(Preliminary manuscript)

Formalizing Argumentative Reasoning in a Possibilistic Logic Programming Setting with Fuzzy Unification [⋆]

Teresa Alsinet [a] Carlos I. Chesñevar [b,d] Lluís Godo [c]
Sandra Sandri [c] Guillermo Simari [d]

[a] *Departament of Computer Science, Universitat de Lleida*
C/Jaume II, 69 – 25001 Lleida, SPAIN – *Email: tracy@diei.udl.es*

[b] *CONICET (National Council of Scientific and Technical Research),* ARGENTINA

[c] *Artificial Intelligence Research Institute (IIIA), CSIC*
Campus UAB, Bellaterra, SPAIN – *Email: {godo, sandri}@iiia.csic.es*

[d] *Departament of Computer Science & Engineering, Universidad Nacional del Sur*
Av. Alem 1253 - 8000 Bahía Blanca, ARGENTINA
Email: {cic,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 the object-language level. In spite of its expressive power, an important limitation in P-DeLP is that imprecise, fuzzy information cannot be expressed in the object language. One interesting alternative for solving this limitation is the use of PGL$^+$, a possibilistic logic over Gödel logic extended with fuzzy constants. Fuzzy constants in PGL$^+$ allow expressing disjunctive information about the unknown value of a variable, in the sense of a magnitude, modeled as a (unary) predicate. The aim of this article is twofold: firstly, we formalize DePGL$^+$, a possibilistic defeasible logic programming language that extends P-DeLP through the use of PGL$^+$ in order to incorporate fuzzy constants and a fuzzy unification mechanism for them. Secondly, we propose a way to handle conflicting arguments in the context of the extended framework.

Key words: Possibilistic logic, fuzzy constants, fuzzy unification, defeasible argumentation, warrant computation

[⋆] This article extends preliminary results presented in the workshop paper *"On the Computation of Warranted Arguments within a Possibilistic Logic Framework with*

1 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 [12]. Recent developments have been oriented towards integrating argumentation as part of logic programming languages. In this context, Possibilistic Defeasible Logic Programming (P-DeLP) [16] 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 [16]) is that the explicit treatment of imprecise, fuzzy information was not actually performed. Such a possibility is indeed very important to properly represent qualitative, symbolic information about continuous numerical magnitudes. To remedy this problem, in this paper we propose the use of PGL$^+$, a possibilistic logic over Gödel fuzzy logic extended with *fuzzy constants*. Fuzzy constants in PGL$^+$ provide a suitable means for expressing such a symbolic / numerical interface between (finite) scales of labels and continuous scales of magnitudes represented by (unary) predicates. Indeed, a fuzzy constant is mapped, under a given PGL$^+$ interpretation, to a fuzzy subset of a (possibly continuous) domain of elements, in contradistinction to single elements in the case of usual object constants in predicate logics. 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 object constant, which will be mapped to a fuzzy set of the (numerical) domain of the variable John's salary. Notice that this kind of statements expresses *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

Fuzzy Unification", Proc. of the 11th Intl. Workshop on Non-Monotonic Reasoning (NMR 2006), pp. 227-236 (J.Dix and A.Hunter, eds.), Windermere, UK, June 2006.

PGL$^+$, instead of (classical) possibilistic logic, in order to incorporate fuzzy constants and fuzzy unification, and second to propose a way to handle conflicting arguments in the context of the extended framework. The rest of the article is structured as follows. First, in Section 2 we present an overview of PGL$^+$ and discuss the fundamentals of defeasible argumentation. Then in Section 3 we define the DePGL$^+$ programming language. Sections 4 and 5 focus on the characterization of arguments in DePGL$^+$ and the analysis of the notion of conflict among arguments in the context of our proposal. In Section 6 we discuss some problematic situations that may arise when trying to define the notion of warranted arguments in DePGL$^+$, and propose some solutions. Finally in Sections 7 and 8 we discuss some related work and present the main conclusions we have obtained.

2 Possibilistic logic and argumentation: an overview

In order to make this article self-contained, this Section discusses the fundamentals of possibilistic logic and defeasible argumentation, with special emphasis on PGL$^+$ and P-DeLP.

2.1 Possibilistic logic and PGL$^+$

Possibilistic logic [17] 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 $\pi : \Omega \to [0, 1]$ 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 $\pi(w) = 0$ then w is considered as totally impossible. A possibilistic formula (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 [1,2] 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 $\mathcal{S}$ 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 $\mu_{m(A)} : U_\sigma \rightarrow [0, 1]$. 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 $\mu_{m(A)}(u_0) = 1$.

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 modeled by a normalized possibility distribution on $\mathcal{I}_{U,m}$, $\pi : \mathcal{I}_{U,m} \to [0,1]$ which provides a ranking of interpretations according to their possibility degree. 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. Due to different technical reasons (see e.g. [4,3]), the necessity measure adopted for PGL$^+$ is defined as follows:

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

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. [17]). According to this semantics, given a context $\mathcal{I}_{U,m}$ a formula like

$$(\mathit{age\_Peter}(\mathit{about\_35}), 0.9)$$

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

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

where $[\mathit{about\_35}]_\beta$ denotes the β-cut of the fuzzy set $m(\mathit{about\_35})$.

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 $\pi : \mathcal{I}_{U,m} \to [0,1]$ that satisfies all the clauses in P. Satisfiable clauses enjoy the following result [4]: 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, always in a given 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 [4], it is proved that $\|\varphi\|_P = \inf\{N([\varphi] \mid \pi) \mid \pi \models P\}$.

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

Generalized resolution:

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

Fusion:

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

Intersection:

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

Resolving uncertainty:

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

Semantic unification:

$$\frac{(p(A), \alpha)}{(p(B), \min(\alpha, \beta))}\ [\mathrm{SU}],\ \text{ where } \beta = N(m(B) \mid m(A))$$

In the description of the GR and FU rules, we have used s and t to denote an arbitrary conjunction of literals, possibly empty. We have also used above several notation conventions regarding fuzzy constants. Namely, $A \cup B$ denotes a fuzzy constant such that $m(A \cup B) = \max(m(A), m(B))$, $A \cap B$ denotes a fuzzy constant such that $m(A \cap B) = \min(m(A), m(B))$, where min and max are applied point-wisely (also the max in the UN inference rule), and the necessity measure $N(m(B) \mid m(A))$ is defined as above, i.e. $N(m(B) \mid m(A)) = \inf_{u \in U_\sigma} \mu_{m(A)}(u) \Rightarrow \mu_{m(B)}(u)$, where A and B are fuzzy constants of sort σ and $\Rightarrow$ is the reciprocal of Gödel implication function. Remarkable properties of this measure are $N(m(A) \mid m(A)) = 1$ and $N(m(B) \mid \max(1 - \alpha, m(B)) = \min(\alpha, N(m(A) \mid m(B))$. In the rest of the paper we will also write all these expressions without the explicit reference to the context mapping m when no confusion is possible.

