Electronic Notes in Theoretical Computer Science 66 No. 5 (2002)
URL: http://www.elsevier.nl/locate/entcs/volume66.html 21 pages

Two formalisms of extended possibilistic logic programming with context-dependent fuzzy unification: a comparative description

Teresa Alsinet [1]

Computer Science Department
Universitat de Lleida (UdL)
25001 Lleida, Spain

Lluís Godo [2]

Institut d'Investigació en Intel·ligència Artificial (IIIA)
Consejo Superior de Investigaciones Científicas (CSIC)
08913 Bellaterra, Spain

Sandra Sandri [3,4]

Lab. Associado de Computação e Mat. Aplicada (LAC)
Instituto Nacional de Pesquisas Espaciais (INPE)
1220-970 S.J. dos Campos (SP), Brasil

Abstract

Possibilistic logic 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 this paper we present a comparative description of two models extending first order possibilistic logic so as to allow for fuzzy unification. The first formalism, called PLFC, is a general extension that allows clauses with fuzzy constants and fuzzily restricted quantifiers. The second formalism is an implication-based extension defined on top of Gödel infinitely-valued logic, capable of dealing with fuzzy constants. In this paper we compare these approaches, mainly their Horn-clause fragments, discussing their basic differences, specially in what regards their unification and automated deduction mechanisms.

[1] Partial support by project LOGFAC (TIC 2001-1577-C03-03).
Email: tracy@eup.udl.es
[2] Partial support by project LOGFAC (TIC 2001-1577-C03-01).
Email: godo@iiia.csic.es
[3] Email: sandri@iiia.csic.es
[4] On sabbatical leave at IIIA - CSIC, Spain

1 Introduction

Fuzzy unification has attracted attention in the recent past and a number of approaches have been proposed in the literature. Some of them allow degrees of partial matching between classical logical objects (propositions, constants, predicates) while other ones can support both the handling of imprecise or fuzzy information as well as a graded pattern macthing mechanism.

The introduction of fuzzy constants in logic programming languages was suggested in the early eighties by Cayrol et al. [10] and Bel et al. [8] with the aim of including fuzzy values in a pattern matching procedure. Subsequently, Umano [24] defined a fuzzy pattern matching process using the extension principle for one and two variate functions, and Baldwin et al. [7] implemented a semantic unification procedure based on the theory of mass assignments which allows a unified framework for the treatment of fuzzy and probabilistic data. Godo and Vila [15] proposed a possibilistic-based logic to deal with fuzzy temporal constraints based on many-valued semantics and a necessity-like measure to allow a pattern matching mechanism between fuzzy temporal constraints. Virtanen [25] defined a fuzzy unification algorithm based on fuzzy equality relations and Rios-Filho and Sandri [13] addressed the problem of unification involving fuzzy constants in which a separation between general and specific patterns can be drawn. Arcelli et al. [6] proposed three different kinds of unification in the fuzzy context: the first one is based on similarity relations, the second one identifies similar objects through an equivalence relation and the last one uses "semantic constraints" for defining a more flexible unification. More recently, Gerla and colleagues [14,16] formalized a methodology for transforming an interpreter for SLD Resolution into an interpreter that computes on abstract values which express similarity properties on the set of predicate and function symbols of the language. Also a similarity-based approach to unification, in the framework of generalized many-valued logic programming (called multi-adjoint logic programming) can be found in [28,20].

Concerning general approaches to deal both with imprecision and fuzzy unification, Dubois, Prade and Sandri [12] proposed an extension of possibilistic logic dealing with fuzzy constants and fuzzily restricted quantifiers (called PLFC), and [5] provided PLFC logic with a formal semantics and a sound resolution-style calculus by refutation. Possibility theory is a framework in which imprecision, fuzziness and uncertainty can be dealt with in a uniform way. Possibilistic logic (PL) [11] is a logic of uncertainty where to each classical proposition or closed first order formula is attached a certainty degree between 0 and 1, which is interpreted as a lower bound of a necessity measure. For instance, the uncertain statement "it is almost sure that Konstanze likes sun-bathing " can be represented by a *certainty-weighted* formula of the form

$$(likes\_Konstanze(sun - bathing),\ 0.9)\ \ .$$

To enhance its knowledge representation power, Dubois et al. [12] defined a syntactic extension to first order possibilistic logic, called PLFC, capable of dealing with fuzzy constants and fuzzily restricted quantifiers inside the language.

Fuzzy constants represent disjunctive knowledge and can be seen as (flexible) restrictions on an existential quantifier. For instance, the fuzzy statement "Peter is *about*_35 years old" can be represented by a formula of the form

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

where *age_Peter* is a classical predicate and *about*_35 is a fuzzy constant defined over the domain $[0, 120]$ (years). In the case *about*_35 denotes a crisp interval of ages, say $[34, 36]$, the formula $age\_Peter(about\_35)$ is to be interpreted as

$$\text{“}\exists\ x \in [34, 36] \text{ such that } age\_Peter(x)\text{”}\quad .$$

In case *about*_35 denotes a fuzzy interval the semantics of the formula is more complex and will be specified later on.

On the other hand, the introduction of variable weights in PL (again see e.g. [11]) is a suitable technique for modeling statements of the form

"the more x is A (or x belongs to A), the more certain is $p(x)$",

where A is a fuzzy set with membership function $\mu_A(x)$. This is formalized in PLFC as

"for all x, $p(x)$ is certain with a necessity of at least $\mu_A(x)$",

and is represented as $(p(x), A(x))$. When A is imprecise but not fuzzy, the interpretation of such a formula is just

"for all $x \in A$, $p(x)$".

So variable weights in PLFC act equivalently as (flexible if they are fuzzy) restrictions on a universal quantifier or as a kind of conjunctive constants.

Alsinet et al. [5] defined for PLFC a formal semantics and a sound resolution-style calculus by refutation, where the resolution rule includes an *implicit fuzzy unification* mechanism between fuzzy constants. In [18,19], a Horn clause formalism for this logic, called PLFC-H, was studied and a refutation based theorem prover for it was developed using system KOMET [9]. This was made possible by first translating PLFC-H to the generalized annotated logic formalism introduced by [23,17], which underlies system KOMET knowledge representation and inference mechanism.

Alternatively to PLFC, Alsinet and Godo defined in [1] an extension of possibilistic logic, called PGL, defined on top of Gödel infinitely-valued logic, and in [2] the Horn-rule sublogic of PGL was extended with fuzzy constants (but not with variable weights) and with a modus ponens-style calculus based on an *explicit fuzzy unification* mechanism between fuzzy constants that was shown to be complete for a restricted class of Horn-rules. This logic is called PGL^+.

