



HAL
open science

Possibilistic logic with fuzzy constants and fuzzily restricted quantifiers

Didier Dubois, Henri Prade, Sandra Sandri

► **To cite this version:**

Didier Dubois, Henri Prade, Sandra Sandri. Possibilistic logic with fuzzy constants and fuzzily restricted quantifiers. Martin, Trevor P.; Arcelli-Fontana, Francesca. Logic Programming and Soft Computing, Research Studies Press, pp.69–90, 1998, 978-0863802256. <hal-04028794>

HAL Id: hal-04028794

<https://hal.science/hal-04028794v1>

Submitted on 17 Mar 2023

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire HAL, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



HAL Authorization

Possibilistic Logic with Fuzzy Constants and Fuzzily Restricted Quantifiers[†]

Didier Dubois*, Henri Prade* and Sandra A. Sandri**

* IRIT, Université Paul Sabatier - CNRS, 118 route de Narbonne, 31062 Toulouse
Cedex 4, France, Email: {dubois, prade}@irit.fr

** INPE 1758 Av. dos Astronautas, C.P. 515, 12225 São José dos Campos SP,
Brasil, Email: sandri@lac.inpe.br

ABSTRACT

In standard possibilistic logic, a formula is an ordered pair made of a classical first order logic formula and a weight. The weight associated with this formula expresses its level of certainty and is formally interpreted in terms of a necessity measure. This weight, which takes its value in a linearly ordered scale, may itself depend on variables involved in the classical logic formula. Letting the weight be a function of variable(s) involved in the formula, makes the certainty of the formula depend on some proviso expressed in terms of regular, or fuzzy, predicate(s) whose characteristic functions appear in the expression of the weight. This enables us to attach a fuzzily restricted, universal quantifier to a classical formula. Proceeding one step further, but still using regular predicates in the first component of the possibilistic logic formula, variables are allowed to be instantiated with fuzzy values representing imprecise, unsharp information, i.e., fuzzy constants are allowed. This paper presents basic inference patterns involving fuzzily restricted quantifiers and fuzzy constants. The handling of fuzzy constants and fuzzily restricted quantifiers raises algorithmic issues for the unification procedure which are discussed in the last section of the paper. This article should be seen as a preliminary piece of work towards a tentative extension of possibilistic logic, with also some concerns about programming aspects.

[†] This paper is a fully revised version of a paper entitled "Possibilistic logic augmented with fuzzy unification" which appeared in Volume II of the Proceedings of the Sixth International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems (IPMU'96), Granada, Spain, July 1-5, 1996, pp. 1009-1014.

1. INTRODUCTION

In possibilistic logic (Dubois, Lang and Prade, 1994a, b), propositional or first order logic formulas are associated with their level of certainty. The formulas themselves remain classical formulas, although the certainty weight attached to a formula may involve logical variables and even fuzzy predicates, in case the certainty of the formula depends on the level of satisfaction of gradual properties (as it is recalled in the next section). Although some proposals have been made for handling fuzzy predicates in logical formulas in possibilistic logic (Dubois and Prade, 1990), or in different other frameworks (e.g., Mukaidono, Shen and Ding, 1989; Novák, 1990), we are investigating another direction in this paper for representing fuzzy information in a logical setting. Namely, we are allowing for fuzzy set constants in the language. Indeed there are several ways of encoding fuzzy information. For instance, consider the piece of information "John is young"; it can be written as $age(John, young)$ or as $young(John)$. In the first expression 'young' is a fuzzy constant and 'age' a regular predicate; in the second one, 'young' is a fuzzy predicate. It formally corresponds to the equivalence $(x,y) \in R \Leftrightarrow x \in R^{-1}(y)$ where R is a relation, letting y being fuzzy.

Apparently, there have been quite a few attempts for introducing "fuzzy constants" in a knowledge representation language involving variables. It was suggested in the early eighties by Cayrol, Farreny and Prade (1982) and Bel et al. (1986) when discussing the introduction of fuzzy values in a pattern matching procedure, in relation with the evaluation of rules whose condition parts involve variables and control symbols. More recently, Virtanen (1994), Rios-Filho and Sandri (1995) have considered algorithmic aspects of fuzzy unification, mainly in a fuzzy logic programming perspective. See also Fehrer (1990) for a treatment of imprecisely defined predicates.