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

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

The notion of *proof* in PGL^+, denoted by $\vdash$, is that of deduction by means of the triviality axiom, $(\varphi, 0)$, 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 $P \vdash (\varphi, \alpha)$. In [2,4] it is shown that 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, that satisfy two further constraints, called *modularity* and *context* constraints. Actually, the modularity constraint can be achieved by a pre-processing of the program which extends the original PGL^+ program with valid clauses by means of the GR and FU inference rules. For instance, if the original program contains clauses like $(p(A) \rightarrow q, \alpha)$ and $(p(B) \rightarrow q, \beta)$, then the new clause $(p(A \cup B) \rightarrow q, \min(\alpha, \beta))$ would be added in this pre-processing step. This is indeed the first step of an efficient and complete proof procedure for PGL^+ programs satisfying what we call *context* constraint. The idea is that in a PGL^+ program satisfying the context constraint, the use of the SU and MP inference rules is enough to attain a degree of deduction equal to the degree of possibilistic entailment. Then, the second step of the proof procedure is based on the MP, SU, UN and IN rules and translates a PGL^+ program satisfying the modularity constraint into a semantically equivalent set of 1-weighted facts, whenever the program satisfied the context constraint. The final step is a deduction step, based on the SU rule, which computes the maximum degree of possibilistic entailment of a goal from the equivalent set of 1-weighted facts.

2.2 Defeasible Argumentation and P-DeLP

Defeasible argumentation [12,28] has evolved in the last decade as a successful approach to formalize commonsense reasoning. When a rule supporting a conclusion may be defeated by new information, it is said that such reasoning is *defeasible* [24,25]. When defeasible reasons or rules are chained to reach a conclusion, we have *arguments* instead of proofs. Arguments may compete, rebutting each other, so a *process* of argumentation is a natural result of the search for arguments. Adjudication of competing arguments must be performed, comparing arguments in order to determine what beliefs are ultimately accepted as *warranted* or *justified*. Preference among conflicting arguments is defined in terms of a *preference criterion* which establishes a partial order “ $\preceq$ ” among possible arguments; thus, for two arguments A and B in conflict, it may be the case that A is strictly preferred over B ($A \succ B$), that A and B are equally preferable ($A \succeq B$ and $A \preceq B$) or that A and B are not comparable with each other. Arguments may be defeated by other arguments, which on their turn may be defeated by other arguments, and so on. This prompts a recursive analysis, which is usually modelled by means of a tree structure called *dialectical tree* or *argument tree*. When an argument is ultimately accepted after considering all possible defeaters, the argument is said to be *warranted* or *justified*.

In the last few years the argumentation community has given particular attention to several extensions of *logic programming* which have turned out to be computationally manageable for formalizing knowledge representation and argumentative inference. Several approaches have been developed, some of them based on normal logic programming [22], extended logic programming [27], and defeasible logic programming or DeLP [19], among others. The DeLP approach has been particularly attractive in the context of real-world applications, such as recommender systems [15], knowledge management [11] and natural language processing [13]. *Possibilistic Defeasible Logic Programming* (P-DeLP) [1] is a logic programming framework based on DeLP, and hence combining features from argumentation theory and logic programming which incorporates a treatment of possibilistic uncertainty at the object-language level.

The language $\mathcal{L}$ of P-DeLP is inherited from the language of logic programming, including the usual notions of atom, literal, rule and fact. In particular, the symbol $\sim$ stands for (strong) *negation*. A *literal* $L \in \mathcal{L}$ is a ground (fuzzy) atom q or a negated ground (fuzzy) atom $\sim q$, where q is a ground (fuzzy) propositional variable. A *goal* in P-DeLP is any literal $L \in \mathcal{L}$. A *program* $\mathcal{P}$ in P-DeLP is a set of *weighted* clauses, where every weighted clause is a pair of the form (φ, α), where φ is a rule $p \leftarrow q_1, q_2, \ldots, q_k$ or fact $p \leftarrow$ (i.e., a rule with empty antecedent), where $p, q_1, q_2, \ldots, q_k$ are literals, and $\alpha \in [0, 1]$ expresses a lower bound for the necessity degree of φ. The subset $\Pi_{\mathcal{P}}$ of weighted clauses in $\mathcal{P}$ whose necessity degree is 1 corresponds to certain clauses, and is assumed to be *non-contradictory*. A set of clauses Γ will be deemed as *contradictory*, denoted $\Gamma \vdash \bot$, if $\Gamma \vdash (q, \alpha)$ and $\Gamma \vdash (\sim q, \beta)$, with $\alpha > 0$ and $\beta > 0$, for some atom q in $\mathcal{L}$.

As in most argument-based logic programming frameworks, in P-DeLP solving a goal Q accounts for finding an *argument* supporting Q which is ultimately accepted or *warranted*. Given a P-DeLP program $\mathcal{P}$, the notion of an argument $\mathcal{A}$ supporting a literal Q with a necessity degree α (denoted $\langle \mathcal{A}, Q, \alpha \rangle$) is based inferring (Q, α) from $\mathcal{P}$ using Generalized Modus Ponens as a possibilistic resolution rule. The set $\mathcal{A}$ accounts for the set of weighted clauses from $\mathcal{P}$ with necessity degree < 1 used to derive (Q, α).

The set of uncertain clauses in a given P-DeLP program $\mathcal{P}$ account for tentative and incomplete information. Hence *conflicting arguments* may arise. An argument $\langle \mathcal{A}, Q, \alpha \rangle$ may be *defeated* by another argument $\langle \mathcal{B}, R, \beta \rangle$. The notion of defeat in P-DeLP is associated with determining a sub-argument (sub-proof) $\langle \mathcal{A}', Q, \alpha' \rangle$ in the attacked argument $\langle \mathcal{A}, Q, \alpha \rangle$ such that $\Pi_{\mathcal{P}} \cup$

[1] For a full description of P-DeLP the reader is referred to [16].

$\{(Q', \alpha'), (R, \beta)\}$ is contradictory and $\beta \geq \alpha'$. In this case, the argument $\langle \mathcal{B}, R, \beta \rangle$ is called a defeater for $\langle \mathcal{A}, Q, \alpha \rangle$ As defeaters are arguments, they may be in turn defeated by other arguments. This prompts a recursive analysis, associated with solving a goal Q in P-DeLP.

Given a P-DeLP program $\mathcal{P}$, solving a goal Q_0 accounts for first finding an argument $\langle \mathcal{A}_0, Q_0, \alpha_0 \rangle$ supporting (Q_0, α_0), and then performing an exhaustive analysis of possible defeaters for $\langle \mathcal{A}_0, Q_0, \alpha_0 \rangle$, defeaters for such defeaters, and so on. Every one of such sequences $\lambda = [\ \langle \mathcal{A}_0, Q_0, \alpha_0 \rangle, \langle \mathcal{A}_1, Q_1, \alpha_1 \rangle, \ldots, \langle \mathcal{A}_n, Q_n, \alpha_n \rangle, \ldots]$ is called an *argumentation line*, standing for a *dialogue* between two parties (a proponent who advances the even-level arguments, starting with the original argument at issue, and an opponent who attacks the proponent's arguments, by advancing odd-level arguments). If all possible argumentation lines rooted in $\langle \mathcal{A}_0, Q_0, \alpha_0 \rangle$ are of odd length, this implies that every possible dialogue on the basis of the program $\mathcal{P}$ was won by the proponent, and hence the original argument $\langle \mathcal{A}_0, Q_0, \alpha_0 \rangle$ is *warranted.*

3 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 [16] will be extended with fuzzy constants and fuzzy propositional variables, while arguments will keep 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* is interpreted by a crisp interval of rpm's, say $[0, 2000]$, then $speed(low)$ will be true in I iff such element $i(speed)$ 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 "$not[\exists x \in low$ 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. One usually takes $n(x) = 1 - x$, but any other is also allowed. Indeed, 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) $\neg A$.