In this paper we review and compare both extensions of possibilistic logic and focus on the main differences between them, specially from the point of view of their semantics and their unification and automated deduction mechanisms. We shall emphasize that the semantics adopted in both languages (disjunction-based in PLFC, implication-based on PGL) is crucial for adopting a particular *sound unification* mechanism. In both formalisms, unification between two object (fuzzy) constants is a matter of degree and it is computed in terms of a necessity measure (slightly different from one another) between fuzzy sets. But, in contrast to classical logic and other frameworks, this matching process is in most of the cases directional and thus non symmetric, actually only when precise constants are involved it reduces to classical unification.

In the next section we review main concepts of possibilistic logic. In Section 3 we describe formal aspects of PLFC while in Section 4 we focus on PLFC-H. The possibilistic logic programming formalism PGL^+ is described in Section 5. Finally, in Section 6 we conclude with a discussion on the main differences between PLFC (and PLFC-H) and PGL^+.

2 Background on necessity-valued possibilistic logic

In necessity-valued possibilistic logic each formula is represented by a pair (φ, α), φ being a classical, first order logic formula and $\alpha \in (0, 1]$ being a lower bound on the belief on φ in terms of necessity measures. A formula (φ, α) is thus interpreted as a constraint $N(\varphi) \geq \alpha$, where N is a necessity measure on propositions, a mapping from the set of logical formulae to a totally ordered bounded scale, usually (but not necessarily) given by [0, 1], characterized by the axioms

(i) $N(\top) = 1$,

(ii) $N(\bot) = 0$,

(iii) $N(\varphi \wedge \psi) = min(N(\varphi), N(\psi))$,

(iv) $N(\varphi) = N(\psi)$ when φ and ψ are classically equivalent,

where $\top$ and $\bot$ denote respectively tautology and contradiction.

The necessity-valued possibilistic logic (simply possibilistic logic from now on) is axiomatized (Hilbert-style) by the axioms of classical first-order logic weighted by 1, together with the following graded versions of the usual *modus ponens* and *generalization* inference rules,

$$\frac{(\varphi, \alpha), (\varphi \rightarrow \psi, \beta)}{(\psi, \min(\alpha, \beta))}[MP], \quad \frac{(\varphi, \alpha)}{((\forall x)\varphi, \alpha)}[G]$$

together with a *weight weakening* rule

$$\frac{(\varphi, \alpha)}{(\varphi, \beta)}[W]$$

for $\beta \leq \alpha$. We shall denote by $\vdash_{PL}$ the notion of proof in possibilistic logic derived from this formal system of axioms and rules.

Deduction by resolution has been easily adapted to possibilistic logic. Indeed, let K be a knowledge base formed by possibilistic clauses, i.e. possibilistic formulas of the type (ψ, α), where ψ is a (first-order or propositional) clause in the usual sense. Namely, we write $K \vdash^r_{PL} (\varphi, \alpha)$ to denote that we obtain a proof of $(\bot, \alpha)$ by successively applying the *resolution* rule

$$\frac{(\neg p \vee q, \alpha), (p \vee r, \beta)}{(q \vee r, \min(\alpha, \beta))}[Res]$$

in $K \cup \{(\neg\delta_i, 1) \mid i = 1, n\}$, where $\bigwedge_{i=1,n} \delta_i$ is the clausal form of φ. Then it holds that $K \vdash_{PL} (\varphi, \alpha)$ iff $K \vdash^r_{PL} (\varphi, \alpha)$. Moreover, using the $\vdash^r_{PL}$ procedure, other rules can be derived, for instance the *fusion* rule

$$\frac{(p, \alpha), (p, \beta)}{(p, max(\alpha, \beta))}$$

and when the unification mechanism employed in the resolution rule is the same as in classical logic, the following *particularization* rule is also derivable:

$$\frac{((\forall x)p(x), \alpha)}{(p(s), \alpha)} .$$

Now, let us recall here the usual (monotonic) semantics for possibilistic logic. For the sake of an easier understanding we consider the propositional case, the first order case being an easy extension. We shall make use of the following notation. Let L be a propositional language and let Ω be the set of classical interpretations for L, that is, the set of evaluations w of the atoms of the language into the boolean truth value set $\{0, 1\}$. Each evaluation of atoms w extends to any clause in the usual way, and thus for any φ, $w(\varphi) \in \{0, 1\}$. For any clause φ, we will write $w \models \varphi$ iff $w(\varphi) = 1$. We shall also write $[\varphi]$ to denote the set of models of φ, i.e. $[\varphi] = \{w \in \Omega \mid w \models \varphi\}$.

Belief states are modeled by normalized possibility distributions $\pi : \Omega \to [0, 1]$ on the set of possible interpretations, where $\pi(w) < \pi(w')$ means that w' is a more plausible interpretation than w. A possibility distribution π is normalized when there is at least one $w \in \Omega$ such that $\pi(w) = 1$. In other words, belief states modeled by normalized distributions are consistent states, in the sense that at least one interpretation (or state or possible world) has to be fully plausible. The satisfaction relation between possibilistic models (i.e. possibility distributions) and possibilistic formulas is defined as follows:

$$\pi \models (\varphi, \alpha) \text{ iff } N([\varphi] \mid \pi) \geq \alpha,$$

where $N(. \mid \pi)$ is the necessity measure induced by π on the power set of Ω, defined as

(1) $$N([\varphi] \mid \pi) = \inf_{w \in \Omega} \max(1 - \pi(w), w(\varphi)) = \inf_{w \not\models \varphi} 1 - \pi(w).$$

If $\pi \models (\varphi, \alpha)$ we say that π is a model of (φ, α). As usual, if Γ denotes a set of possibilistic clauses, we say that π is a model of Γ iff π is a model of each formula in Γ. The possibilistic entailment, denoted $\models_{PL}$, is then defined as follows.

$$\Gamma \models_{PL} (\varphi, \alpha) \text{ iff } \pi \models (\varphi, \alpha),$$

for each π being model of Γ. Dubois, Lang and Prade have shown [11] that this semantics makes possibilistic logic sound and complete, and moreover, using refutation, the resolution-based proof system is also sound and complete wrt to the above semantics, that is, the following equivalences hold:

$$\Gamma \models_{PL} (\varphi, \alpha) \text{ iff } \Gamma \vdash_{PL} (\varphi, \alpha) \text{ iff } \Gamma \vdash^{r}_{PL} (\varphi, \alpha).$$

3 PLFC

As already mentioned, PLFC is an extension of possibilistic logic that provides a powerful framework for reasoning under possibilistic uncertainty and representing disjunctive and conjunctive vague knowledge. Following [5], a general PLFC clause is a pair of the form

$$(\varphi(\bar{x}), f(\bar{y})),$$

where $\bar{x}$ and $\bar{y}$ denote sets of free and implicitly universally quantified variables, each one having its sort, such that $\bar{y} \supseteq \bar{x}$; $\varphi(\bar{x})$, called *base formula*, is a disjunction of (positive and negative) literals with typed classical predicates and possibly with fuzzy constants, each one having its sort; and $f(\bar{y})$ is a well-formed valuation function, defined for a superset of the variables in the left-hand side, which expresses a lower bound of the certainty of $\varphi(\bar{x})$ in terms of necessity measures. Basically, valuation functions $f(\bar{y})$ are either constant values in the real interval $[0, 1]$, or membership functions of fuzzy sets (fuzzy constants), or max-min combinations of them, or necessity measures on them. An example of PLFC clause may be

$$(p(A, x) \vee q(y), \min(\alpha, B(x), C(y)))$$

where A, B and C are fuzzy constants.