The general idea underlying possibilistic logic, and thus the approach presented in the following, is to stay as close as possible to the framework of classical logic, while trying to slightly increase the expressivity of the language in order to be able to cope with uncertainty, gradual properties, fuzzy constants and so on.

The paper is organized in the following way. Section 2 first provides the necessary background about possibilistic logic. We take advantage of the format of possibilistic logic expressions, as pairs made of a classical formula and a weight, for modeling fuzzy restrictions on universal quantifiers in the second part of Section 2. Then, the representation of imprecise or fuzzy constants is discussed in Section 3, where patterns of inferences involving fuzzily restricted quantifiers and fuzzy constants are then established. Section 4 deals with algorithmic issues raised by unification, in this fuzzy setting.

2. POSSIBILISTIC LOGIC

2.1. Background

In possibilistic logic (Dubois, Lang and Prade, 1994a, b), classical first order logic formulas p are weighted in terms of lower bounds $\alpha \in (0,1]$ of necessity measures, i.e., the possibilistic logic expression (p,α) is understood as $N(p) \geq \alpha$, where N is

a necessity measure. A necessity measure N is a function from the set of logical formulas to a totally ordered bounded scale, which is characterized by the axioms

- i) $N(\top) = 1$,
- ii) $N(\perp) = 0$ where \top and \perp stand for tautology and contradiction respectively,

and 0 and 1 are the bottom and the top element of the scale,

- iii) $N(p \wedge q) = \min(N(p), N(q))$.

We use the real interval $[0,1]$ as the range of necessity measures in the following, but this is not compulsory. A, finite or not, totally ordered scale bounded by a bottom and a top element is enough. A possibility measure Π is associated by duality with N , namely

$$\Pi(p) = 1 - N(\neg p),$$

where $1 - (\cdot)$ is the order-reversing map of the scale. It expresses that the absence of certainty in favor of $\neg p$ makes p possible.

The following conventions are followed for the notations. Lower case letters $a, b, c, \dots, p, q, r, \dots$ denote formulas. Logical variables are denoted by x, y, z and appear between parentheses after the formula if these variables are universally quantified in the formula (free variables). Constants are denoted by letters s, t, u . (Fuzzy) sets of constant values or extensions of predicates are denoted by capital letters A, B, C, \dots, P, Q, R . Lastly, weights are denoted by Greek letters $\alpha, \beta, \dots, \varphi, \psi, \dots, \mu$ (which may be indexed); variables are indicated between parentheses after the weight if it depends on these variables.

The min-decomposability of necessity measures allow us to work with weighted *clauses* without lack of generality, since $N(\hat{A}_{i=1,n} p_i) \geq \alpha \Leftrightarrow \forall i, N(p_i) \geq \alpha$, i.e., $(\hat{A}_{i=1,n} p_i, \alpha) \Leftrightarrow \hat{A}_{i=1,n} (p_i, \alpha)$. The basic inference rules in possibilistic logic are the following ones:

- $(\neg p \vee q, \alpha); (p \vee r, \beta) \vdash (q \vee r, \min(\alpha, \beta))$ (resolution rule)
- $(\forall x, p(x), \alpha) \vdash (p(s), \alpha)$ (particularization)
- for $\beta \leq \alpha$ $(p, \alpha) \vdash (p, \beta)$ (weight weakening)
- if p entails q classically, $(p, \alpha) \vdash (q, \alpha)$ (formula weakening)
- $(p, \alpha); (p, \beta) \vdash (p, \max(\alpha, \beta))$ (fusion)

where \vdash denotes the syntactic inference in possibilistic logic. Classical resolution is retrieved when all the weights are equal to 1. Note that formulas of the form $(p, 0)$ which do not contain any information ($\forall p, N(p) \geq 0$ always holds) are never written.