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 $\tau(\Gamma) \vdash \tau((\varphi, \alpha))$, where the elements in Γ and (φ, α) are DePGL$^+$ clauses.

4 Arguments in DePGL$^+$

In Section 2.1 we have formalized the many-valued and the possibilistic semantics of PGL$^+$, 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 $|p(C)|_{\Gamma_2} > 0$, for some fuzzy constants B and C such that $A = B \cap C$.

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 the transformation τ) is required to satisfy the modularity constraint [2,4]. This is formally stated as follows.

Definition 2 (DePGL$^+$ program) *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) program clauses are considered. Indeed, since fuzzy constants are interpreted as (flexible) restrictions on an existential quantifier, atomic formulas clearly express disjunctive information. For instance, within a context $\mathcal{I}_{U,m}$, when $m(A) = \{a_1, \ldots, a_n\}$, $p(A)$ is semantically 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 clauses we also have to consider all those new clauses that can be obtained through a completion process of the program which is based on the GR and FU inference rules.

Example 3 *(Adapted from [16]) 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.*

(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)	$(temp(interval\_25\_31), 1)$
(17)	$(temp(around\_31), 0.85)$

Fig. 1. DePGL$^+$ program $\mathcal{P}_{eng}$ (example 3)

- *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,*
- *the temperature is in the interval* $[25, 31]^o C$*, and*
- *the temperature seems to be around* $31^o C$*.*

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 while imprecise knowledge is represented by means of fuzzy object constants like high, low,* $around\_31$ *and* $interval\_25\_31$*.*

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) .

Definition 4 (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) $\Pi \cup \mathcal{A}$ *is non contradictory;*
(2) $\alpha = \sup\{\beta \in [0,1] \mid \Pi \cup \mathcal{A} \vdash_\tau (Q, \beta)\}$*, i.e.* α *is the greatest degree of deduction of* Q *from* $\tau(\Pi \cup \mathcal{A})$*, denoted as* $|Q|_{\tau(\Pi \cup \mathcal{A})}$*; and*
(3) $\mathcal{A}$ *is minimal wrt set inclusion, i.e. there is no* $\mathcal{A}_1 \subset \mathcal{A}$ *such that* $\Pi \cup \mathcal{A}_1 \vdash_\tau$

(Q, α).

Definition 5 (Subargument) *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 $\mathcal{A}$.*

Note that for the program $\mathcal{P}_{eng}$ in Example 3 the sets of uncertain clauses $\mathcal{S} = \{(pump\_fuel \leftarrow sw1, 0.6)\}$ and $\mathcal{A} = \{(pump\_fuel \leftarrow sw1, 0.6), (fuel\_ok \leftarrow pump\_fuel, 0.6)\}$ are arguments for the goals $pump\_fuel$ and $fuel\_ok$, respectively, with necessity degree 0.6 and $\langle \mathcal{S}, pump\_fuel, 0.6 \rangle$ is a subargument of $\langle \mathcal{A}, fuel\_ok, 0.6 \rangle$.

Let $\mathcal{I}_{U,m}$ be a context, let P be a DePGL$^+$ program and let p be a predicate symbol of type (σ) appearing in P. Then, in [2,4] it is shown that

$$|p(A)|_{\tau(P)} = |p(A)|_{(p(C),1)} = N(m(A) \mid m(C)),$$

where C is the object constant of sort σ such that, for each $u \in U_\sigma$, $\mu_{m(C)}(u) = \inf\{\mu_{m(B)}(u) \mid B$ object constant of sort σ such that $P \vdash_\tau (p(B), 1)\}$. Thus, the greatest degree of deduction of $p(A)$ from $\tau(P)$ corresponds with the unification degree between the fuzzy constant A and the most specific[2] fuzzy constant that can be deduced from P with necessity degree 1. Then, in order to compute arguments with the greatest degree of deduction, we need to introduce the notion of canonical argument.

Definition 6 (Canonical argument) *Let $\mathcal{P}$ be a DePGL$^+$ program and let $\mathcal{I}_{U,m}$ be a context. An argument wrt $\mathcal{P}$ and $\mathcal{I}_{U,m}$ $\langle \mathcal{A}, Q, \alpha \rangle$ is called* canonical *if either Q is a propositional variable, or if $Q = p(A)$ then $\alpha = 1$ and there is no fuzzy constant C more specific than A such that $\Pi \cup \mathcal{A} \vdash (p(C), 1)$.*

It must be noted that given an argument of the form $\langle \mathcal{A}, p(A), \alpha \rangle$, the canonical argument associated with the set $\mathcal{A}$ and predicate p is unique. As we will see later in Sections 5 and 6, the notion of canonical argument will turn to be very useful since it will allow us to restrict the search for conflicting arguments and simplify the process of deciding when an argument is ultimately acceptable or not. In [2,4] an efficient algorithm has been presented for computing the most specific fuzzy constant that can be deduced, for a given predicate symbol, from a set of clauses with necessity degree 1 which is based on the MP, SU, IN and UN inference rules. Consequently, this algorithm can be then used to compute canonical arguments.

The next procedure addresses the important issue of how to build arguments

[2] That is, the smallest, as membership function, with respect to the point-wise order.

for a DePGL$^+$ program.

Algorithm 7 (Argument construction procedure) *Given a context $\mathcal{I}_{U,m}$, 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 $\alpha > 0$ wrt $\mathcal{P}$ and $\mathcal{I}_{U,m}$ iff $\mathcal{A}$ and α can be computed by (recursively) applying any of the following construction rules:*

(1) Building arguments from facts (INTF):
- **If** $(Q, 1) \in \Pi$
  then $\mathcal{A} = \emptyset$ *and* $\alpha = 1$
- **If** $(Q, \beta) \in \Delta$ *and* $\Pi \cup \{(Q, \beta)\} \not\vdash_\tau \bot$ *and* $\Pi \not\vdash_\tau (Q, \gamma)$ *for any* $\gamma > \beta$
  then $\mathcal{A} = \{(Q, \alpha)\}$ *and* $\alpha = \beta$