Next, let us briefly recall PLFC semantics. A *many-valued interpretation* $w = (U, i, m)$ maps:

(i) each sort σ into a non-empty domain U_σ of U;

(ii) a predicate p of type $(\sigma_1, \ldots, \sigma_n)$ into a *crisp* relation $i(p) \subseteq U_{\sigma_1} \times \ldots \times U_{\sigma_n}$; and

(iii) an object constant A (precise or fuzzy constant) of sort σ into a normalized fuzzy set $m(A)$ with membership function $\mu_{m(A)} : U_\sigma \to [0,1]$. We denote by $\mu_{m(A)}$ the membership function of $m(A)$. When A is a precise constant c, then $\mu_{m(A)}$ will represent the singleton $m(c)$.

An *evaluation* of variables is a mapping e associating to each variable x of sort σ an element $e(x) \in U_\sigma$. The truth value of a base formula under an interpretation $w = (U, i, m)$ and an evaluation of variables e is defined by cases:

(i) $w_e(p(x, \ldots, A)) = \sup_{(u,\ldots,v)\in i(p)} \min(\mu_{e(x)}(u), \ldots, \mu_{m(A)}(v))$.

(ii) $w_e(\neg p(x, \ldots, A)) = \sup_{(u,\ldots,v)\notin i(p)} \min(\mu_{e(x)}(u), \ldots, \mu_{m(A)}(v))$.

(iii) $w_e(L_1 \vee \cdots \vee L_r) = \max(w_e(L_1), \ldots, w_e(L_r))$, where $L_1, \ldots, L_r$ are (positive or negative) literals.

Finally, the truth value of a base formula φ under an interpretation w is defined as $w(\varphi) = \inf\{w_e(\varphi) \mid e \text{ is an evaluation of variables}\}$. Notice that the negation in this semantics is not truth-functional. Moreover, $w(\varphi)$ may take any intermediate value between 0 and 1 as soon as φ contains some fuzzy constant and $w(\varphi)$ depends not only on the crisp relations assigned to predicate symbols, but on the fuzzy sets assigned to fuzzy constants. Then, in order to define the possibilistic semantics, we need to fix a *context* and to consider some extension for fuzzy sets (of interpretations) of the standard notion of necessity measure.

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 $\Omega_{U,m}$ is just the set $\{w \text{ interpretation} \mid w = (U, i, m)\}$. Now, for each possibility distribution on the context $\pi : \Omega_{U,m} \to [0,1]$, and each PLFC formula (φ, α), we define

$$\pi \models (\varphi, \alpha) \text{ iff } N_1([\varphi] \mid \pi) \geq \alpha,$$

where $N_1(\cdot \mid \pi)$ is the necessity measure induced by π on fuzzy sets of interpretations defined by

$$N_1([\varphi] \mid \pi) = \inf_{w\in\Omega_{U,m}} \max(1 - \pi(w), w(\varphi)), \tag{2}$$

where we take $\mu_{[\varphi]}(w) = w(\varphi)$. Here, we have considered a PLFC formula with a constant weight α. If the formula has a variable weight, e.g. $(\varphi(x), A(x))$, then the above definition, always in the same context, extends to

$$\pi \models (\varphi(x), A(x)) \text{ iff } \pi \models (\varphi(c), \mu_{m(A)}(m(c)))$$

for all precise object constants c.

An interesting and remarkable consequence of possibilistic satisfiability of

PLFC clauses is the following one:

$$\pi \models (p(A) \vee q(B), \alpha) \text{ iff } \pi \models (p([A]_\alpha) \vee q([B]_\alpha), \alpha),$$

where p and q can be positive or negative literals, and $[A]_\alpha$ and $[B]_\alpha$ denote the imprecise constants corresponding to the α-cuts of the fuzzy constants A and B, respectively. This property has important consequences since it means that in PLFC with (only) fuzzy constants we can in a way forget about fuzzy constants as such and focus only on imprecise but crisp constants. For instance, in PLFC, the fuzzy statement "it is almost sure that Peter is *about_35* years old" can be represented by a *certainty-weighted* formula of the form

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

where *age_Peter* is a classical predicate of type (`years_old`) and *about_35* is a fuzzy constant defined over the domain `years_old` $[0, 120]$ (years). In the case *about_35* denotes a crisp interval of ages, say $[34, 36]$, the formula is to be interpreted as

"$\exists\ x \in [34, 36]$ s.t. $age\_Peter(x)$" is certain with a necessity of at least .9.

In the case *about_35* denotes a fuzzy interval with a membership function $\mu_{about\_35} : [0, 120] \to [0, 1]$, the formula is to be interpreted as

"$\exists\ x \in [\mu_{about\_35}]_{.9}$ s.t. $age\_Peter(x)$" is certain with a necessity of at least .9,

where $[\mu_{about\_35}]_{0.9}$ denotes the crisp interval of ages associated with the α-cut of the fuzzy set μ_{about_35} at the level of 0.9.

Notation convention: Since we need to fix a context $\Omega_{U,m}$ in order to perform deduction, we can identify a fuzzy constant A with its interpreted fuzzy set $m(A)$ and also with its membership function $\mu_{m(A)}$. Hence, for the sake of a simpler notation we shall consider fuzzy constants simply as fuzzy sets. Further, if A and B are fuzzy constants, $A \cap B$ and $A \cup B$ will refer to their fuzzy set min-intersection and max-union respectively.

One of the main advantages of the present semantics for PLFC is that it provides a *sound refutation by resolution* proof mechanism. Given a context $\Omega_{U,m}$, the *PLFC resolution rule*, which implicitly manages the unification mechanism between fuzzy constants, can be generalized as follows:

$$\frac{(\neg p(x) \vee \varphi(x), \min(\alpha, A(x))),\ (p(B) \vee \psi, \beta)}{(\varphi(B) \vee \psi, \min(\alpha, \beta, N_1(A \mid [B]_\beta)))}\ [\text{GR}],$$

where $N_1(A \mid [B]_\beta) = \inf_{u \in [B]_\beta} A(u)$. The resolution rule produces conclusions which are all the stronger as $\mu_{m(A)}$ is large and $\mu_{m(B)}$ is small. Therefore, in order to get higher necessity degrees during the refutation proof procedure, it is interesting to have PLFC clauses with larger variable weights. Then, the following *generalized fusion rule* must be applied after each resolution step:

$$\frac{(\varphi(\bar{x}), f_1(\bar{x})),\ (\varphi(\bar{y}), f_2(\bar{y}))}{(\varphi(\bar{x}), \max(f_1(\bar{x}), f_2(\bar{x})))}[\text{FU}].$$