Refutation can be easily extended to possibilistic logic. Let K be a knowledge base made of possibilistic formulas. Proving (p, α) from K amounts to adding $(\neg p, 1)$, put in clausal form, to K , and using the above rules repeatedly to showing that $K \cup \{(\neg p, 1)\} \vdash (\perp, \alpha)$. Clearly, we are interested here in getting the empty clause with the greatest possible weight. Semantic aspects of possibilistic logic,

including soundness and completeness results with respect to the above syntactic inference machinery, are presented in (Dubois, Lang and Prade, 1994a, b). Possibilistic logic semantics is based on the association of the possibilistic knowledge base $K = \{(p_i, \alpha_i)\}_{i=1, n}$ with the possibility distribution π_K representing the fuzzy set of models of K :

$$\pi_K(v) = \min_{i=1, n} \max(\mu_{[p_i]}(v), 1 - \alpha_i)$$

(where $[p_i]$ denotes the sets of models of p_i). A necessity measure N_K is associated with π_K . Namely

$$N_K(p) = \min_{v \in [\neg p]} (1 - \pi_K(v)).$$

It can be shown that π_K is the largest possibility distribution such that $N_K(p_i) \geq \alpha_i$, $\forall i = 1, n$, i.e., the possibility distribution which allocates the greatest possible possibility degree to each interpretation in agreement with the constraints induced by K .

2.2. Variable Weights

Clearly, this framework allows for the modeling of uncertain information in the setting of possibility theory (Dubois and Prade, 1988). If we want to accommodate fuzzy predicates, one way is to allow for fuzzy predicates in the language and to have extended resolution rules for them (Dubois and Prade, 1990). Another route, recalled in this subsection, is to keep classical predicates, but to allow for *variable weights*, as already suggested in (Dubois et al., 1994a, c). Namely, the lower bounds can depend on variables involved in the formula. For instance, the possibilistic logic formula

$$(p(x), \mu_A(x))$$

where x is a free variable, means that for any x , we are certain that $p(x)$ is true with a necessity degree at least equal to $\mu_A(x)$. If μ_A is the characteristic function of a vague predicate a , it enables us to express that the more x satisfies a , the more certain the clause $p(x)$ is, as in the sentence "the younger the person, the more certain he/she is single" (here $\mu_A = \mu_{\text{young}}$), represented by the possibilistic formula $(\text{single}(x), \mu_{\text{young}}(x))$.

Besides, it can be shown (Dubois, Lang and Prade, 1994a, b) that the possibilistic formula

$$(\neg p(x) \vee q(x), \alpha)$$

is semantically equivalent to

$$(q(x), \min(\mu_p(x), \alpha)),$$

where $\mu_{p(x)} = 1$ if $p(x)$ is true and $\mu_{p(x)} = 0$ if $p(x)$ is false. Thus, it expresses here that $q(x)$ is α -certainly true given the proviso that $p(x)$ is true.

It is sometimes interesting to restrict universal quantifiers. This can be made in a fuzzy way, if we are unsure about the domain of applicability of the logical formula. Indeed the logical expression $\forall x \in A p(x)$ can be encoded by the clause $\neg a(x) \vee p(x)$ (where A is an ordinary set which is the extension of predicate a), or equivalently as the possibilistic expression $(p(x), \mu_A(x))$ (which is equivalent to $(\neg a(x) \vee p(x), 1)$). Then letting A be fuzzy, we indeed express that $p(x)$ is true is all the more certain as $\mu_A(x)$ is high. It expresses a conjunctive information which holds for all elements which are somewhat A , namely, it amounts to the set of possibilistic formulas $(p(s), \mu_A(s))$ for all s such that $\mu_A(s) > 0$. Thus, $(p(x), \mu_A(x))$ can be viewed as a formula quantified by a universal quantifier which is fuzzily restricted by μ_A .

The instantiation of a variable certainty weight will be determined by the instantiation of the involved variable, e.g.,

from $(\neg p(x) \vee q(x), \varphi(x))$ and from $(p(s), \alpha)$

we infer

$(q(s), \min(\alpha, \varphi(s)))$.