(2) Building arguments from program rules by applying the modus ponens rule (MPA):
- **If** $(Q \leftarrow L_1 \wedge \ldots \wedge L_k, 1) \in \Pi$ *and* $\langle \mathcal{A}_1, L_1, \beta_1 \rangle, \ldots, \langle \mathcal{A}_k, L_k, \beta_k \rangle$ *are arguments and* $\Pi \cup \bigcup_{i=1}^k \mathcal{A}_i \not\vdash_\tau \bot$ *and there is no* $\mathcal{B} \subset \bigcup_{i=1}^k \mathcal{A}_i$ *such that* $\Pi \cup \mathcal{B} \vdash_\tau (Q, \gamma)$ *with* $\gamma \geq \min(\beta_1, \beta_2, \ldots, \beta_k)$
  then $\mathcal{A} = \bigcup_{i=1}^k \mathcal{A}_i$ *and* $\alpha = \min(\beta_1, \beta_2, \ldots, \beta_k)$
- **If** $(Q \leftarrow L_1 \wedge L_2 \wedge \ldots \wedge L_k, \beta) \in \Delta$ *and* $\langle \mathcal{A}_1, L_1, \beta_1 \rangle, \ldots, \langle \mathcal{A}_k, L_k, \beta_k \rangle$ *are arguments and* $\Pi \cup \{(Q \leftarrow L_1 \wedge L_2 \wedge \ldots \wedge L_k, \beta)\} \cup \bigcup_{i=1}^k \mathcal{A}_i \not\vdash_\tau \bot$ *and there is no* $\mathcal{B} \subset \bigcup_{i=1}^k \mathcal{A}_i \cup (Q \leftarrow L_1 \wedge L_2 \wedge \ldots \wedge L_k, \beta)$ *such that* $\Pi \cup \mathcal{B} \vdash_\tau (Q, \gamma)$ *with* $\gamma \geq \min(\beta, \beta_1, \beta_2, \ldots, \beta_k)$
  then $\mathcal{A} = \bigcup_{i=1}^k \mathcal{A}_i \cup (Q \leftarrow L_1 \wedge L_2 \wedge \ldots \wedge L_k, \beta)$ *and* $\alpha = \min(\beta, \beta_1, \beta_2, \ldots, \beta_k)$

(3) Building arguments from canonical arguments by applying the unification rule (SUA):
If $Q = p(B)$ *and* $\langle \mathcal{A}_1, p(A), 1 \rangle$ *is a canonical argument such that* $N(m(B) \mid m(A)) \neq 0$ *and there is no* $\mathcal{A}_2 \subset \mathcal{A}_1$ *such that* $\Pi \cup \mathcal{A}_2 \vdash_\tau (p(B), \gamma)$ *with* $\gamma \geq N(m(B) \mid m(A))$
then $\mathcal{A} = \mathcal{A}_1$ *and* $\alpha = N(m(B) \mid m(A))$

(4) Building arguments from canonical arguments by applying the intersection rule (INA):
If $Q = p(C)$ *and* $\langle \mathcal{A}_1, p(A), 1 \rangle$ *and* $\langle \mathcal{A}_2, p(B), 1 \rangle$ *are a pair of canonical arguments such that* $m(C) = m(A) \cap m(B)$ *and* $\Pi \cup \mathcal{A}_1 \cup \mathcal{A}_2 \not\vdash_\tau \bot$ *and there is no* $\mathcal{B} \subset \mathcal{A}_1 \cup \mathcal{A}_2$ *such that* $\Pi \cup \mathcal{B} \vdash_\tau (p(C), 1)$
then $\mathcal{A} = \mathcal{A}_1 \cup \mathcal{A}_2$ *and* $\alpha = 1$

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 and to ensure that $\alpha = |Q|_{\tau(\Pi \cup \mathcal{A})}$. On the one hand, appropriate preconditions ensure that the proof obtained satisfies the non-contradiction constraint of arguments wrt the certain knowledge Π of the program and that computed arguments are minimal wrt set inclusion. On the other hand, the completeness results of the PGL$^+$ proof method (see [4]) ensure that necessity degrees computed by means of the MP, SU and IN inference rules after resolving uncertainty on both program facts and new derived facts, correspond with greatest degrees of deduction. 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 $\mathcal{P}$ and that Q can not be proved from Π with a necessity degree greater or equal than α. 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 $(Q \leftarrow L_1 \wedge \ldots \wedge L_k, \beta)$ when $\beta < 1$, and whenever it is strictly necessary to consider all these clauses in order to prove Q with a greater necessity degree. Rule SUA accounts for semantic unification from canonical arguments; i.e. corresponds to the unification between the fuzzy constant B and the more specific fuzzy constant that can be deduced from $\mathcal{A}_1$ with necessity degree 1. As the rule does not deal with new uncertain knowledge, we do not need to check the non-contradictory constraint. However, it is necessary to ensure that all defeasible information is strictly necessary to derive the goal. In a similar way, rule INA applies intersection between canonical arguments provided that the resulting intersection is non contradictory wrt Π and minimal wrt set inclusion.

Note that we cannot ensure that arguments with necessity degree 1 are canonical arguments. The following proposition establishes the relationship between arguments and canonical arguments.

Proposition 8 *Let $\mathcal{I}_{U,m}$ be a context, let $\mathcal{P} = (\Pi, \Delta)$ be a DePGL$^+$ program and let $\mathcal{A} \subseteq \Delta$ be a set of uncertain clauses. If $\langle \mathcal{A}, p(A), \alpha \rangle$ is an argument then there exists one, and only one, fuzzy constant C such that $\langle \mathcal{A}, p(C), 1 \rangle$ is a canonical argument.*

Proof: On the one hand, if $\langle \mathcal{A}, p(A), \alpha \rangle$ is an argument then $\alpha = \sup\{\beta \in [0,1] \mid \Pi \cup \mathcal{A} \vdash_\tau (p(A), \beta)$. Therefore, as we proved in [4], $\Pi \cup \mathcal{A} \vdash_\tau (p(A), \alpha)$ and, by the soundness of the UN inference rule, $\Pi \cup \mathcal{A} \vdash_\tau (p(B), 1)$ where B is a fuzzy constant such that $m(B) = \max(1 - \alpha, m(A))$. Hence, we can ensure that $\mathcal{B} = \{B \text{ object constant} \mid \Pi \cup \mathcal{A} \vdash_\tau (p(B), 1)\}$ is a non-empty set, and thus, we can safely define C as the most specific fuzzy constant that can be deduced from $\Pi \cup \mathcal{A}$ with necessity degree 1.
On the other hand, if $\langle \mathcal{A}, p(A), \alpha \rangle$ is an argument then $\mathcal{A}$ is minimal wrt set inclusion, and thus, for all $\mathcal{A}_1 \subset \mathcal{A}$, $|p(A)|_{\tau(\Pi \cup \mathcal{A}_1)} < \alpha$ and, by the completeness of PGL$^+$, $\|p(A)\|_{\tau(\Pi \cup \mathcal{A}_1)} < \alpha$. Therefore, by the PGL$^+$ semantics, $\|p(B)\|_{\tau(\Pi \cup \mathcal{A}_1)} < 1$ where B is a fuzzy constant such that $m(B) = \max(1 - \alpha, m(A))$. Hence, as C is either B or is more specific than B, $\|p(C)\|_{\tau(\Pi \cup \mathcal{A}_1)} <$

1, and thus, $|p(C)|_{\tau(\Pi \cup \mathcal{A}_1)} < 1$ for all $\mathcal{A}_1 \subset \mathcal{A}$. $\square$

Example 9 *Consider the program $\mathcal{P}_{eng}$ in Example 3, where $temp(\cdot)$ is a unary predicate of type* (`degrees`)*, $speed(\cdot)$ is a unary predicate of type* (`rpm`)*, "high", "interval_25_31" and "around_31" are object constants of sort* `degrees`*, and* low *is an object constant of sort* `rpm`*. Further, consider the context $\mathcal{I}_{U,m}$ such that:*

$U = \{U_{\texttt{degrees}} = [-100, 100]\ ^oC,\ U_{\texttt{rpm}} = [0, 200]\}$;
$m(high) = [28, 30, 100, 100]$[3],
$m(interval\_25\_31) = [25, 25, 31, 31]$,
$m(around\_31) = [26, 31, 31, 36]$,
$m(low) = [10, 15, 25, 30]$, *and*
$m(\neg low) = 1 - m(low)$.