In order to define a refutation proof procedure for PLFC, we cannot borrow the unification concept used in classical first-order logic programming systems. Let us consider one illustrative example. For instance, from

$$(\neg p(A) \vee \psi, 1) \quad \text{and} \quad (p(A), 1),$$

which, if A is not fuzzy, are interpreted respectively as

$$\text{“}[\exists x \in A,\ \neg p(x)] \vee \psi\text{”} \quad \text{and} \quad \text{“}\exists x \in A,\ p(x)\text{”},$$

we can infer ψ iff A is a precise constant. Then, resolution for $\neg p(A)$ and $p(A)$ must fail unless A is a precise constant, even though, obviously, $p(A)\theta = p(A)\theta$ for each (classical) substitution of variables θ. However, as variable weights are interpreted as conjunctive (fuzzy) knowledge, an implicit semantical unification between fuzzy events is performed between variable weights and fuzzy constants. This points out that before applying the FU inference rule to a knowledge base, it is interesting to transform each precise object constant (appearing in the logic component of clauses) into variable weights by means of the following *transformation* rule:

$$\frac{(\varphi(\bar{x}, c), f(\bar{y}))}{(\varphi(\bar{x}, t), \min(f(\bar{y}), c(t)))}\ [\text{TR}],$$

where $t \notin \bar{y}$ and c is a precise constant.

Finally, the *refutation by resolution* proof method is extended to PLFC as follows. Let $K = \{(\varphi_i, f_i) \mid i = 1, \ldots, n\}$ be a set of PLFC clauses and let (φ, f) be a PLFC query of the form $(p(A), \alpha)$ or $(p(x), \min(\alpha, A(x)))$. Then one perfoms the following steps:

(i) Negate the query in the following way:
- $\neg(p(A), \alpha)$ is $(\neg p(x), A(x))$.
- $\neg(p(x), \min(\alpha, A(x)))$ is $(\neg p(A_{>0}), 1)$, where $A_{>0}$ denotes the support of A.

(ii) $K' = K \cup \neg(\varphi, f)$.

(iii) Search for a deduction $(\perp, \alpha)$ from K', in the context determined by U and m, by repeatedly applying the GR, FU and TR inference rules.

(iv) If so, then $K \models (\varphi, f)$.

4 PLFC with Horn clauses

As we have seen, PLFC provides indeed with a powerful framework representing disjunctive and conjunctive fuzzy information, but has some computational limitations, at least as when considered in its whole generality. Namely, on the one hand, the current proof method for PLFC (refutation through a generalized resolution rule, a fusion rule and a transformation rule) is not complete; on the other hand, during the proof process, the merging rule must be applied after every resolution step, and thus the search space consists of all possible orderings of the literals in the knowledge base.

Due to these and other computational limitations, it was considered in [18] to restrict the language to Horn clauses, called PLFC-H, and in [19] an automated theorem prover for PLFC-H was proposed. A possibilistic clause (φ, v) is a *possibilistic Horn clause* if φ is a Horn clause, i.e. φ has at most one positive literal. A possibilistic Horn clause $(p_0 \vee \neg p_1 \vee \ldots \vee \neg p_n, v)$ is denoted by $(p_0 \leftarrow p_1 \wedge ... \wedge p_n, v)$, where p_0 and $p_1 \wedge ... \wedge p_n$ are respectively called the head and the body of the clause. When the head is inexistant, e.g. $(\leftarrow p_1, 1)$ the clause is called a *query*, and when the body is inexistant, e.g. $(p_0 \leftarrow, 1)$, the clause is called a *fact*.

In [19], the working framework is restricted to knowledge bases whose non-query possibilistic Horn clauses obey the following constraints:

- The body of a clause may only contain variables as terms.
- The head of the clause may have terms involving variables, fuzzy functions and imprecise (but not vague) constants.

The restriction on constants does not really affect completely grounded facts since due to PLFC semantics (see [5]) a fact $(p_0(A_1, ..., A_n) \leftarrow, \alpha)$, where the A_i's are fuzzy constants is equivalent to $(p_0([A_1]_\alpha, ..., [A_n]_\alpha) \leftarrow, \alpha)$, which obeys the restrictions. For example, let A and B be imprecise but not vague constants. We write:

- "it is certain to a degree 0.7 that $\forall x \in A$, $p(x) \vee \neg q(x)$" as $(p(x) \leftarrow q(x), min(A(x), 0.7))$,
- "it is certain to a degree 0.7 that $\exists x \in B, p(x)$" as $(p(B) \leftarrow, 0.7)$,
- "it is certain $\forall x \in A, \forall y \in B, p(x \oplus y) \vee \neg q(x) \vee \neg r(y)$" as $(p(x \oplus y) \leftarrow q(x) \wedge r(y), min(A(x), B(y)))$,

where $\oplus$ denotes the "fuzzy sum", i.e. addition extended to fuzzy numbers. These restrictions are valid for all Horn clauses apart from the completely negative ones (queries). For example, during the reasoning process we may have a clause such as $(\leftarrow p(B), 0.7)$ whose interpretation is "it is certain to a degree 0.7 that $\exists x \in B, \neg p(x)$.

In [18], the following restricted resolution rule was proposed for PLFC-H:

$$R_*^\pi : \frac{(p_0 \leftarrow p_1 \wedge ... \wedge p_n, v_p) \ ; \ (\leftarrow q_1 \wedge ... \wedge q_m, v_q)}{[(\leftarrow q_1 \wedge ... \wedge q_{i-1} \wedge p_1 \wedge ... \wedge p_n \wedge q_{i+1} \wedge ... \wedge q_m, min(v_q, v_p))]\theta}$$

where θ is the mgu that unifies p_0 and q_i [5]. This rule is called "restricted" because it only allows for the resolution of a non-negative clause (rule or fact)

[5] The most general unifier (mgu) in PLFC is constructed from a mgu of classical logic, with

with a negative clause (the query). Therefore, a rule $(p_0 \leftarrow p_1 \wedge ... \wedge p_n, v_p)$ and a fact $(q_f \leftarrow, v_f)$ is not resolved together in that framework, even if there exists a mgu θ that unifies q_f and a $p_i, i > 0$, contrary to what would happen with the usual resolution rule. This does not imply a lack of generality, because if p_0 is ever resolved with the j-th query, then p_i will be part of the $j+1$-th query, which can then be unified with the fact.

Let $v = \mu_A(x)$ be a valuation. If, during the deduction process, a mgu θ unifies a variable x with a fuzzy constant B, then the application of θ to v yields $v\theta = N_1(A \mid B) = inf_x \max(\mu_A(x), 1 - \mu_B(x))$. This resolution rule is only sound when all the constants are not fuzzy (cf. with the sound rule GR in the previous section).

