$ such that Γ_1 and Γ_2 are satisfiable and $|p(B)|_{\Gamma_1}>0$ and $|p(C)|_{\Gamma_2}>0$, with $A=B\cap C$.

Example 1 *Consider the set of clauses* $\Gamma=\{(q,0.8),(r,1),(p(A)\leftarrow q,0.5),(p(B)\leftarrow q\wedge r,0.3)\}$. *Then,* $\Gamma\vdash_\tau$

$(p(A), 0.5)$ *and* $\Gamma \vdash_\tau (p(B), 0.3)$*, and, by the IN inference rule,* $\Gamma \vdash_\tau (p(A \cap B), 0.3)$*. Hence, in a particular context* $\mathcal{I}_{U,m}$*,* Γ *is contradictory as soon as* $m(A) \cap m(B)$ *is a non-normalized fuzzy set whereas, for instance,* $\Gamma \backslash \{(r, 1)\}$ *is satisfiable.*

A DePGL$^+$ program is a set of clauses in $\mathcal{L}$ in which we distinguish certain from uncertain information. As additional requirement, certain knowledge is required to be non-contradictory and the corresponding PGL$^+$ program (by means of transformation τ) is required to satisfy the modularity constraint (Alsinet & Godo 2001; Alsinet 2003). Formally: Given a context $\mathcal{I}_{U,m}$, a *DePGL$^+$ program* $\mathcal{P}$ is a pair (Π, Δ), where Π is a non-contradictory finite set of certain clauses, Δ is a finite set of uncertain clauses, and $\tau(\Pi \cup \Delta)$ satisfies the modularity constraint.

The requirement of the modularity constraint of a DePGL$^+$ program ensures that all (explicit and hidden) clauses of programs are considered. Indeed, since fuzzy constants are interpreted as (flexible) restrictions on an existential quantifier, atomic formulas clearly express disjunctive information. For instance, when $A = \{a_1, \ldots, a_n\}$, $p(A)$ is equivalent to the disjunction $p(a_1) \vee \cdots \vee p(a_n)$. Then, when parts of this (hidden) disjunctive information occur in the body of several program formulas we also have to consider all those new formulas that can be obtained through a completion process of the program which is based on the RE and FU inference rules.

Example 2 *(Adapted from (Chesñevar* et al. *2004)) Consider an intelligent agent controlling an engine with three switches* $sw1$*,* $sw2$ *and* $sw3$*. These switches regulate different features of the engine, such as pumping system, speed, etc. The agent's generic (and incomplete) knowledge about how this engine works is the following:*

- *If the pump is clogged, then the engine gets no fuel.*
- *When* $sw1$ *is on, apparently fuel is pumped properly.*
- *When fuel is pumped, fuel seems to work ok.*
- *When* $sw2$ *is on, usually oil is pumped.*
- *When oil is pumped, usually it works ok.*
- *When there is oil and fuel, normally the engine is ok.*
- *When there is heat, the engine is almost sure not ok.*
- *When there is heat, normally there are oil problems.*
- *When fuel is pumped and speed is low, there are reasons to believe that the pump is clogged.*
- *When* $sw2$ *is on, usually speed is low.*
- *When* $sw2$ *and* $sw3$ *are on, usually speed is not low.*
- *When* $sw3$ *is on, normally fuel is ok.*

Suppose also that the agent knows some particular facts about the current state of the engine:

- $sw1$*,* $sw2$ *and* $sw3$ *are on, and*
- *the temperature is around 31°C.*

This knowledge can be modelled by the program $\mathcal{P}_{engine}$ *shown in Fig. 1. Note that uncertainty is assessed in terms of different necessity degrees and vague knowledge is represented by means of fuzzy constants (*low*,* $around\_31$*,* $high$*).*

Next we introduce the notion of *argument* in DePGL$^+$. Informally, an argument for a literal (goal) Q with necessity degree α is a tentative (as it relies to some extent on uncertain, possibilistic information) proof for (Q, α) .

(1)	$(\sim fuel\_ok \leftarrow pump\_clog, 1)$
(2)	$(pump\_fuel \leftarrow sw1, 0.6)$
(3)	$(fuel\_ok \leftarrow pump\_fuel, 0.85)$
(4)	$(pump\_oil \leftarrow sw2, 0.8)$
(5)	$(oil\_ok \leftarrow pump\_oil, 0.8)$
(6)	$(engine\_ok \leftarrow fuel\_ok \wedge oil\_ok, 0.6)$
(7)	$(\sim engine\_ok \leftarrow temp(high), 0.95)$
(8)	$(\sim oil\_ok \leftarrow temp(high), 0.9)$
(9)	$(pump\_clog \leftarrow pump\_fuel \wedge speed(low), 0.7)$
(10)	$(speed(low) \leftarrow sw2, 0.8)$
(11)	$(\sim speed(low) \leftarrow sw2, sw3, 0.8)$
(12)	$(fuel\_ok \leftarrow sw3, 0.9)$
(13)	$(sw1, 1)$
(14)	$(sw2, 1)$
(15)	$(sw3, 1)$
(16)	$(temp(around\_31), 0.85)$

Figure 1: DePGL$^+$ program $\mathcal{P}_{eng}$ (example 2)

Definition 3 (Argument) *Given a context* $\mathcal{I}_{U,m}$ *and a DePGL$^+$ program* $\mathcal{P} = (\Pi, \Delta)$*, a set* $\mathcal{A} \subseteq \Delta$ *of uncertain clauses is an* argument *for a goal* Q *with necessity degree* $\alpha > 0$*, denoted* $\langle \mathcal{A}, Q, \alpha \rangle$*, iff:*
(1) $\Pi \cup \mathcal{A} \vdash_\tau (Q, \alpha)$*;*
(2) $\Pi \cup \mathcal{A}$ *is non contradictory; and*
(3) $\mathcal{A}$ *is minimal wrt set inclusion, i.e. there is no* $\mathcal{A}_1 \subset \mathcal{A}$ *satisfying (1) and (2).*

Let $\langle \mathcal{A}, Q, \alpha \rangle$ and $\langle \mathcal{S}, R, \beta \rangle$ be two arguments. We will say that $\langle \mathcal{S}, R, \beta \rangle$ is a *subargument* of $\langle \mathcal{A}, Q, \alpha \rangle$ iff $\mathcal{S} \subseteq \mathcal{A}$. Notice that the goal R may be a subgoal associated with the goal Q in the argument $\mathcal{A}$.