Then, particularization is performed both on the formula-part and on the weight-part of the possibilistic expression, as in the following example (where the 'age' function is italicized for avoiding confusion with the 'age' predicate)

$$\begin{array}{l} (\neg \text{likes}(x,z) \vee \text{likes}(y,z), \mu_{\text{approx_equal}}(\text{age}(x), \text{age}(y))) \\ (\text{likes}(\text{John}, \text{watching TV}), 1) \\ \text{age}(\text{Tom}) = 12, \text{age}(\text{John}) = 13 \\ \hline (\text{likes}(\text{Tom}, \text{watching TV}), \mu_{\text{approx_equal}}(12,13)). \end{array}$$

Involved variables can be also eliminated in this process. For instance, from

$(\neg p(x) \vee q(y), \varphi(x,y))$ and $(p(x), \psi(x))$

we can infer $(q(y), \min(\varphi(x,y), \psi(x)))$ applying the resolution rule for every value of x ; hence we infer the uncertain clause

$(q(y), \max_{x \in X} \min(\varphi(x,y), \psi(x)))$,

where X is the supposedly finite domain of variable x , by application of the fusion rule. For instance, from

$(\neg \text{single}(x) \vee \text{single}(y), \min(\alpha, \mu_{\text{approx_equal}}(\text{age}(x), \text{age}(y))))$

and from

$$(\text{single}(x), \mu_{\text{young}}(\text{age}(x)))$$

we get

$$(\text{single}(y), \min(\alpha, \max_{x \in X} \min(\mu_{\text{approx_equal}}(\text{age}(x), \text{age}(y)), \mu_{\text{young}}(\text{age}(x))))))$$

which can be turned into

$$(\text{single}(y), \min(\alpha, \max_{z \in Z} \min(\mu_{\text{approx_equal}}(z, \text{age}(y)), \mu_{\text{young}}(z))))$$

where $Z = \text{age}(X) = \bigcup_{x \in X} \{\text{age}(x)\}$.

This can be seen as the application of the max-min compositional rule of inference of fuzzy reasoning (Zadeh, 1979) performed inside the manipulation step of the weights, where φ ($\mu_{\text{approx_equal}}$ in the example) and ψ (μ_{young} in the example) are viewed as the membership functions of a fuzzy relation and a fuzzy set respectively. However, the membership degrees are used here as lower bounds of necessity degrees, which does not correspond to the usual view of fuzzy reasoning where they are generally understood as possibility degrees.

3. FUZZY CONSTANTS

In the above examples, constants are precise values as usual. Allowing for imprecise or even fuzzy constants representing ill-known values can enlarge the knowledge representation power of the possibilistic logic language, including the use of fuzzy functions as in the example

$$(\text{age}(x,y) \wedge \text{father}(z,x) \rightarrow \text{age}(z, y \oplus \widetilde{20-35}), .9)$$

where $\widetilde{20-35}$ stands for a fuzzy quantity expressing "about 20 to 35", and \oplus denotes the addition extended to fuzzy numbers.

We first study the problem raised by the representation of *imprecise* constants B corresponding to expressions of the form $\exists x \in B p(x)$, whose possible values for x are restricted by a classical *crisp* subset B, before studying the general case where B is a *fuzzy* set. Then, the inference from possibilistic formulas involving imprecise or fuzzy constants and fuzzily restricted quantifiers, is discussed in detail.

3.1. Imprecise and Fuzzy Constants

The expression $\exists x \in B p(x)$ clearly expresses a disjunctive information about the elements in B. Thus an imprecise constant $B = \{t_1, \dots, t_n\}$ applied to a predicate p is equivalent to the disjunction $p(t_1) \vee \dots \vee p(t_n)$. This will be symbolically written $p(B)$. In case we have to represent the fact that only one value of x is such

that $p(x)$ holds ($\exists!x \in B p(x)$), the t_i 's are mutually exclusive, i.e., $\neg p(t_i) \vee \neg p(t_j)$ $\forall i \neq j$ has to be added in the knowledge base. This later situation will be studied in Section 3.4.