In [18], a first modelisation of reduction in PLFC-H was proposed, in order to obtain a higher valuation than the ones obtained only using resolution:

$$D_*^\pi : \frac{(p_0 \leftarrow p_1 \wedge ... \wedge p_n, v_p) \ ; \ (q_0 \leftarrow q_1 \wedge ... \wedge q_m, v_q)}{[(p_0 \leftarrow p_1 \wedge ... \wedge p_n \wedge q_1 \wedge ... \wedge q_m, \max(v_p, v_q))]\theta}$$

where θ is the mgu that unifies p_0 and q_0. However, this rule is not general enough to treat cases as the one involving the following rules and facts:

R1: $(Tan(x, normal) \leftarrow Beach(x, y), \min(France(y), \alpha))$
R2: $(Tan(x, dark) \leftarrow Beach(x, y), \min(Spain(y), \beta))$
R3: $(Tan(x, normal) \leftarrow Solarium(x), \gamma)$
F1: $(Beach(Teresa, Catalonia), 1)$
F2: $(Beach(Konstanze, France), 1)$
F3: $(Solarium(Konstanze), 1)$

where "*normal*" and "*dark*" are precise constants, and "*France*", "*Spain*" and "*Catalonia*" are imprecise but not vague constants respectively.[6]

To the query "Q1: *What kind of tan did Konstanze get?*", the deduction mechanism is capable of yielding the expected answer "*normal*", weighted with a belief of at least $\max(\alpha, \gamma)$, as expected in the context of possibilistic theory. The query "Q2: *What kind of tan did Teresa get?*" is more complex and cannot be treated by the resolution rule above. Resolving R1 and F2 together would yield a fail (idem in relation to R2 and F2). Moreover, the reduction rule would not be applicable to R1 and R2, since the constants "*normal*" and "*dark*" in the heads of R1 and R2 cannot be unified together. In [19], a more powerful framework to model reduction was proposed, consisting of the use of 2 rules: generalised fusion and factorization.

---

the only remarkable distinction that two (fuzzy) constants A and B in Ω are only unified if $\forall x \in \Omega,\ A(x) = B(x)$ and if $\exists a \in \Omega, A(a) = 1$, then $\forall x \neq a,\ A(x) = 0$.

[6] Here Catalonia is taken as a region close to the border between France and Spain, encompassing some towns from both countries.

The generalized fusion rule G^π_* is defined as:

$$G^\pi_* : \frac{(p_0(t_1, t_2, ..., t_k) \leftarrow p_1 \wedge ... \wedge p_n, v_p)(p_0(s_1, s_2, ..., s_k) \leftarrow q_1 \wedge ... \wedge q_m, v_q)}{[(p_0(r_1, r_2, ..., r_k) \leftarrow p_1 \wedge ... \wedge p_n \wedge q_1 \wedge ... \wedge q_m, \max(v_p, v_q))]\theta}$$

where t_i and s_i and r_i are terms. Each term r_i is given by $r_i = t_i \cup s_i$, if both t_i and s_i are constants, or by $r_i = t_i\theta$, the application of the most general unifier (mgu) θ to t_i, otherwise.

The factorization rule is applied to predicates inside a single clause:

$$T^\pi_* : \frac{(p_0(...) \leftarrow p_1(t_1, ..., t_k) \wedge p_2(r_1, ..., r_k) \wedge p_3(...) \wedge ... \wedge p_n(...), v_p) \ ;}{[(p_0(...) \leftarrow p_1(t_1, ..., t_n) \wedge p_3(...) \wedge ... \wedge p_n(...), v_p)]\theta}$$

where the t_i's and r_i's are variables, $p_1(t_1, ..., t_n)$ is a variant of $p_2(r_1, ..., r_n)$, and θ is a mgu that unifies p_1 and p_2 and is such that it does not produce any inconsistency throughout the entire clause.

The application of the generalized fusion rule on R1 and R2 followed by the factorization rule on the literal *Beach* in the resultant clause produces

R12: $(Tan(x, normal \cup dark) \leftarrow Beach(x, y),$
$\max(\min(France(y), \alpha), \min(Spain(y), \beta))$

whose inclusion in the knowledge base allows us to obtain the answer "*normal or dark*" with a belief of at least $\min(\alpha, \beta)$ for Q2.

A theorem prover for PLFC-H was implemented in system KOMET [9]. This system has generalized annotated logics (GAL) [23,17] as underlying representation framework, and is duly equipped with GAL resolution and reduction rule. In order to make it possible to use KOMET as a theorem prover to PLFC-H, in [18] it was proposed to translate PLFC-H into GAL formalism and also to modify KOMET's resolution rule, such that the mgu θ that unifies the heads of two clauses be applied not only to the logical part of the reductant, as in the original definition, but also to its contraints. When the constraints do not involve fuzzy quantifiers (as in possibilistic logic PL), the results obtained by the original GAL reduction and the modified rules are the same. In order to allow the treatment of the example shown above, in [19] it was proposed to apply the generalized fusion and factorization rules to the knowledge base –with the consequent creation of new clauses–, prior to the use of KOMET as a theorem prover. In the example above, rule R12 would be first added to R1, R2 and R3, and only then queries Q1 and Q2 would be made, yielding the expected result.

5 PGL$^+$

In [1,2,3] an alternative Horn-rule style formalism to PLFC has been defined, allowing only disjunctive fuzzy constants. Within this restricted framework the aim was to fully define a logical system for reasoning under possibilistic uncertainty and disjunctive vague knowledge with an efficient and complete proof procedure oriented to goals.

To achieve this objective, first a possibilistic logic programming language of Horn rules with fuzzy propositional variables, called PGL, was defined and equipped with a complete modus ponens-style calculus [1]. Later, in [2,3], this language was extended to allow disjunctive fuzzy constants and the proof method was enlarged with a mechanism of semantical unification of fuzzy constants together with three other inference patterns. This extension, called PGL$^+$, was proved to be complete for atomic deduction when clauses fulfill two kinds of constraints.

The *basic components* of PGL$^+$ formulas are: a set of primitive propositions *Var*; *sorts* of constants; a set $\mathcal{C}$ of object *constants* (crisp and fuzzy constants), each having its sort; a set *Pred* of *unary* [7] *regular* predicates, each one having a type (a *type* is a tuple of sorts); 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 $\alpha \in [0, 1]$.

Fuzzy constants, like in PLFC, are seen as (flexible) restrictions on an existential quantifier. Moreover, it is natural to take the truth-value of, for instance, (*low*), under a given interpretation in which the salary is x_0 euros, as the degree in which the salary x_0 is considered to be *low*, i.e. $\mu_{low}(x_0)$. This leads to treat formulas as many-valued, with the unit interval $[0, 1]$ as set the of truth-values.