Given a context $\mathcal{I}_{U,m}$, the set of arguments for a DePGL$^+$ program $\mathcal{P} = (\Pi, \Delta)$ can be found by the iterative application of the following construction rules:

1) Building arguments from facts (INTF):

$$\frac{(Q, 1)}{\langle \emptyset, Q, 1 \rangle} \qquad \frac{(Q, \alpha), \Pi \cup \{(Q, \alpha)\} \not\vdash_\tau \perp, \alpha < 1}{\langle \{(Q, \alpha)\}, Q, \alpha \rangle}$$

for any $(Q, 1) \in \Pi$ and any $(Q, \alpha) \in \Delta$.

2) Building Arguments by SU (SUA):

$$\frac{\langle \mathcal{A}, p(A), \alpha \rangle}{\langle \mathcal{A}, p(B), \min(\alpha, N(m(B) \mid m(A))) \rangle}$$

if $N(m(B) \mid m(A)) \neq 0$.

3) Building Arguments by UN (UNA):

$$\frac{\langle \mathcal{A}, p(A), \alpha \rangle}{\langle \mathcal{A}, p(A'), 1 \rangle}$$

where $m(A') = \max(1 - \alpha, m(A))$.

4) Building Arguments by IN (INA):

$$\frac{\langle \mathcal{A}_1, p(A), \alpha \rangle, \ \langle \mathcal{A}_2, p(B), \beta \rangle, \ \Pi \cup \mathcal{A}_1 \cup \mathcal{A}_2 \not\vdash_\tau \perp}{\langle \mathcal{A}_1 \cup \mathcal{A}_2, p(A \cap B), \min(\alpha, \beta) \rangle}$$

5) Building Arguments by MP (MPA):

$$\frac{\langle \mathcal{A}_1, L_1, \alpha_1\rangle \quad \langle \mathcal{A}_2, L_2, \alpha_2\rangle \; \ldots \; \langle \mathcal{A}_k, L_k, \alpha_k\rangle \quad (L_0 \leftarrow L_1 \wedge L_2 \wedge \ldots \wedge L_k, 1) \quad \Pi \cup \bigcup_{i=1}^{k} \mathcal{A}_i \not\vdash_\tau \bot}{\langle \bigcup_{i=1}^{k} \mathcal{A}_i, L_0, \beta\rangle}$$

for any certain rule $(L_0 \leftarrow L_1 \wedge L_2 \wedge \ldots \wedge L_k, 1) \in \Pi$, with $\beta = min(\alpha_1, \ldots, \alpha_k)$.

$$\frac{\langle \mathcal{A}_1, L_1, \alpha_1\rangle \quad \langle \mathcal{A}_2, L_2, \alpha_2\rangle \; \ldots \; \langle \mathcal{A}_k, L_k, \alpha_k\rangle \quad (L_0 \leftarrow L_1 \wedge L_2 \wedge \ldots \wedge L_k, \gamma), \text{ with } \gamma < 1 \quad \Pi \cup \{(L_0 \leftarrow L_1 \wedge L_2 \wedge \ldots \wedge L_k, \gamma)\} \cup \bigcup_{i=1}^{k} \mathcal{A}_i \not\vdash_\tau \bot}{\langle \bigcup_{i=1}^{k} \mathcal{A}_i \cup \{(L_0 \leftarrow L_1 \wedge L_2 \wedge \ldots \wedge L_k, \gamma)\}, L_0, \beta\rangle}$$

for any weighted rule $(L_0 \leftarrow L_1 \wedge L_2 \wedge \ldots \wedge L_k, \gamma) \in \Delta$, with $\beta = min(\alpha_1, \ldots, \alpha_k, \gamma)$.

The basic idea with the argument construction procedure is to keep a trace of the set $\mathcal{A} \subseteq \Delta$ of all uncertain information in the program $\mathcal{P}$ used to derive a given goal Q with necessity degree α. Appropriate preconditions ensure that the proof obtained always ensures the non-contradictory constraint of arguments wrt the certain knowledge Π of the program. Given a context $\mathcal{I}_{U,m}$ and a DePGL$^+$ program $\mathcal{P}$, rule INTF allows to construct arguments from facts. An empty argument can be obtained for any certain fact in $\mathcal{P}$. An argument concluding an uncertain fact (Q, α) in $\mathcal{P}$ can be derived whenever assuming (Q, α) is not contradictory wrt the set Π in $\mathcal{P}$. Rules SUA and UNA accounts for semantical unification and resolving uncertainty, respectively. As both rules do not combine new uncertain knowledge, we do not need to check the non-contradictory constraint. Rule INA applies intersection between previously argumented goals provided that the resulting intersection is non contradictory wrt Π. Rules MPA account for the use of modus ponens, both with certain and defeasible rules. Note they assume the existence of an argument for every literal in the antecedent of the rule. Then, in a such a case, the MPA rule is applicable whenever no contradiction results when putting together Π, the sets $\mathcal{A}_1, \ldots, \mathcal{A}_k$ corresponding to the arguments for the antecedents of the rule and the rule $(L_0 \leftarrow L_1 \wedge L_2 \wedge \ldots \wedge L_k, \gamma)$ when $\gamma < 1$.