Let B be a fuzzy set. B can be described by its β -cuts $B_\beta = \{t, \mu_B(t) \geq \beta\}$ with $\beta > 0$. The certainty that B_β holds, i.e., that $p(B_\beta)$ is satisfied by the imprecise constant B_β , is lower bounded by $1 - \beta$. Indeed in possibility theory, it can be checked that the necessity measure that B_β is true, knowing the fuzzy information B , is such that $N(B_\beta) = \inf_{t \notin B_\beta} 1 - \mu_B(t) \geq 1 - \beta$. Thus, $p(B)$ when B is fuzzy is equivalent to the set of possibilistic formulas $(p(B_\beta), N(B_\beta))$ where B_β is an imprecise constant. This set is finite if and only if the number of distinct possibility levels β which are necessary for describing B is finite. Assume that these levels are ordered in the following way $\beta_1 = 1 > \beta_2 > \dots > \beta_k > \beta_{k+1} = 0$; then $N(B_{\beta_i}) = 1 - \beta_{i+1}$, for $i = 1, k$. In particular, $N(B_{\beta_k}) = 1$, which expresses that we are certain that the values restricted by B are in the support of B , i.e., $\{t, \mu_B(t) > 0\} = B_{\beta_k}$. In case the number of distinct β -cuts is infinite (which requires that the universe on which the fuzzy set B is defined is also infinite), and the membership function μ_B is continuous, we would have $N(B_\beta) = 1 - \beta$.

It is easy to see that $p(B_1 \cup B_2) = p(B_1) \vee p(B_2)$ in the crisp and in the fuzzy cases (since $(B_1 \cup B_2)_\beta = B_{1\beta} \cup B_{2\beta}$).

Note that we have to be cautious with the understanding of notation $p(B)$ (which means $\exists x \in B p(x)$) when p involves a negation. Thus if $p \equiv \neg r$, $\neg r(B)$ means $\exists x \in B \neg r(x)$, which clearly differs from $\neg[\exists x \in B r(x)] \equiv \forall x \in B \neg r(x)$. Both expressions $\neg r(B)$ and $\neg[\exists x \in B r(x)]$ coincide only for ordinary constants $B = \{t\}$.

Due to the equivalence $\neg[\exists x \in B r(x)] \equiv \forall x \in B \neg r(x)$, when an imprecise constant appears in a negated formula it is always possible to change it into a possibilistic formula with a fuzzily restricted quantifier. Namely, for instance

$$(\neg[\exists y \in A p(x,y)] \vee q, \alpha)$$

is semantically equivalent to

$$(\neg p(x,y) \vee q, \min(\alpha, \mu_A(y))).$$

This can be easily seen when A is non-fuzzy and finite: $A = \{s_1, \dots, s_n\}$. Then the clause is equivalent to the family of clauses $(\neg p(x, s_i) \vee q, \alpha)$ for $i=1, n$, i.e., $(\neg p(x,y) \vee q, \min(\alpha, \mu_{\{s_i\}}(y)))$ and applying the fusion rule we get $(\neg p(x,y) \vee q, \max_i \min(\alpha, \mu_{\{s_i\}}(y)))$ which can still be written $(\neg p(x,y) \vee q, \min(\alpha, \max_i \mu_{\{s_i\}}(y)))$ where A is decomposed into the union of its singletons. For instance, $(\neg[\exists y \in \text{France lives_in}(x,y)] \vee \text{speaks}(x, \text{French}), \alpha)$ is equivalent to $(\neg \text{lives_in}(x,y) \vee \text{speaks}(x, \text{French}), \min(\alpha, \mu_{\text{France}}(y)))$. This can be generalized when A is fuzzy, using weighted singletons.

Lastly, note that the proposed framework does not completely allow for a proper handling of conjunction inside the scope of an existential quantifier (i.e., an imprecise constant). Namely, since $\exists x \in A p(x) \wedge q(x)$ only entails (but is not equivalent to) $\exists x \in A p(x) \wedge \exists x \in A q(x)$, the representation of the first formula cannot be handled in our setting since we can only express $(p(A), 1)$ and $(q(A), 1)$ which is the exact counterpart of the second formula. Indeed here A does not play the role of a Skolem constant, and the fact that two ill-located constants have the same imprecise restriction A does not guarantee their identity at all, when A is not a singleton. A proper handling would require some "coloring" technique for expressing, when necessary, that the imprecise instantiation of a variable should be the same in several places (see Lang et al. (1991) on this point). For the same kind of reason we cannot represent $\exists y \in B \forall x \in A p(x,y)$, but only $\forall x \in A \exists y \in B p(x,y)$, as $(p(x,B), \mu_A(x))$. This also prevents us to resolve $p(A)$ with $\neg p(A)$, except if A reduces to a singleton.