A many-valued *interpretation* for the language is a structure $w = (U, i, m)$ which maps each basic sort σ into a non-empty domain U_σ; a primitive proposition q into a value $i(q) \in [0, 1]$; a predicate p of type (σ) into a value $i(p) \in U_\sigma$; and an object constant A (crisp or fuzzy constant) of sort σ into a normalized fuzzy set $m(A)$ with membership function $\mu_{m(A)} : U_\sigma \rightarrow [0, 1]$. Remark that interpretations are *disjunctive* in the sense that, for each predicate symbol p, $i(p)$ is a unique value of the domain. Indeed, in contrast to PLFC, fuzzy constants in PGL$^+$ always express disjunctive fuzzy knowledge. The *truth value* of an atomic formula φ under an interpretation $w = (U, i, m)$, denoted by $w(\varphi)$, is just $i(q)$ if φ is a primitive proposition q, and it is computed as $\mu_{m(A)}(i(p))$ if φ is of the form $p(A)$. This truth value extends to rules by means of the min-conjunction and Gödel's many-valued implication:

$$w(p_1 \wedge \cdots \wedge p_k \rightarrow q) = \begin{cases} 1, & \text{if } \min(w(p_1), \ldots, w(p_k)) \leq w(q) \\ w(q), & \text{otherwise} \end{cases}$$

[7] We restrict ourselves to unary predicates for the sake of simplicity. However, since variables and function symbols are not allowed, the language still remains propositional.

As for the possibilistic semantics for PGL$^+$, as in PLFC, 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*. Recall that by $\Omega_{U,m}$ we denote the context of interpretations sharing a domain U and an interpretation of constants m and, once fixed the context, by $[\varphi]$ the fuzzy set of models for a formula φ defining $\mu_{[\varphi]}(w) = w(\varphi)$, for all $w \in \Omega_{U,m}$.

Now, in a fixed context $\Omega_{U,m}$, a belief state (or *possibilistic model*) is determined by a normalized possibility distribution on $\Omega_{U,m}$, $\pi : \mathcal{I}_{U,m} \to [0,1]$. 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 $N_2([\varphi] \mid \pi)$, is indeed at least α. Here, for the sake of soundness preservation, we take

$$N_2([\varphi] \mid \pi) = \inf_{w \in \Omega_{U,m}} \pi(w) \Rightarrow \mu_{[\varphi]}(w) \qquad (3)$$

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. [11]). Notice the difference of N_2 in (3) with respect the necessity measure N_1 adopted for PLFC in equation (2). For example, the formula

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

is to be interpreted in PGL$^+$ as

$(age\_Peter([about\_35]_\beta)$ is certain with necessity of at least $\min(0.9, 1 - \beta)$

for each $\beta \in [0,1]$. 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). 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 [2], it is proved that $\|\varphi\|_P = \inf\{N_2([\varphi] \mid \pi) \mid \pi \models P\}$.

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

Generalized resolution:

$$\frac{(p \wedge s \to q(A), \alpha),\ (q(B) \wedge t \to r, \beta)}{(p \wedge s \wedge t \to r, \min(\alpha, \beta))} \text{ [GR]}, \quad \text{if } A \subseteq B$$

Fusion:

$$\frac{(p(A) \wedge s \to q(D), \alpha),\ (p(B) \wedge t \to q(E), \beta)}{(p(A \cup B) \wedge s \wedge t \to q(D \cap E), \min(\alpha, \beta))} \text{ [FU]}$$

Intersection:

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

Resolving uncertainty:

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

Semantical unification:

$$\frac{(p(A),\alpha)}{(p(B),\min(\alpha, N_2(B \mid A)))}\ [\mathrm{SU}]$$

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

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

Finally, the notion of *proof* in PGL$^+$, denoted by $\vdash$, is deduction by means of the triviality axiom and the PGL$^+$ inference rules. Then, 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 [3] 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. This is indeed the first step of an efficient and complete proof procedure for PGL$^+$ programs satisfying what we call *context* constraint. A second step, based on the MP, SU, UN and IN rules, translates a PGL$^+$ program satisfying the modularity constraint into a semantically equivalent set of 1-weighted facts, whenever the program satisfied the context constraint. And, finally, 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.

6 Comparison and discussion

The main differences between PLFC and PGL$^+$ are (i) at the level of the syntax and semantics of the language; (ii) at the level of providing the language with a sound calculus; and (iii) at the level of defining an automated deduction method based on (ii).

Regarding the syntax, in PLFC, formulas are pairs of the form $(\varphi(\bar{x}), f(\bar{y}))$, where $\bar{x}$ and $\bar{y}$ denote sets of free and implicitly universally quantified variables and $\bar{y} \supseteq \bar{x}$, $\varphi(\bar{x})$ is a disjunction of literals with fuzzy constants, and $f(\bar{y})$ is a valid valuation function which expresses the certainty of $\varphi(\bar{x})$ in terms of necessity measures. Basically, valuation functions $f(\bar{y})$ are constant values and variable weights. In PGL$^+$, formulas are pairs of the form (φ, α), where φ is a first-order definite clause or a query with fuzzy constants and regular predicates and $\alpha \in [0, 1]$ is a lower bound on the belief (necessity) of φ.

On the other hand, in PLFC, fuzzy constants can be used in positive or negative literals, representing in both cases a flexible restriction on a existential quantifier over the literal. Therefore, as the proof method for PLFC is defined by refutation through a generalized resolution rule between positive and negative literals, unification (in the classical sense) of fuzzy constants is not allowed. However, as variable weights are interpreted as conjunctive (fuzzy) knowledge, an implicit semantical unification between fuzzy events can be performed between variable weights and fuzzy constants. In PGL$^+$, fuzzy constants are interpreted as disjunctive knowledge too but, in contrast to PLFC, there are no negative literals in the language; this allows to provide the language with a sound modus ponens-style calculus by derivation based on an explicit unification mechanism of fuzzy constants; furthermore, for a restricted class of clauses, completeness can be achieved by extending the system with a mechanism of fusion and an intersection between fuzzy constants.

Regarding the semantics, due to the presence of fuzzy constants, the truth evaluation of formulas is many-valued in both systems, and belief states are modeled by normalized possibility distributions on a set of many-valued interpretations, also in both systems. However, the basic connectives of PLFC are negation $\neg$ and disjunction $\vee$ while in PGL$^+$, they are conjunction $\wedge$ and implication $\rightarrow$, and the semantics for the two sets of connectives are not equivalent, i.e. the two sets of connectives are not inter-definable. Moreover, the generalized necessity measures used in PLFC and in PGL$^+$ for setting the possibilistic semantics of formulas (see expressions (2) and (3)) are different, although both are extensions of the classical necessity measure used in possibilistic logic (see expression (1)).