Example 4 *Consider the program $\mathcal{P}_{eng}$ in Example 2, where $temp(\cdot)$ is a unary predicate of type* (degrees), *$speed(\cdot)$ is a unary predicate of type* (rpm), heat *and* around_31 *are two object constants of type* degrees, *and* low *is an object constant of type* rpm. *Further, consider the context $\mathcal{I}_{U,m}$ such that:*

- $U = \{U_{\texttt{degrees}} = [-100, 100]\ ^oC,\ U_{\texttt{rpm}} = [0, 200]\}$;
- $m(high) = [28, 30, 100, 100]$[2],
  $m(around\_31) = [26, 31, 31, 36]$,
  $m(low) = [10, 15, 25, 30]$, *and*
  $m(\neg low) = 1 - m(low)$.

Then the following arguments can be derived from $\mathcal{P}_{eng}$:

[2]We represent a trapezoidal fuzzy set as $[t_1; t_2; t_3; t_4]$, where the interval $[t_1, t_4]$ is the support and the interval $[t_2, t_3]$ is the core.

1. *The argument $\langle \mathcal{B}, fuel\_ok, 0.6\rangle$ can be derived as follows:*
   i) $\langle \emptyset, sw1, 1\rangle$ *from (13) via INTF.*
   ii) $\langle \mathcal{B}', pump\_fuel, 0.6\rangle$ *from (2) and i) via MPA.*
   iii) $\langle \mathcal{B}, fuel\_ok, 0.6\rangle$ *from (3) and ii) via MPA.*
   where $\mathcal{B}' = \{(pump\_fuel \leftarrow sw1, 0.6)\}$ *and* $\mathcal{B} = \mathcal{B}' \cup \{(fuel\_ok \leftarrow pump\_fuel, 0.85)\}$.
2. *Similarly, the argument $\langle \mathcal{C}_1, oil\_ok, 0.8\rangle$ can be derived using the rules (15), (4) and (5) via INTC, MPA, and MPA respectively, with: $\mathcal{C}_1 = \{(pump\_oil \leftarrow sw2, 0.8);$ $(oil\_ok \leftarrow pump\_oil, 0.8)\}$.*
3. *The argument $\langle \mathcal{A}_1, engine\_ok, 0.6\rangle$ can be derived as follows:*
   i) $\langle \mathcal{B}, fuel\_ok, 0.6\rangle$ *as shown above.*
   ii) $\langle \mathcal{C}_1, oil\_ok, 0.8\rangle$ *as shown above.*
   iii) $\langle \mathcal{A}_1, engine\_ok, 0.6\rangle$ *from i), ii), (6) via MPA.*
   with $\mathcal{A}_1 = \{(engine\_ok \leftarrow fuel\_ok \wedge oil\_ok, 0.6)\} \cup \mathcal{B} \cup \mathcal{C}_1$. *Note that $\langle \mathcal{C}_1, oil\_ok, 0.8\rangle$ and $\langle \mathcal{B}, fuel\_ok, 0.6\rangle$ are subarguments of $\langle \mathcal{A}_1, engine\_ok, 0.6\rangle$.*
4. *One can also derive the argument $\langle \mathcal{C}_2, \sim oil\_ok, 0.8\rangle$, where $\mathcal{C}_2 = \{(temp(around\_31), 0.85), (\sim oil\_ok \leftarrow temp(high), 0.9)\}$, as follows:*
   i) $\langle \{(temp(around\_31),\ 0.85)\}, temp(around\_31), 0.85\rangle$ *from (16) via INTF.*
   ii) $\langle \{(temp(around\_31),\ 0.85)\}, temp(high), 0.8\rangle$ *from i) via SUA, where $N(high \mid around\_31) = 0.8$ and $0.8 = \min(0.85, 0.8)$.*
   iii) $\langle \mathcal{C}_2, \sim oil\_ok, 0.8\rangle$ *from i), ii), (6) via MPA.*
5. *Similarly, an argument $\langle \mathcal{A}_2, \sim engine\_ok, 0.8\rangle$ can be derived using the rules (16) and (7) via INTF, SUA, and MPA, with*
   $\mathcal{A}_2 = \{(temp(around\_31), 0.85);$
   $(\sim engine\_ok \leftarrow temp(high), 0.95)\}$.

Counter-argumentation and defeat in DePGL$^+$

Given a program and a particular context, it can be the case that there exist conflicting arguments for one literal and its negation. For instance, in the above example, $\langle \mathcal{A}_1, engine\_ok, 0.6\rangle$ and $\langle \mathcal{A}_2, \sim engine\_ok, 0.8\rangle$, and $\langle \mathcal{C}_1, oil\_ok, 0.8\rangle$ and $\langle \mathcal{C}_2, \sim oil\_ok, 0.8\rangle$, and thus, the program $\mathcal{P}_{eng}$ considering the context $\mathcal{I}_{U,m}$ is contradictory. Therefore, it is necessary to define a formal framework for solving conflicts among arguments in DePGL$^+$. This is formalized next by the notions of counterargument and defeat, based on the same ideas used in P-DeLP (Chesñevar *et al.* 2004) but incorporating the treatment of fuzzy constants.

Definition 5 (Counterargument) *Let $\mathcal{P}$ be a DePGL$^+$ program, let $\mathcal{I}_{U,m}$ be a context, and let $\langle \mathcal{A}_1, Q_1, \alpha_1\rangle$ and $\langle \mathcal{A}_2, Q_2, \alpha_2\rangle$ be two arguments wrt $\mathcal{P}$ in the context $\mathcal{I}_{U,m}$. We will say that $\langle \mathcal{A}_1, Q_1, \alpha_1\rangle$ counterargues $\langle \mathcal{A}_2, Q_2, \alpha_2\rangle$ iff there exists a subargument (called* disagreement subargument*) $\langle \mathcal{S}, Q, \beta\rangle$ of $\langle \mathcal{A}_2, Q_2, \alpha_2\rangle$ such that $Q = \sim Q_1$*[3].