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

(1) The argument $\langle \mathcal{B}_1, fuel\_ok, 0.6\rangle$ can be derived as follows:
  i) $\langle \emptyset, sw1, 1\rangle$ from rule (13) via INTF.
  ii) $\langle \mathcal{D}, pump\_fuel, 0.6\rangle$ from rule (2) and i) via MPA.
  iii) $\langle \mathcal{B}_1, fuel\_ok, 0.6\rangle$ from rule (3) and ii) via MPA.
  where
    $\mathcal{D}=\{(pump\_fuel \leftarrow sw1, 0.6)\}$,
    $\mathcal{B}_1 = \mathcal{D} \cup \{(fuel\_ok \leftarrow pump\_fuel, 0.85)\}$.

(2) Similarly, the argument $\langle \mathcal{C}_1, oil\_ok, 0.8\rangle$ can be derived using the rules (15), (4) and (5) via INTF, 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) $\langle \mathcal{B}_1, fuel\_ok, 0.6\rangle$ 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) and the rule (6) via MPA.
  with $\mathcal{A}_1=\{(engine\_ok \leftarrow fuel\_ok \wedge oil\_ok, 0.6)\} \cup \mathcal{B}_1 \cup \mathcal{C}_1$.
  Note that $\langle \mathcal{C}_1, oil\_ok, 0.8\rangle$ and $\langle \mathcal{B}_1, fuel\_ok, 0.6\rangle$ 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) As $N(m(around\_31) \mid m(interval\_25\_31)) = 0$, it is not possible to derive an argument for $temp(around\_31)$ from the set of certain clauses of program $\mathcal{P}_{eng}$. Then,
  $$\langle \{(temp(around\_31),\ 0.85)\}, temp(around\_31), 0.85\rangle$$
  can be derived from rule (17) via INTF.

[3] 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.

ii) Consider one new fuzzy constant "specific_around_31" interpreted in the context $\mathcal{I}_{U,m}$ *as*

$$m(specific\_around\_31) = \\ \min(m(interval\_25\_31), \max(1 - 0.85, m(around\_31))).$$

The canonical argument for i) is

$$\langle\{(temp(around\_31),\ 0.85)\}, temp(specific\_around\_31), 1\rangle.$$

Now, as it is not possible to derive an argument for $temp(high)$ *from the set of certain clauses of program* $\mathcal{P}_{eng}$ *and* $N(m(high) \mid m(specific\_around\_31)) = 0.8$*, from the canonical argument via SUA we get*

$$\langle\{(temp(around\_31),\ 0.85)\}, temp(high), 0.8\rangle.$$

iii) $\langle\mathcal{C}_2, \sim oil\_ok, 0.8\rangle$ *from ii) and the rule (8) via MPA.*

(5) Similarly, an argument $\langle\mathcal{A}_2, \sim engine\_ok, 0.8\rangle$ *can be derived using the rules (17), (16) and (7) via INTF, SUA, and MPA, with* $\mathcal{A}_2 = \{(temp(around\_31),\ 0.85); (\sim engine\_ok \leftarrow temp(high), 0.95)\}$.

5 Counter-argumentation and defeat in DePGL$^+$

Given a program and a particular context, it can be the case that there exist arguments for contradictory literals. 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 [16] but incorporating the treatment of fuzzy constants.

Definition 10 (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 either*

(i) Q_1 *and* Q *are propositional variables and* $Q_1 = \sim Q$ [4]*,*
(ii) or $Q_1 = p(A)$ *and* $Q = p(B)$ *for some predicate* p *and fuzzy constants* A *and* B*, such that* $m(A) \cap m(B)$ *is a non-normalized fuzzy set.*

Note that our definition of counterargument accounts for the two usual conflict situations in argumentation systems [12,28]: *direct attacks* (also called *rebutters*) , in which conflicting arguments have opposite conclusions, and *indirect*

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

attacks (sometimes referred to as *undercutters* in the literature), in which a given argument is in conflict with some intermediate step or subargument of another argument.

Since arguments rely on uncertain and hence defeasible information, conflicts among arguments may be resolved by comparing their strength. Therefore, a notion of defeat amounts to establish a *preference criterion* on conflicting arguments. In our framework, when no fuzzy constants are involved, it seems natural to define it on the basis of necessity degrees associated with arguments, following [16]. When fuzzy constants are involved, due to the concept of contradiction we have adopted, the comparison of conflictive arguments becomes more involved.

To simplify, assume we have two arguments

$$Arg_1 = \langle X, p(A), \alpha \rangle, \quad Arg_2 = \langle Y, p(B), \beta \rangle$$

such that $A \cap B$ is non-normalized [5], hence Arg_1 counter-argues Arg_2 and viceversa. In order to compare these arguments what we do is to analyze how much each of them supports the negated conclusion of the other. In fact, from Arg_1 we can build an argument for $\sim p(B)$ by applying the SUA inference rule to its corresponding canonical argument $(X, p(A'), 1)$, where $A' = \max(1-\alpha, A)$, which yields the argument

$$Arg_1' = \langle X, \sim p(B), \min(\alpha, N(\neg B \mid A)) \rangle$$

taking into account that, by definition, $\sim p(B) = p(\neg B)$ and that $N(\neg B \mid \max(1-\alpha, A)) = \min(\alpha, N(\neg B \mid A))$. Analogously, from Arg_2 we can build the following argument for $\sim p(A)$:

$$Arg_2' = \langle Y, \sim p(A), \min(\beta, N(\neg A \mid B)) \rangle$$

Therefore, we need to actually compare the strengths of Arg_1' and Arg_2 on the one hand, and of Arg_2' and Arg_1 on the other hand. The following possibilities arise:

(1) if $\min(\alpha, N(\neg B \mid A)) > \beta$, then it follows that $\alpha > \min(\beta, N(\neg A \mid B))$ as well. In this case Arg_1' is stronger than Arg_2 and Arg_1 stronger than Arg_2'. Then it is clear that Arg_1 is strictly stronger than Arg_2. Conversely, if $\min(\beta, N(\neg A \mid B)) > \alpha$, then Arg_2 is strictly stronger than Arg_1;

(2) if $\min(\alpha, N(\neg B \mid A)) = \beta$ and $\min(\beta, N(\neg A \mid B)) = \alpha$, then Arg_1' and Arg_2 are equally strong, as well as as Arg_1 and Arg_2'. In this case, we have that $\min(N(\neg B \mid A), N(\neg A \mid B)) \geq \alpha = \beta$, and we can compare

[5] Note that we write $A \cap B$ for $m(A) \cap m(B)$, and similarly in other expressions which follow, dropping the m function symbol when no confusion arises.

the values of $N(\neg B \mid A))$ and $N(\neg A \mid B))$ to decide whether Arg_1 or Arg_2 finally wins;

(3) if $\min(\alpha, N(\neg B \mid A)) = \beta$ and $\min(\beta, N(\neg A \mid B)) < \alpha$, we have that Arg'_1 is equally strong to Arg_2 but Arg'_2 is weaker than Arg_1. In this case we consider Arg_1 as the winner. Conversely, when $\min(\alpha, N(\neg B \mid A)) < \beta$ and $\min(\beta, N(\neg A \mid B)) = \alpha$, Arg_2 is considered as winner;

(4) finally, if $\min(\alpha, N(\neg B \mid A)) < \beta$ and $\min(\beta, N(\neg A \mid B)) < \alpha$ then there is no winner argument, we have a tie.