Regarding the unification mechanism between fuzzy constants, in both systems it consists of estimating to what degree the information expressed by a fuzzy constant follows from another piece of fuzzy information. This basically amounts to evaluate a (asymmetrical) matching between fuzzy events, and it is performed by computing a necessity measure. However, the necessity measure and the unification mechanism itself are different in each system. In PLFC unification is allowed between variable weights and fuzzy constants, and the degree of unification is implicitly computed during the resolution process. In PGL$^+$, unification can be performed between fuzzy constants and is explicitly handled by a separate inference rule. It is also worth noticing that the necessity measure N_1 used in PLFC has an interesting feature: $N_1(A \mid B) = 1$ iff the support of $B \subseteq$ the core of A; however, in general, $\frac{1}{2} \leq N_1(A \mid A) \leq 1$ and $N_1(A \mid A) = 1$ iff A is crisp. On the other hand, PGL$^+$ makes use of a different necessity measure N_2 which intuitively computes an inclusion degree. To be precise, $N_2(A \mid B) = 1$ iff $B \subseteq A$. However, this necessity measure has a side effect: if $\mu_B(u) = 1$ and $\mu_A(u) \neq 1$, then necessarily $N_2(A \mid B) = 0$.

A final remark regarding the unification mechanisms is that the unification degree between two different and precise constants is null in both systems. Sometimes this is a rather unpleasant behavior, specially if we are trying to model approximate knowledge. To remedy this particular situation, a possible solution is to extend (in both systems) each basic sort with a fuzzy similarity relation and to fuzzify precise constants by means of this similarity relation. This similarity-based unification approach is under development in [4] for PLFC and is somewhat related to those proposed by Arcelli, Formato, Gerla and Sessa [6,16,22] on the one hand, and the ones proposed by Vinař and Vojtáš [26,27] and Medina, Ojeda-Aciego and Vojtáš [20], in different frameworks.

Regarding automated deduction, a sound resolution-style refutation procedure for PLFC has been developed based on the computation of the necessity evaluation of fuzzy events. In order to get PLFC clauses with the greatest possible weights (i.e. to get higher unification degrees), a fusion mechanism must be applied after each resolution step. Therefore, the refutation procedure cannot be oriented to a resolvent clause and the search space consists of all possible orderings of the literals in the knowledge base. As already pointed out, in order to gain completeness, a fusion mechanism is also needed for PGL$^+$. Therefore, from a computational point of view, the extension of a knowledge base with all (explicit and hidden) clauses through a fusion mechanism is a drawback in both systems. However, this problem can be partially overcome in both the Horn fragment of PLFC (PLFC-H) and PGL$^+$ by performing a pre-processing step on the knowledge base.

In Table 1 we summarize the main differences between PGL$^+$ and PLFC-H, put into context by means of the following example. Consider the following fuzzy and uncertain statements:

s1 "The price of the book is about 34 euros with a certainty degree of 0.75".

s2 "If the price of a book is around 35 euros, buy the book".

Let $price\_book(\cdot)$ be a unary predicate of type (`price`), let buy_book be a propositional variable, and let $about\_34$ and $around\_35$ be two fuzzy object constants of sort `price`. These statements are differently represented in each system. In PLFC-H we get the following representation:

s1 $(price\_book(about\_34), 0.75)$

s2 $(buy\_book \leftarrow price\_book(x), around\_35(x))$

and in PGL$^+$:

s1 $(price\_book(about\_34), 0.75)$

s2 $(price\_book(around\_35) \rightarrow buy\_book, 1)$

	PLFC-H	PGL$^+$
Syntax	$(salary(x, low) \leftarrow age(x, y), young(y))$ $(speaks(Anna, x), \mu_{\{Spanish,English\}}(x))$ Disjunctive & conjunctive fuzzy constants	$(age(x, young) \rightarrow salary(x, low), 1)$ Disjunctive fuzzy constants
Interpr.	$w = (U, i, m)$ $i(p) \subseteq U$ (conjunctive inter.) $e(x) \in U$ (crisp evaluations) $w(p(A)) = \sup_{u \in i(p)} \mu_{m(A)}(u)$ $w(\neg q(B)) = \sup_{u \notin i(q)} \mu_{m(B)}(u)$ $w(\neg q(B) \vee p(A)) = \max(w(\neg q(B)), w(p(A)))$	$w = (U, i, m)$ $i(p) \in U$ (disjunctive inter.) $e(x) : U \rightarrow [0, 1]$ (fuzzy evaluations) $w(p(A)) = \mu_{m(A)}(i(p))$ $w(q(B) \rightarrow p(A))$ Gödel truth functions
Poss. Models	$\pi : \Omega_{U,m} \rightarrow [0, 1]$ $\pi \models (\varphi, \alpha)$ iff $N_1([\varphi] \mid \pi) \geq \alpha$ $N_1([\varphi] \mid \pi) = \inf_w \max(1 - \pi(w), w(\varphi))$	$\pi : \Omega_{U,m} \rightarrow [0, 1]$ $\pi \models (\varphi, \alpha)$ iff $N_2([\varphi] \mid \pi) \geq \alpha$ $N_2([\varphi] \mid \pi) = \inf_w \pi(w) \Rightarrow w(\varphi)$
Unifi.	Variable weights/Imprecise constants	Fuzzy constants/Fuzzy constants
Proof method	Refutation by resolution (implicit unification) + generalized fusion + factorization	Deduction by modus ponens + unification rule + fusion + intersection
S. & C.	Soundness (for PLFC)	Soundness and context constrained completeness

Table 1
Summary comparison of PLFC-H and PGL$^+$

Consider the following context:

- $U = \{U_{\texttt{price}} = [0, 1000](\text{euros})\}$;
- $m(about\_34) = [32; 34; 34; 36]$[8]; $m(around\_35) = [30; 34; 36; 40]$.

In PLFC-H, since *about_34* is interpreted as fuzzy set in the above context, **s1** should be rewritten as

s1' $(price\_book([about\_34]_{0.75}), 0.75)$

due to their semantical equivalence (see Sections 3 and 4). Now, we are interested in computing the certainty degree of buying the book in that particular context. To this end, in PLFC-H, we extend the knowledge base with the query

q $(\leftarrow buy\_book, 1)$

and we search a proof for $(\perp, \alpha)$ by applying the resolution rule. The resolution of **s1'** and **s2** consists of computing the certainty degree of the price being $m(around\_35)$ euros given the price is in the crisp interval $[m(about\_34)]_{0.75} = [33.5, 34.5]$, and thus, $N_1(m(around\_35) \mid [33.5, 34.5]) =$

[8] 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.

$\inf_{u\in[33.5,34.5]} \mu_{[30;34;36;40]}(u) = 0.87$ and we infer

s3 $(buy\_book, \min(0.75, 0.87))$.

Now, resolving this clause with the query **q** we infer $(\perp, 0.75)$, and thus, the necessity degree of buying the book is 0.75.

On the other hand, in PGL$^+$, as $m(about\_34) \subseteq m(around\_35)$, we get $N_2(m(around\_35) \mid m(about\_34)) = 1$. Then, applying the unification rule to **s1** we derive

s4 $(price\_book(around\_35), \min(0.75, 1))$

and applying the generalized modus ponens rule to **s2** and **s4** we get again **s3**, i.e. $(buy\_book, 0.75)$.