[3]For a given goal Q, we write $\sim Q$ as an abbreviation to denote "$\sim q$" if $Q \equiv q$ (resp.,"$\sim q(A)$" if $Q \equiv q(A)$) and "q" if $Q \equiv \sim q$ (resp., "$q(A)$" if $Q \equiv \sim q(A)$).

Since arguments rely on uncertain and hence defeasible information, conflicts among arguments may be resolved by comparing their strength and deciding which argument is defeated by which one. Therefore, a notion of defeat amounts to establish a *preference criterion* on conflicting arguments. In our framework, following (Chesñevar *et al.* 2004), it seems natural to define it on the basis of necessity degrees associated with arguments.

Definition 6 (Defeat) *Let $\mathcal{P}$ be a DePGL$^+$ program, let $\mathcal{I}_{U,m}$ be a context, and let $\langle \mathcal{A}_1, Q_1, \alpha_1 \rangle$ and $\langle \mathcal{A}_2, Q_2, \alpha_2 \rangle$ be two arguments wrt $\mathcal{P}$ in the context $\mathcal{I}_{U,m}$. We will say that $\langle \mathcal{A}_1, Q_1, \alpha_1 \rangle$* defeats *$\langle \mathcal{A}_2, Q_2, \alpha_2 \rangle$ (or equivalently $\langle \mathcal{A}_1, Q_1, \alpha_1 \rangle$ is a* defeater *for $\langle \mathcal{A}_2, Q_2, \alpha_2 \rangle$) iff:*

(1) the argument $\langle \mathcal{A}_1, Q_1, \alpha_1 \rangle$ counterargues the argument $\langle \mathcal{A}_2, Q_2, \alpha_2 \rangle$ with disagreement subargument $\langle \mathcal{A}, Q, \alpha \rangle$; and

(2) either it holds that $\alpha_1 > \alpha$, in which case $\langle \mathcal{A}_1, Q_1, \alpha_1 \rangle$ will be called a proper defeater *for $\langle \mathcal{A}_2, Q_2, \alpha_2 \rangle$, or $\alpha_1 = \alpha$, in which case $\langle \mathcal{A}_1, Q_1, \alpha_1 \rangle$ will be called a* blocking defeater *for $\langle \mathcal{A}_2, Q_2, \alpha_2 \rangle$.*

Following Examples 2 and 4, we have that argument $\langle \mathcal{A}_2, \sim engine\_ok, 0.8 \rangle$ is a defeater of argument $\langle \mathcal{A}_1, engine\_ok, 0.6 \rangle$ while $\langle \mathcal{C}_2, \sim oil\_ok, 0.8 \rangle$ is a blocking defeater of $\langle \mathcal{C}_1, oil\_ok, 0.8 \rangle$.

Computing warranted arguments in DePGL$^+$

As in most argumentation systems, a main goal in DePGL$^+$ is to devise a procedure to determine whether a given argument $\langle \mathcal{A}, Q, \alpha \rangle$ is *warranted* (or ultimately accepted) wrt a program $\mathcal{P}$. Intuitively, an argument $\langle \mathcal{A}, Q, \alpha \rangle$ is warranted when

1. it has no defeaters, or
2. every defeater for $\langle \mathcal{A}, Q, \alpha \rangle$ is on its turn defeated by another argument which is warranted.

In P-DeLP this is done by an exhaustive dialectical analysis of all argumentation lines rooted in a given argument (see (Chesñevar *et al.* 2004) for details) which can be efficiently performed by means of a top-down algorithm, as described in (Chesñevar, Simari, & Godo 2005). For instance, given the following simple P-DeLP program $\mathcal{P} = \{(p, 0.45), (\sim p, 0.7)\}$, a short dialectical analysis would conclude that the argument $A = \langle \{(\sim p, 0.7)\}, \sim p, 0.7 \rangle$ is warranted.

However, even with similar simple programs, the situation DePGL$^+$ gets more involved. Indeed, in order to provide DePGL$^+$ with a similar dialectical analysis, due to the disjunctive interpretation of fuzzy constants and their associated fuzzy unification mechanism, new blocking situations between arguments have to be considered as we show in the following example.

Example 7 *Consider the DePGL$^+$ program*

$$\mathcal{P} = \{(temp(around\_31), 0.45), (temp(between\_25\_30), 0.7)\}$$

where $temp(\cdot)$ is a unary predicate of type (degrees)*, and the context $\mathcal{I}_{U,m}$ with $U = \{U_{\texttt{degrees}} = [-100, 100]\ ^oC\}$ and*

$m(around\_31) = [26, 31, 31, 36]$,
$m(between\_25\_30) = [20, 25, 30, 35]$,
$m(\neg around\_31) = 1 - m(around\_31)$, *and*
$m(\neg between\_25\_30) = 1 - m(between\_25\_30)$.

Consider the following sets of clauses:

$\mathcal{A}_1 = \{(temp(around\_31), 0.45)\}$
$\mathcal{A}_2 = \{(temp(between\_25\_30), 0.7)\}$.