According to the above considerations we define the following notions of proper and blocking defeaters.

Definition 11 (Defeat) *Let $\mathcal{P}$ be a DePGL$^+$ program, let $\mathcal{I}_{U,m}$ be a context, and let the argument $\langle \mathcal{A}_1, Q_1, \alpha_1 \rangle$ counterargue the argument $\langle \mathcal{A}_2, Q_2, \alpha_2 \rangle$ with disagreement subargument $\langle \mathcal{A}, Q, \beta \rangle$. We distinguish two cases:*

Case(1): Q_1 and Q are propositional variables
We say that $\langle \mathcal{A}_1, Q_1, \alpha_1 \rangle$ is a proper *(resp.* blocking*)* defeater *for $\langle \mathcal{A}_2, Q_2, \alpha_2 \rangle$ when $\alpha_1 > \beta$ (resp. $\alpha_1 = \beta$).*

Case(2): $Q_1 = p(A)$ and $Q = p(B)$
We say that $\langle \mathcal{A}_1, Q_1, \alpha_1 \rangle$ is a proper defeater *for $\langle \mathcal{A}_2, Q_2, \alpha_2 \rangle$ when either*
- $\min(\alpha_1, N(\neg B \mid A)) > \beta$,
- *$\alpha_1 = \beta$ and $N(\neg B \mid A) > N(\neg A \mid B)$, or*
- *$\min(\alpha_1, N(\neg B \mid A)) = \beta$ and $\min(\beta, N(\neg A \mid B)) < \alpha_1$.*

We say that $\langle \mathcal{A}_1, Q_1, \alpha_1 \rangle$ is a blocking defeater *for $\langle \mathcal{A}_2, Q_2, \alpha_2 \rangle$ when*
- *$\alpha_1 = \beta$ and $N(\neg B \mid A)) = N(\neg A \mid B))$, or*
- *$\min(\alpha_1, N(\neg B \mid A)) < \beta$ and $\min(\beta, N(\neg A \mid B)) < \alpha_1$.*

In any case above, if the argument $\langle \mathcal{A}_1, Q_1, \alpha_1 \rangle$ is canonical, it will be called canonical *(proper or blocking)* defeater*.*

Example 12 *Following Examples 3 and 9, it is the case that the argument $\langle \mathcal{A}_2, \sim engine\_ok, 0.8 \rangle$ is a proper defeater for the 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 for $\langle \mathcal{C}_1, oil\_ok, 0.8 \rangle$.*

Example 13 *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, and thus, A_1 counterargues A_2, and viceversa. However, since we have

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

one can only derive arguments for the negated literals $\sim temp(around\_31)$ and $\sim temp(between\_25\_30)$ with necessity degree 0. Hence, A_1 is as a blocking defeater for A_2, and viceversa.

Note that the unification degree between fuzzy constants depends on the context we are considering. For instance, if for the above context $\mathcal{I}_{U,m}$ we would consider the Gödel negation instead of the standard 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 would get

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

However, as $0.2 < 0.45$ and $0.2 < 0.7$, in this new particular context we would still have that A_1 is blocking defeater for A_2 and viceversa.

6 Computing warranted arguments in DePGL$^+$

As already explained in Section 2, argument-based inference involves a dialectical process in which arguments are compared in order to determine which beliefs are ultimately accepted (or *warranted*) on the basis of a given knowledge base. In the case of argument-based logic programming, such knowledge base is given by the underlying logic program (in our case, a DePGL$^+$ program).

Skeptical argument-based semantics [18,28] are commonly used for computing warranted arguments. The intuition behind such skeptical approaches to the notion of warrant (using the object language of DePGL$^+$) can be defined as follows:

(1) An argument $\langle \mathcal{A}, Q, \alpha \rangle$ is warranted if $\langle \mathcal{A}, Q, \alpha \rangle$ has no defeaters;
(2) An argument $\langle \mathcal{A}, Q, \alpha \rangle$ is warranted if it has defeaters $\langle \mathcal{B}_1, Q_1, \beta_1 \rangle, \ldots,$ $\langle \mathcal{B}_k, Q_k, \beta_k \rangle$, such that every defeater $\langle \mathcal{B}_i, Q_i, \beta_i \rangle$, $(1 \leq i \leq k)$ is in turn defeated by a warranted argument.

In DeLP and in P-DeLP the above intuition is formalized in terms of an exhaustive dialectical analysis of all possible argumentation lines rooted in a given argument (see [16] for details) which can be efficiently performed by means of a top-down algorithm, as described in [14].

Example 14 *Given the following simple P-DeLP program* $\mathcal{P} = \{(p, 0.45), (\sim p, 0.7)\}$*, we can see that* $A = \langle \{(\sim p, 0.7)\}, \sim p, 0.7 \rangle$ *is warranted, as there is no argument defeating* A *from the program* $\mathcal{P}$*. Similarly, we can conclude that the argument* $A' = \langle \{(p, 0.45)\}, p, 0.45 \rangle$ *is not warranted, as argument* A *is a proper defeater for the argument* A'*. Argument* A' *is therefore not warranted as it is defeated by a warranted argument,*

In DePGL$^+$, one can perform a similar dialectical analysis provided some care is taken with the management of fuzzy constants and their associated fuzzy unification mechanism as we show in the following example.

Example 15 *Consider the DePGL$^+$ program* $\mathcal{P}$ *and the context* $\mathcal{I}_{U,m}$ *of Example 13. 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*

$$m(about\_25) = [24, 25, 25, 26],$$

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

Notice that arguments A_1 and A_2 from Example 13 are still arguments with respect to the new program $\mathcal{P}'$. Now, in the frame of the program $\mathcal{P}'$, from the canonical argument associated with 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 $N(m(between\_31\_32) \mid \max(1 - 0.45, m(around\_31))) = \min(0.45, 1) = 1$. *One can easily check that, as in the case of A_1, A_3 is a blocking defeater for A_2, and viceversa. 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 a spurious blocking defeater 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, as $m(about\_25) \cap m(around\_31)$ is a non-normalized fuzzy set, the argument A_4 counterargues the argument A_1, and viceversa. Moreover, as

$$N(m(\neg around\_31) \mid m(about\_25)) = N(m(\neg about\_25) \mid m(around\_31)) = 1$$

and $0.9 > 0.45$, *the argument A_4 is a proper defeater for the argument A_1. Now, from the canonical argument attached with A_4 and by applying the SUA procedural rule, we can build the argument*

$$A_5 = \langle \mathcal{A}_3, temp(about\_25\_ext), 0.9\rangle,$$

since $N(m(about\_25\_ext) \mid \max(1 - 0.9, m(about\_25))) = \min(0.9, 1) = 0.9$. *As $m(about\_25\_ext) \cap m(around\_31)$ is a non-normalized fuzzy set, the argument A_5 counterargues the argument A_1, and viceversa. Moreover, as it holds that $N(m(\neg around\_31) \mid m(about\_25\_ext)) = 0.5$, $N(m(\neg about\_25\_ext) \mid m(around\_31)) = 0$ and* $\min(0.9, 0.5) > 0.45$, *the argument A_5 is a proper defeater for the argument A_1. However, as the fuzzy constant "$about\_25$" is more* specific *than the fuzzy constant "$about\_25\_ext$", the argument A_5 can be considered a spurious proper defeater for the argument A_1.*