Hence, in the above particular context, both systems provide the same result. However, if we consider a different context U' with for instance $m(around\_35) = [33, 37]$, we get $N_1(m(around\_35) \mid [m(about\_34)]_{0.75}) = \inf_{u\in[33.5,34.5]} \mu_{[33,37]}(u) = 1$ and $N_2(m(around\_35) \mid m(about\_34)) = \inf_{u\in[0,1000]} \mu_{[32;34;34;36]}(u) \Rightarrow \mu_{[33,37]}(u) = 0.5$, and thus, in this particular context, the necessity degree of buying the book is 0.75 in PLFC-H and 0.5 in PGL$^+$.

References

[1] T. Alsinet and L. Godo. A complete calculus for possibilistic logic programming with fuzzy propositional variables. In *Proc. of UAI'2000 Conference*, pp. 1–10, Stanford, CA, 2000.

[2] T. Alsinet and L. Godo. A complete proof method for possibilistic logic programming with semantical unification of fuzzy constants. In *Proc. of ESTYLF'2000 Conference*, pp. 279–284, Sevilla, Spain, 2000.

[3] T. Alsinet and L. Godo. A proof procedure for Possibilistic logic programming with fuzzy constants. In *Proc. of ECSQARU'2001*, Toulouse (France), Sept. 2001, LNAI 2143, Springer, pp. 760–771.

[4] T. Alsinet and L. Godo. Adding similarity-based reasoning capabilities to a Horn fragment of Possbilistic Logic with Fuzzy Constants. Submitted.

[5] T. Alsinet, L. Godo, and S. Sandri. On the semantics and automated deduction for PLFC, a logic of possibilistic uncertainty and fuzziness. In *Proc. of UAI'99 Conference*, pp. 3–12, Stockholm, Sweden, 1999.

[6] F. Arcelli, F. Formato, and G. Gerla. Fuzzy unification as a foundation of fuzzy logic programming. In F. Arcelli and T.P. Martin, editors, *Logic Programming and Soft Computing*, chapter 3, pp. 51–68. Research Studies Press, 1998.

[7] J.F. Baldwin, T.P. Martin, and B.W. Pilsworth. *Fril – Fuzzy and Evidential Reasoning in Artificial Intelligence*. Research Studies Press, 1995.

[8] G. Bel, D. Farreny, and H. Prade. Towards the use of fuzzy rule-based systems in the monitoring of manufactoring systems. In J.F. Mc Waters J.P. Crestin, editor, *Software for Discret Manufacturing*, pp. 525–535. North-Holland, 1986.

[9] J. Calmet, S. Jekutsch, P. Kullmann, and S. Schü. KOMET - A system for the integration of heterogeneous information sources, In Z.W. Raś, and A. Skowron, eds, *Foundations of Intelligent Systems*, *Lecture Notes in Artificial Intelligence 1325*, Springer Verlag, 1997, pp. 318–327.

[10] M. Cayrol, H. Farreny, and H. Prade. Fuzzy pattern matching. *Kybernetes*, 11:103–116, 1982.

[11] D. Dubois, J. Lang, and H. Prade. Possibilistic logic. In D.M. Gabbay, C.J. Hogger and J.A. Robinson, editor, *Handbook of Logic in Artificial Intelligence and Logic Programming*, pp. 439–513. Oxford University Press, 1994.

[12] D. Dubois, H. Prade, and S. Sandri. Possibilistic logic with fuzzy constants and fuzzily restricted quantifiers. In F. Arcelli and T.P. Martin, editors, *Logic Programming and Soft Computing*, pp. 69–90. Research Studies Press, 1998.

[13] L. Rios-Filho, S. Sandri. Contextual fuzzy unification. In *Proc. IFSA'95* São Paulo (Brazil), pp. 81-84, 1995.

[14] G. Gerla and M.I. Sessa. Similarity in Logic Programming. In G. Chen, M. Ying, and K. Cai, editors, *Fuzzy Logic and Soft Computing*, chapter 2, pp. 19–31. Kluwer, 1999.

[15] L. Godo and L. Vila. Possibilistic Temporal Reasoning based on Fuzzy Temporal Constraints. In *Proceedings IJCAI'95*, Montreal (Canada), Vol. 2, pp. 1916-1922, 1995.

[16] F. Formato, G. Gerla, and M. Sessa. Similarity-based unification. *Fundamenta Informaticae*, 40:1–22, 2000.

[17] M. Kifer and V.S. Subrahmanian. Theory of Generalized Annotated Logic Programming, *Journal of Logic Programming*, Vol. 12(1), 1999, pp. 335–367.

[18] P. Kullmann, and S. Sandri, Possibilistic logic as an annotated logic. In *Proc. Fuzz-IEEE'99*, Seoul (S. Korea), 1999, pp. 210–215.

[19] P. Kullmann, and S. Sandri. Implementation of an Extended Possibilistic Logic in an Annotated Logic Theorem Prover. In *Proc. IFSA-NAFIPS'2001*, Vancouver (Ca), Jul. 2001, pp. 1529–1534.

[20] J. Medina, M. Ojeda-Aciego, and P. Vojtás. Similarity-based unification: a multi-adjoint approach. In *Proc. EUSFLAT Conference in Fuzzy Logic and Technology*, pp. 273-276, 2001.

[21] S. Sandri, and L. Godo. Treatment of temporal information in possibilistic logic with fuzzy constants. In *Proc. IFSA'99*, Taipei (ROC), pp. 561-565, 1999.

[22] M.I. Sessa. Approximate reasoning by similarity-based SLD resolution. *Theoretical Computer Science*, 275(1-2):389–426, 2002.

[23] V.S. Subrahmanian. On the Semantics of Quantitative Logic Programs. *Proc. 4th IEEE Symposium on Logic Programming*, Computer Society Press, Washington DC, pp. 173–182, 1987.

[24] M. Umano. Fuzzy set prolog. In *Proceedings 2nd IFSA Congress*, pp. 750–753, Tokyo, Japan, 1987.

[25] H.E. Virtanen. Linguistic logic programming. In F. Arcelli and T.P. Martin, editors, *Logic Programming and Soft Computing*, pp. 91–128. Research Studies Press, 1998.

[26] J. Vinař and P. Vojtáš. A formal model for fuzzy knowledge based systems with similarities. *Neural Network World*, 10(5):891–905, 2000.

[27] P. Vojtáš. Fuzzy logic programming *Fuzzy Sets and Systems*, 124(3):361–370, 2001.

[28] P. Vojtás, T. Alsinet, and L. Godo. Different models of fuzzy logic programming with fuzzy unification (towards a revision of fuzzy databases). In *Proc. IFSA-NAFIPS'2001*, July 2001, Vancouver (Canada) pp. 1541-1545.