Within the context $\mathcal{I}_{U,m}$, the arguments

$A_1 = \langle \mathcal{A}_1, temp(around\_31), 0.45 \rangle$,
$A_2 = \langle \mathcal{A}_2, temp(between\_25\_30), 0.7 \rangle$,

can be derived from $\mathcal{P}$, but notice that $m(around\_31) \cap m(between\_25\_30)$ is a non-normalized fuzzy set. However, since we have

$N(m(\neg around\_31) \mid m(between\_25\_30)) = 0$
$N(m(\neg between\_25\_30) \mid m(around\_31)) = 0$,

using the SUA procedural rule, one can only derive arguments for the negated literals $\sim temp(around\_31)$ and $\sim temp(between\_25\_30)$ with necessity degree 0. Hence, neither A_1 nor A_2 has a proper defeater. Then, in this particular context, neither A_1 nor A_2 can be warranted, and thus A_1 acts as a blocking argument for A_2, and viceversa.

Remark that the unification degree, or the partial matching, between fuzzy constants depends on the context we are considering. For instance, if for the above context $\mathcal{I}_{U,m}$ we consider the Gödel negation instead of the involutive negation; i.e.,

$$m(\neg A)(t) = \begin{cases} 1, & \text{if } m(A)(t) = 0 \\ 0, & \text{otherwise} \end{cases}$$

for any fuzzy constant A, we get that

$$N(m(\neg around\_31) \mid m(between\_25\_30)) = 0.2$$
$$N(m(\neg between\_25\_30) \mid m(around\_31)) = 0.2$$

However, as $0.2 < 0.45$ and $0.2 < 0.7$, in this new particular context neither A_1 nor A_2 can be warranted as well.

Therefore we introduce the following notion of pair of blocking arguments.

Definition 8 (Blocking arguments) *Let $\mathcal{P}$ be a DePGL$^+$ program, let $\mathcal{I}_{U,m}$ be a context, and let $\langle \mathcal{A}_1, q(A), \alpha_1 \rangle$ and $\langle \mathcal{A}_2, q(B), \alpha_2 \rangle$ be two arguments wrt $\mathcal{P}$ in the context $\mathcal{I}_{U,m}$. We will say that $\langle \mathcal{A}_1, q(A), \alpha_1 \rangle$* blocks *$\langle \mathcal{A}_2, q(B), \alpha_2 \rangle$, and viceversa, when*

1. $m(A) \cap m(B)$ is a non-normalized fuzzy set; and
2. $N(m(\neg A) \mid m(B)) < \alpha_1$ and $N(m(\neg B) \mid m(A)) < \alpha_2$.

By extension, if $\langle \mathcal{A}_1, Q_1, \alpha_1 \rangle$ is a subargument of $\langle \mathcal{A}, Q, \alpha \rangle$ and $\langle \mathcal{A}_1, Q_1, \alpha_1 \rangle$ and $\langle \mathcal{A}_2, Q_2, \alpha_2 \rangle$ are a pair of blocking arguments, argument $\langle \mathcal{A}, Q, \alpha \rangle$ cannot be warranted and $\langle \mathcal{A}_2, Q_2, \alpha_2 \rangle$ is a *blocking argument* for $\langle \mathcal{A}, Q, \alpha \rangle$.

Given a DePGL$^+$ program and a particular context, there may exist both multiple blocking arguments and multiple proper defeaters for a same argument, all of them derived from a same set of clauses by applying the semantical unification procedural rule SUA as we show in the following example.

Example 9 *Consider the DePGL$^+$ program $\mathcal{P}$ and the context $\mathcal{I}_{U,m}$ of Example 7. Let*

$$\mathcal{A}_3 = \{(temp(about\_25), 0.9)\},$$

and let $\mathcal{P}' = \mathcal{P} \cup \mathcal{A}_3$ be a new program. Further, consider two new fuzzy constants "between_31_32" and "about_25_ext". The three new fuzzy constants are interpreted in the context $\mathcal{I}_{U,m}$ as

$$m(about\_25) = [24, 25, 25, 26],$$
$$m(\neg about\_25) = 1 - m(about\_25),$$
$$m(between\_31\_32) = [26, 31, 32, 37], \text{ and}$$
$$m(about\_25\_ext) = [24, 25, 25, 32].$$

Notice that arguments A_1 and A_2 from Example 7 are still arguments with respect the new program $\mathcal{P}'$. Now, in the frame of the program $\mathcal{P}'$, from the argument A_1 and by applying the SUA procedural rule, we can build the argument

$$A_3 = \langle \mathcal{A}_1, temp(between\_31\_32), 0.45\rangle,$$

since $N(m(between\_31\_32) \mid m(around\_31)) = 1$. One can easily check that A_3 and A_2 are a pair of blocking arguments. Moreover, as $m(around\_31) \leq m(between\_31\_32)$, i.e. "around_31" is more specific *than "between_31_32", we have $N(m(\neg between\_25\_30) \mid m(around\_31)) \geq N(m(\neg between\_25\_30) \mid m(between\_31\_32))$, and thus, the argument A_3 can be considered as a redundant blocking argument for the argument A_2.*

On the other hand, the argument

$$A_4 = \langle \mathcal{A}_3, temp(about\_25), 0.9\rangle,$$

can be derived from $\mathcal{P}'$. Then, from the argument A_4 and by applying the SUA procedural rule, we can build the argument

$$A_5 = \langle \mathcal{A}_3, {\sim} temp(around\_31), 0.9\rangle,$$

since $N(m(\neg around\_31) \mid m(about\_25)) = 1$, and thus, the argument A_5 is a proper defeater for the argument A_1. Now, from the argument A_4 and by applying the SUA procedural rule, we can build the argument

$$A_6 = \langle \mathcal{A}_3, temp(about\_25\_ext), 0.9\rangle,$$

since $N(m(about\_25\_ext) \mid m(about\_25)) = 1$. Finally, from the argument A_6 and by applying the SUA procedural rule, we can build the argument

$$A_7 = \langle \mathcal{A}_3, {\sim} temp(around\_31), 0.5\rangle,$$

since $N(m(\neg around\_31) \mid m(about\_25\_ext)) = 0.5$, and thus, the argument A_7 is a proper defeater for the argument A_1. However, as arguments A_5 and A_7 have been computed both from the same specific information of the program and $0.9 > 0.5$, the argument A_7 can be considered as a redundant proper defeater for the argument A_1.