Considering suitable extensions (by adding ambiguity) of fuzzy constants one can find multiple spurious (proper and blocking) defeaters for arguments. Then, in order to provide DePGL$^+$ with an efficient procedure for computing warrants (based on an exhaustive dialectical analysis of all argumentation lines), we have to restrict ourselves to canonical defeaters. The formalization of the notion of argumentation line in the framework of DePGL$^+$ is done as follows. An *argumentation line* starting in an argument $\langle \mathcal{A}_0, Q_0, \alpha_0 \rangle$ is a sequence of arguments

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

where each $\langle \mathcal{A}_i, Q_i, \alpha_i \rangle$ is *a defeater* for the previous argument $\langle \mathcal{A}_{i-1}, Q_{i-1}, \alpha_{i-1} \rangle$ in the sequence, $i > 0$.

In order to avoid fallacious reasoning, most argument-based approaches impose additional constraints on such an argument exchange to be rationally acceptable (see e.g. [22,10]). In particular, for DeGLP+ we impose the following constraints on the argumentation lines:

(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 blocking defeater $\langle \mathcal{A}_i, Q_i, \alpha_i \rangle$ in λ with $i > 0$ is defeated by a proper defeater[6] $\langle \mathcal{A}_{i+1}, Q_{i+1}, \alpha_{i+1} \rangle$ in λ.
(3) **Canonicity:** every argument $\langle \mathcal{A}_i, Q_i, \alpha_i \rangle$ in λ with $i > 0$ is canonical; i.e. $\langle \mathcal{A}_i, Q_i, \alpha_i \rangle$ is the best proper or blocking defeater one can consider from a given set of clauses.

An argumentation line satisfying these three conditions are called *acceptable.* 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. The third condition avoids the use of spurious defeaters, due to the application of the SUA inference rule, with weaker information than what it actually could carry, and thus able to be potentially defeated by stronger counter-arguments. The enforced use of canonical arguments in the process of exchange of arguments ensures that both the proponent and the opponent are arguing with the best arguments for a given goal at hand. Moreover, it also enforces that both the length of acceptable argumentation lines and the number of acceptable argumentation lines rooted in a given argument is finite, since for any subset of uncertain clauses $\mathcal{A} \subseteq \Delta$ and each predicate p appearing in a given program (there are finitely-many such predicates), there can be at most one canonical argument of the kind $(\mathcal{A}, p(C), 1)$.

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

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 $\langle \mathcal{A}_0, Q_0, \alpha_0 \rangle$.

Definition 16 (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) there is no other argument of the form $\langle \mathcal{A}_1, Q, \beta \rangle$, with $\beta > \alpha$, satisfying the above.

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. 16, we will also say that the argument $\langle \mathcal{A}, Q, \alpha \rangle$ is warranted. Continuing with Examples 13 and 15, 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 17 *Consider the DePGL$^+$ program $\mathcal{P}'$ and the context $\mathcal{I}_{U,m}$ of Example 15. Further, consider two new fuzzy constants "between_25_30$_{0.7}$" and "about_25$_{0.9}$" interpreted in the context $\mathcal{I}_{U,m}$ as*

$$m(between\_25\_30_{0.7}) = \max(1 - 0.7, m(between\_25\_30)), \textit{ and}$$
$$m(about\_25_{0.9}) = \max(1 - 0.9, m(about\_25)).$$

Let us recall the following arguments:

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

Consider first the argument A_4. On the one hand, it has neither a proper defeater nor a blocking defeater, 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 16, the goal "temp(about_25)" can be warranted wrt $\mathcal{P}'$ in the context $\mathcal{I}_{U,m}$ with a maximum necessity degree of 0.9. *On the other hand, the canonical argument attached with A_4 is*

$$A_6 = \langle \mathcal{A}_3, temp(about\_25_{0.9}), 1 \rangle$$

and, obviously, A_6 is also warranted wrt $\mathcal{P}'$ in the context $\mathcal{I}_{U,m}$.

Consider now the case of argument A_1. On the one hand, the argument A_6 is a canonical proper defeater for A_1 and A_6 is a warranted argument. On the other hand, the canonical argument attached with A_2 is

$$A_7 = \langle \mathcal{A}_2, temp(between\_25\_30_{0.7}), 1\rangle$$

and A_7 is a canonical blocking defeater for A_1. Therefore two acceptable argumentation lines rooted at A_1 can be built: $[A_1, A_6]$ and $[A_1, A_7]$. 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. However, the goal $temp(between\_25\_30)$ can be warranted from $\mathcal{A}_3$ with the maximum necessity degree of 0.9 as follows: From the canonical argument A_6, by applying the SUA procedural rule, we get the argument

$$A_8 = \langle \mathcal{A}_3, temp(between\_25\_30), 0.9\rangle$$

since $N(m(between\_25\_30) \mid m(about\_25_{0.9}) = 0.9$, and obviously, A_8 is also warranted wrt $\mathcal{P}'$ in the context $\mathcal{I}_{U,m}$.

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 [14].

7 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 [30,29,31]. 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 [31]. Their proposal, on the other hand, does not include an explicit treatment of possibilistic uncertainty as in our case.

There have been different approaches connecting argumentative inference, defeasible reasoning and possibilistic logic (e.g.[9,7,8]). Including possibilistic logic as part of an argumentation framework for modelling preference handling and information merging has recently been considered by Amgoud & Kaci [6] and Amgoud & Cayrol [5]. 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 smoothly integrates an explicit representation of fuzziness together with a possibilistic uncertainty handling. Indeed, in the proposed framework we attach necessity degrees to object level formulas, which are propagated according to suitable inference rules and play an important role in determining the final status of arguments.

Besides of considering possibilistic logic and fuzziness, a number of hybrid approaches connecting argumentation and uncertainty have been developed, such as Probabilistic Argumentation Systems [20,21], which use probabilities to compute degrees of support and plausibility of goals, related to Dempster-Shafer belief and plausibility functions. However this approach is not based on a dialectical theory (with arguments, defeaters, etc.) nor includes fuzziness as presented in this paper. In a recent paper [23] a declarative language to handle arguments with modalities like *possible*, *probable*, *plausible*, etc. is proposed. The resulting framework is applied to modelling problems in the context of a medical domain. In contrast with our approach, no possibilistic logic semantics is associated with the framework, as modalities are categorized in terms of a declarative semantics formalized on the basis of a complete lattice. Besides, no representation of fuzziness at object level is provided in this framework, as in the case of our proposal.

8 Conclusions and future work

In this paper we have provided a formalization of DePGL$^+$, a possibilistic defeasible logic programming language that integrates argumentation capabilities and the characterization of fuzziness at object level in terms of fuzzy constants and fuzzy propositional variables. Our extended framework is motivated on previous research which showed how to successfully integrate defeasible argumentation and possibilistic uncertainty [16]. We have shown how PGL$^+$ can be suitably adapted to be included in an argument-based setting. 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. It must be remarked that the notions of argument, defeat and dialectical analysis –common to all argumentation frameworks– could be naturally borrowed into our formalization, and their expressivity was augmented by the incorporation of fuzziness, integrated in the argument-based inference process (rules INTF, MPA, SUA and INA). However, as discussed in Section 5, the notion of canonicity of an argument was an additional requirement in the new, extended framework, needed to ensure the proper computation of argumentation lines (as discussed in Section 5) by enforcing that the number of argumentation lines rooted in any argument be finite.

Part of our current work is focused on studying complexity issues in the context of our proposal, as well as emerging logical properties which could help to speed up computation of warranted arguments. In that respect, we think that many of the results already available for PGL$^+$ can be used as a basis for exploring such possibilities in the context of DePGL$^+$. It must be also noted that we have not introduced *default negation* in DePGL+, even though this form of negation is available in DeLP [19] (where an extended literal *not* p is proven iff the literal p fails to be ultimately acceptable). Part of our current work involves exploring the inclusion of default negation into our formalism. 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 [26]. Research in these directions is currently being pursued.

As for the knowledge representation capabilities of DePGL$^+$, the formalism proposed has some representation limitations due to the restriction of allowing only unary predicates. Clearly,having an underlying full predicate logic would make the framework more powerful. Indeed, PGL$^+\forall$, the first order extension of PGL$^+$, has been already developed in [4], so it remains as an interesting future work to extend the argumentative framework over PGL$^+\forall$. Given that this would considerably increase the technical complexity of the paper without providing new conceptual insights, we also leave it as a future task to develop.

Acknowledgments

We thank anonymous reviewers for their comments and suggestions to improve the final version of this paper. This research was partially supported by CICYT Projects MULOG (TIN2004-07933-C03-01/03) and IEA (TIN2006-15662-C02-01/02), by CONICET (Argentina), by the Secretaría General de Ciencia y Tecnología de la Universidad Nacional del Sur (Project 24/ZN10), and by Agencia Nacional de Promoción Científica y Tecnológica (PICT 2002 No. 13096).

References

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

[2] T. Alsinet and L. Godo. A proof procedure for possibilistic logic programming with fuzzy constants. In *Proc. of the ECSQARU-2001 Conf., Lecture Notes in Artificial Intelligence 2143*, pages 760–771, 2001.

[3] T. Alsinet, L. Godo, and S. Sandri. Two formalisms of extended possibilistic logic programming with context-dependent fuzzy unification: a comparative description. *Electronic Notes in Theoretical Computer Science*, 66(5), 2002.

[4] T. Alsinet. *Logic Programming with Fuzzy Unification and Imprecise Constants: Possibilistic Semantics and Automated Deduction.* Number 15 in Monografies de l' IIIA. Institut d'Investigació en Intel.ligencia Artificial (IIIA-CSIC). Bellaterra, Spain, 2003.

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

[6] L. Amgoud and S. Kaci. An argumentation framework for merging conflicting knowledge bases: The prioritized case. In Lluis Godo, editor, *Intl. ECSQARU Conference*, volume 3571 of *Lecture Notes in Computer Science*, pages 527–538. Springer, 2005.

[7] S. Benferhat, D. Dubois, and H. 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.

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

[9] S. Benferhat, D. Dubois, and H. Prade. Argumentative inference in uncertain and inconsistent knowledge base. In *Proceedings of the 9th Annual Conference on Uncertainty in Artificial Intelligence (UAI-93)*, pages 411–419, San Francisco, CA, 1993. Morgan Kaufmann.

[10] P. Besnard and A. Hunter. A logic-based theory of deductive arguments. *Artif. Intell.*, 128(1-2):203–235, 2001.

[11] R. Brena, J. Aguirre, C. Chesñevar, E. Ramirez, and L. Garrido. Knowledge and information distribution leveraged by intelligent agents. *Knowledge and Information Systems*, 2006.

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

[13] C. I. Chesñevar, A. Maguitman, and M. Sabaté. An argument-based decision support system for assessing natural language usage on the basis of the web corpus. *International Journal of Intelligent Systems (IJIS)*, 21(11):1151–1180, 2006.

[14] C. I. Chesñevar, G. Simari, and L. Godo. Computing dialectical trees efficiently in possibilistic defeasible logic programming. *LNAI/LNCS Springer Series (Proc. of the 8th Intl. Conference on Logic Programming and Nonmonotonic Reasoning LPNMR 2005)*, pages 158–171, September 2005.

[15] C. I. Chesñevar, A. G. Maguitman, and G. R. Simari. Argument-Based Critics and Recommenders: A Qualitative Perspective on User Support Systems. *Journal of Data and Knowledge Engineering*, 59(2):293–319, 2006.

[16] C. I. Chesñevar, G. Simari, T. Alsinet, and L. Godo. A Logic Programming Framework for Possibilistic Argumentation with Vague Knowledge. In *Proc. of UAI 2004. Banff, Canada*, pages 76–84, July 2004.

[17] D. Dubois, J. Lang, and H. Prade. Possibilistic logic. In D.Gabbay, C.Hogger, and J.Robinson, editors, *Handbook of Logic in Art. Int. and Logic Prog. (Nonmonotonic Reasoning and Uncertain Reasoning)*, pages 439–513. Oxford Univ. Press, 1994.

[18] P. Dung. On the acceptability of arguments and its fundamental role in nonmonotonic reasoning and logic programming and n-person games. *Artificial Intelligence*, 77:321–357, 1995.

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

[20] R. Haenni, J. Kohlas, and N. Lehmann. Probabilistic argumentation systems. In *Handbook of Defeasible Reasoning and Uncertainty Management Systems.* Kluwer, 2000.

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

[22] A. Kakas and F. Toni. Computing argumentation in logic programming. *Journal of Logic Programming*, 9(4):515:562, 1999.

[23] J. C. Nieves and U. Cortés. Modality argumentation programming. In *Modeling Decisions for Artificial Intelligence, Third International Conference, MDAI 2006, Tarragona, Spain, April 3-5, 2006, Proceedings (LNCS 3885, Springer)*, pages 295–306, 2006.

[24] J. L. Pollock. *Knowledge and Justification.* Princeton, 1974.

[25] J. L. Pollock. Defeasible Reasoning. *Cognitive Science*, 11:481–518, 1987.

[26] J. L. Pollock. Defeasible reasoning with variable degrees of justification. *Artificial Intelligence*, 133(1-2):233–282, 2001.

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

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

[29] M. Schroeder and R. Schweimeier. Fuzzy argumentation for negotiating agents. In *The First International Joint Conference on Autonomous Agents & Multiagent Systems, AAMAS 2002, July 15-19, 2002, Bologna, Italy, Proceedings*, pages 942–943. ACM, 2002.

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

[31] R. Schweimeier and M. Schroeder. Fuzzy unification and argumentation for well-founded semantics. In *SOFSEM 2004: Theory and Practice of Computer Science, 30th Conference on Current Trends in Theory and Practice of Computer Science, Merin, Czech Republic*, pages 102–121. Springer (LNCS 2932), 2004.