Therefore, if we aim at an efficient procedure for computing warrants (based on an exhaustive dialectical analysis of all argumentation lines), we have to avoid for a given argument both redundant blocking arguments and redundant proper defeaters . According to the above discussion, we introduce the following definitions of redundant blocking arguments and defeaters.

Definition 10 (Redundant blocking arguments) *Let $\mathcal{P}$ be a DePGL$^+$ program, let $\mathcal{I}_{U,m}$ be a context, and let $\langle \mathcal{A}_1, p(A), \alpha_1\rangle$ and $\langle \mathcal{A}_2, p(B), \alpha_2\rangle$ be a pair of blocking arguments wrt $\mathcal{P}$ in the context $\mathcal{I}_{U,m}$. We will say that $\langle \mathcal{A}_2, p(B), \alpha_2\rangle$ is a* redundant blocking argument *for $\langle \mathcal{A}_1, p(A), \alpha_1\rangle$ iff there exists an argument $\langle \mathcal{A}_2, p(C), 1\rangle$ such that:*

1. *$\langle \mathcal{A}_1, p(A), \alpha_1\rangle$ and $\langle \mathcal{A}_2, p(C), 1\rangle$ are a pair of blocking arguments; and*
2. *$m(C) \leq \max(1 - \alpha_2, m(B))$.*

Definition 11 (Redundant defeater) *Let $\mathcal{P}$ be a DePGL$^+$ program, let $\mathcal{I}_{U,m}$ be a context, and let $\langle \mathcal{A}_1, Q_1, \alpha_1\rangle$ and $\langle \mathcal{A}_2, Q_2, \alpha_2\rangle$ be two arguments wrt $\mathcal{P}$ in the context $\mathcal{I}_{U,m}$ such that $\langle \mathcal{A}_1, Q_1, \alpha_1\rangle$ is a proper defeater for $\langle \mathcal{A}_2, Q_2, \alpha_2\rangle$. We will say that $\langle \mathcal{A}_1, Q_1, \alpha_1\rangle$ is a* redundant defeater *for $\langle \mathcal{A}_2, Q_2, \alpha_2\rangle$ iff there exists an argument $\langle \mathcal{A}_1, Q_1, \alpha\rangle$ such that:*

1. *$\langle \mathcal{A}_1, Q_1, \alpha\rangle$ is is a proper defeater for $\langle \mathcal{A}_2, Q_2, \alpha_2\rangle$); and*
2. *$\alpha_1 < \alpha$.*

At this point we are ready to formalize the notion of argumentation line in the framework of DePGL$^+$. An *argumentation line* starting in an argument $\langle \mathcal{A}_0, Q_0, \alpha_0\rangle$ is a sequence of arguments

$$\lambda = [\langle \mathcal{A}_0, Q_0, \alpha_0\rangle, \langle \mathcal{A}_1, Q_1, \alpha_1\rangle, \ldots, \langle \mathcal{A}_n, Q_n, \alpha_n\rangle, \ldots]$$

that can be thought of as an exchange of arguments between two parties, a *proponent* (evenly-indexed arguments) and an *opponent* (oddly-indexed arguments). Each $\langle \mathcal{A}_i, Q_i, \alpha_i\rangle$ is *either a defeater or a blocking argument* for the previous argument $\langle \mathcal{A}_{i-1}, Q_{i-1}, \alpha_{i-1}\rangle$ in the sequence, $i > 0$. In order to avoid *fallacious* reasoning, argumentation theory imposes additional constraints on such an argument exchange to be considered rationally acceptable wrt a DePGL$^+$ program $\mathcal{P}$ and a context $\mathcal{I}_{U,m}$, namely:

1. **Non-contradiction:** given an argumentation line λ, the set of arguments of the proponent (resp. opponent) should be *non-contradictory* wrt $\mathcal{P}$ and $\mathcal{I}_{U,m}$.
2. **Progressive argumentation:** every[4] blocking defeater and blocking argument $\langle \mathcal{A}_i, Q_i, \alpha_i\rangle$ in λ, $i > 0$, is defeated by a proper defeater $\langle \mathcal{A}_{i+1}, Q_{i+1}, \alpha_{i+1}\rangle$ in λ.
3. **Non-redundancy:** every proper defeater and blocking argument $\langle \mathcal{A}_i, Q_i, \alpha_i\rangle$ in λ, $i > 0$, is a non-redundant defeater, resp. a non-redundant blocking argument, for the previous argument $\langle \mathcal{A}_{i-1}, Q_{i-1}, \alpha_{i-1}\rangle$ in λ; i.e. $\langle \mathcal{A}_i, Q_i, \alpha_i\rangle$ is the best proper defeater or the most specific blocking argument one can consider from a given set of clauses.

The first condition disallows the use of contradictory information on either side (proponent or opponent). The second condition enforces the use of a proper defeater to defeat an argument which acts as a blocking defeater or as a blocking argument. An argumentation line satisfying restrictions

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

(1) and (2) is called *acceptable*, and can be proven to be finite. Finally, since we consider programs with a finite set of clauses, the last condition ensures that we have a computable number of argumentations lines.

Given a program $\mathcal{P}$, a context $\mathcal{I}_{U,m}$ and an argument $\langle \mathcal{A}_0, Q_0, \alpha_0 \rangle$, the set of all acceptable argumentation lines starting in $\langle \mathcal{A}_0, Q_0, \alpha_0 \rangle$ accounts for a whole dialectical analysis for $\langle \mathcal{A}_0, Q_0, \alpha_0 \rangle$ (i.e. all possible dialogues rooted in $\langle \mathcal{A}_0, Q_0, \alpha_0 \rangle$, formalized as a *dialectical tree*[5]).

Definition 12 (Warrant) *Given a program $\mathcal{P} = (\Pi, \Delta)$, a context $\mathcal{I}_{U,m}$, and a goal Q, we will say that Q is* warranted *wrt $\mathcal{P}$ in the context $\mathcal{I}_{U,m}$ with a* maximum necessity degree *α iff there exists an argument of the form $\langle \mathcal{A}, Q, \alpha \rangle$, for some $\mathcal{A} \subseteq \Delta$, such that:*

1. *every acceptable argumentation line starting with $\langle \mathcal{A}, Q, \alpha \rangle$ has an odd number of arguments; i.e. every argumentation line starting with $\langle \mathcal{A}, Q, \alpha \rangle$ finishes with an argument proposed by the proponent which is in favor of Q with at least a necessity degree α; and*
2. *for each argument of the form $\langle \mathcal{A}_1, Q, \beta \rangle$, with $\beta > \alpha$, there exists at least an acceptable argumentation line starting with $\langle \mathcal{A}_1, Q, \beta \rangle$ that has an even number of arguments.*

Note that we will generalize the use of the term "warranted" for applying it to both goals and arguments: whenever a goal Q is warranted on the basis of a given argument $\langle \mathcal{A}, Q, \alpha \rangle$ as specified in Def. 12, we will also say that the argument $\langle \mathcal{A}, Q, \alpha \rangle$ is warranted. Continuing with Examples 7 and 9, we will next show how to determine, according to the above definition, whether some arguments appearing there (arguments A_4, A_1 and A_2) are warranted.

Example 13 *Let us recall the following arguments:*

$A_1 = \langle \mathcal{A}_1, temp(around\_31), 0.45 \rangle$,
$A_2 = \langle \mathcal{A}_2, temp(between\_25\_30), 0.7 \rangle$,
$A_4 = \langle \mathcal{A}_3, temp(about\_25), 0.9 \rangle$,
$A_5 = \langle \mathcal{A}_3, \sim temp(around\_31), 0.9 \rangle$.

Consider first the argument A_4. It has neither a proper defeater nor a blocking argument, hence there exists an acceptable argumentation line starting with A_4 with just one argument. Indeed, the only possible argumentation line rooted in A_4 that can be obtained is $[A_4]$. Since this line has odd length, according to Definition 12 the goal "$temp(about\_25)$" can be warranted wrt $\mathcal{P}'$ in the context $\mathcal{I}_{U,m}$ with a necessity of 0.9.

Consider now the case of argument A_1. In this case, the argument A_5 is a non-redundant proper defeater for A_1 and A_5 has no defeater, since "$temp(about\_25)$" is a warranted goal with a necessity of 0.9. *Similarly, the argument A_2 is a non-redundant blocking argument for A_1, but A_2 has a proper defeater, namely A_4. However, the line $[A_1, A_2, A_4]$ is not allowed because A_1 and A_4 are contradictory since $m(around\_31) \cap m(about\_25)$ is not normalized. Therefore two acceptable argumentation lines rooted at A_1 can be built: $[A_1, A_5]$ and $[A_1, A_2]$. Since it is not the case that every argumentation line rooted in A_1 has odd length, the argument A_1 cannot be warranted.*

Finally, following a similar discussion for A_2, we can conclude that the argument A_2 is not warranted either.

It must be noted that to decide whether a given goal Q is warranted (on the basis of a given argument A_0 for Q) it may be not necessary to compute *every* possible argumentation line rooted in A_0, e.g. in the case of A_1 in the previous example, it sufficed to detect just one even-length argumentation line to determine that is not warranted. Some aspects concerning computing warrant efficiently by means of a top-down procedure in P-DeLP can be found in (Chesñevar, Simari, & Godo 2005).

Related work

To the best of our knowledge, in the literature there have been not many approaches that aim at combining argumentation and fuzziness, except for the work of Schroeder & Schweimeier (Schweimeier & Schroeder 2001; Schroeder & Schweimeier 2002; Schweimeier & Schroeder 2004). The argumentation framework is also defined for a logic programming framework based on extended logic programming with well-founded semantics, and providing a declarative bottom-up fixpoint semantics along with an equivalent top-down proof procedure. In contrast with our approach, this argumentation framework defines fuzzy unification on the basis of the notion of edit distance, based on string comparison (Schweimeier & Schroeder 2004). Their proposal, on the other hand, does not include an explicit treatment of possibilistic uncertainty as in our case.

There has been generic approaches connecting defeasible reasoning and possibilistic logic (e.g.(Benferhat, Dubois, & Prade 2002)). Including possibilistic logic as part of an argumentation framework for modelling preference handling and information merging has recently been treated by Amgoud & Kaci (Amgoud & Kaci 2005) and Amgoud & Cayrol (Amgoud & Cayrol 2002). Such formulations are based on using a possibilistic logic framework to handle merging of prioritized information, obtaining an aggregated knowledge base. Arguments are then analyzed on the basis of the resulting aggregated knowledge base. An important difference of these proposals with our formulation is that our framework introduces explicit representation of fuzziness along with handling possibilistic logic. Besides, in the proposed framework we attach necessity degrees to object level formulas, propagating such necessity degrees according to suitable inference rules, which differs from the approach used in the proposals above mentioned.

Besides of considering possibilistic logic and fuzziness, a number of hybrid approaches connecting argumentation and uncertainty have been developed, such as Probabilistic Argumentation Systems (Haenni, Kohlas, & Lehmann 2000; Haenni & Lehmann 2003), which use probabilities to compute degrees of support and plausibility of goals, related to Dempster-Shafer belief and plausibility functions. However

[5]It must be remarked that the definition of dialectical tree as well as the characterization of constraints to avoid fallacies in argumentation lines can be traced back to (Simari, Chesñevar, & García 1994). Similar formalizations were also used in other argumentation frameworks (e.g. (Prakken & Sartor 1997)).

this approach is not based on a dialectical theory (with arguments, defeaters, etc.) nor includes fuzziness as presented in this paper.

Conclusions and future work

PGL^+ constitutes a powerful formalism that can be integrated into an argument-based framework like P-DeLP, allowing to combine uncertainty expressed in possibilistic logic and fuzziness characterized in terms of fuzzy constants and fuzzy propositional variables.

In this paper we have focused on characterizing $DePGL^+$, a formal language that combines features from PGL^+ along with elements which are present in most argumentative frameworks (like the notions of argument, counterargument, and defeat). As stated in Sections 5 and 6, part of our current work is focused on providing a formal characterization of warrant in the context of the proposed framework. In particular, we are interested in studying formal properties for warrant that should hold in the context of argumentation frameworks, as proposed in (Caminada & Amgoud 2005). In this paper, Caminada & Amgoud identify anomalies in several argumentation formalisms and provide an interesting solution in terms of rationality postulates which –the authors claim– should hold in any well-defined argumentative system. In (Chesñevar *et al.* 2005) we started a preliminary analysis for this problem in the context of P-DeLP (Chesñevar *et al.* 2004), and currently part of our research is focused on this issue. We are also analyzing how to characterize an alternative conceptualization of warrant in which different warrant degrees can be attached to formulas on the basis of necessity degrees, extending some concepts suggested in (Pollock 2001). Research in these directions is currently being pursued.

Acknowledgments

We thank anonymous reviewers for their comments and suggestions to improve the final version of this paper. This work was supported by Spanish Projects TIC2003-00950, TIN2004-07933-C03-01/03, by Ramón y Cajal Program (MCyT, Spain), by CONICET (Argentina), by the Secretaría General de Ciencia y Tecnología de la Universidad Nacional del Sur and by Agencia Nacional de Promoción Científica y Tecnológica (PICT 2002 No. 13096).

References

Alsinet, T., and Godo, L. 2000. A complete calculus for possibilistic logic programming with fuzzy propositional variables. In *Proc. of UAI-2000 Conf.*, 1–10.

Alsinet, T., and Godo, L. 2001. A proof procedure for possibilistic logic programming with fuzzy constants. In *Proc. of the ECSQARU-2001 Conf.*, 760–771.

Alsinet, T. 2003. *Logic Programming with Fuzzy Unification and Imprecise Constants: Possibilistic Semantics and Automated Deduction*. Number 15. IIIA-CSIC. Bellaterra, Spain.

Amgoud, L., and Cayrol, C. 2002. Inferring from inconsistency in preference-based argumentation frameworks. *J. Autom. Reasoning* 29(2):125–169.

Amgoud, L., and Kaci, S. 2005. An argumentation framework for merging conflicting knowledge bases: The prioritized case. In *Proc. of the ECSQARU-2005 Conf.*, LNAI 3571, 527–538.

Benferhat, S.; Dubois, D.; and Prade, H. 2002. The possibilistic handling of irrelevance in exception-tolerant reasoning. *Annals of Math. and AI* 35:29–61.

Caminada, M., and Amgoud, L. 2005. An axiomatic account of formal argumentation. In *Proc. of the AAAI-2005 Conf.*, 608–613.

Chesñevar, C. I.; Simari, G.; Alsinet, T.; and Godo, L. 2004. A Logic Programming Framework for Possibilistic Argumentation with Vague Knowledge. In *Proc. of the UAI-2004 Conf.*, 76–84.

Chesñevar, C.; Simari, G.; Godo, L.; and Alsinet, T. 2005. On warranted inference in possibilistic defeasible logic programming. In *Proc. of CCIA-2005*. IOS Press, 265–272.

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

Chesñevar, C.; Simari, G.; and Godo, L. 2005. Computing dialectical trees efficiently in possibilistic defeasible logic programming. In *Proc. of LPNMR-2005 Conf.*, 158–171.

Dubois, D.; Lang, J.; and Prade, H. 1994. Possibilistic logic. In D.Gabbay et al. eds., *Handbook of Logic in Art. Int. and Logic Prog. (Nonmonotonic Reasoning and Uncertain Reasoning)*. Oxford Univ. Press. 439–513.

Haenni, R., and Lehmann, N. 2003. Probabilistic Argumentation Systems: a New Perspective on Dempster-Shafer Theory. *Int. J. of Intelligent Systems* 1(18):93–106.

Haenni, R.; Kohlas, J.; and Lehmann, N. 2000. Probabilistic argumentation systems. *Handbook of Defeasible Reasoning and Uncertainty Management Systems*.

Pollock, J. L. 2001. Defeasible reasoning with variable degrees of justification. *Artif. Intell.* 133(1-2):233–282.

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

Schroeder, M., and Schweimeier, R. 2002. Fuzzy argumentation for negotiating agents. In *Proc. of the AAMAS-2002 Conf.*, 942–943.

Schweimeier, R., and Schroeder, M. 2001. Fuzzy argumentation and extended logic programming. In *Proceedings of ECSQARU Workshop Adventures in Argumentation (Toulouse, France)*.

Schweimeier, R., and Schroeder, M. 2004. Fuzzy unification and argumentation for well-founded semantics. In *proc. of SOFSEM 2004*, LNCS 2932, 102–121.

Simari, G.; Chesñevar, C.; and García, A. 1994. The role of dialectics in defeasible argumentation. In *Proc. of the XIV Intl. Conf. of the Chilean Society for Computer Science*, 260–281. Universidad de Concepción, Concepción (Chile).