3.2. Inference with Imprecise Constants and Fuzzily Restricted Quantifiers

Let us consider a pattern of reasoning made of two premises, one involving a quantifier restricted by A , one including an imprecise constant B (together with a standard universal quantifier $\forall y$ and a regular constant s_0)

$$\left\{ \begin{array}{l} \forall x \in A \forall y \neg p(x,y) \vee q(x,y) \\ \exists x \in B p(x,s_0). \end{array} \right. \quad (\text{I})$$

From these two premises, we can conclude that

$$\exists x \in B q(x,s_0) \text{ provided that } B \subseteq A.$$

This can be readily checked, encoding A and B as predicates a and b respectively, i.e., rewriting (I) as below and translating the condition $B \subseteq A$

$$\left\{ \begin{array}{l} \forall x \forall y \neg a(x) \vee \neg p(x,y) \vee q(x,y) \\ \forall x \neg b(x) \vee a(x) \\ p(\hat{x},s_0); b(\hat{x}) \end{array} \right. \quad (\text{II})$$

where \hat{x} is a Skolem constant. Then, from (II), we obviously deduce that $q(\hat{x},s_0)$ holds as well as $b(\hat{x})$.

However, we may think of B as expressing an imprecise constant in (I). Indeed by imprecise constant we mean that we rewrite $\exists x \in B, p(x,s_0)$ as $p(B,s_0)$, where $p(B,s_0)$ expresses that predicate p holds true for the constant s_0 and for an ill-known x restricted by B . Let $B = \{t_1, \dots, t_n\}$; $(p(B,s_0),1)$ is thus equivalent to $(p(t_1,s_0) \vee \dots \vee p(t_n,s_0), 1)$. Let us suppose that B is contained in A ; thus, without lack of generality, we can suppose $A = \{t_1, \dots, t_n, t_{n+1}, \dots, t_m\}$, which makes $(\forall x \in A \forall y \neg p(x,y) \vee q(x,y), 1)$ equivalent to the conjunction of m clauses

$(\neg p(t_1, y) \vee q(t_1, y), 1) \dots$ and $(\neg p(t_n, y) \vee q(t_n, y), 1)$ and $(\neg p(t_{n+1}, y) \vee q(t_{n+1}, y), 1) \dots$ and $(\neg p(t_m, y) \vee q(t_m, y), 1)$. Therefore, if B is contained in A , using the resolution rule n times, we obtain $(q(t_1, s_0) \vee \dots \vee q(t_n, s_0), 1)$ which is equivalent to $(\exists x \in B q(x, s_0), 1)$. Written in possibilistic logic style, pattern (I) thus reads

$$\frac{(\neg p(x,y) \vee q(x,y), \mu_A(x))}{(p(B,s_0), 1)} \quad \text{-----} \quad \text{(III)}$$

$$(q(B,s_0), \min_{x \in B} \mu_A(x))$$

where $\min_{x \in B} \mu_A(x) = 1$ if and only if $B \subseteq A$ and is zero otherwise. This corresponds to a rule empirically stated in (Rios-Filho and Sandri, 1995). A noticeable particular case of III is when A is the full range of x (no restriction), where (III) simplifies into

$$\frac{(\neg p(x,y) \vee q(x,y), 1)}{(p(B,s_0), 1)} \quad \text{-----}$$

$$(q(B,s_0), 1).$$

This can be generalized with weights different from 1.

Pattern (III) can thus be used in the framework of possibilistic logic, as in the example,

$$\frac{(\neg \text{lives_in}(x,y) \vee \text{speaks}(x, \text{French}), \min(\alpha, \mu_{\text{France}}(y)))}{(\text{lives_in}(\text{Jean}, \{\text{Paris}, \text{Orléans}\}), 1)} \quad \text{-----}$$

$$(\text{speaks}(\text{Jean}, \text{French}), \alpha)$$

where 'France' is described by the set of towns it includes (France = {Orléans, Paris, Toulouse, ...}).

Pattern (III) can be readily extended to the case of a fuzzily restricted quantifier, where μ_A is the membership function of a fuzzy set A . Indeed using $B = \{t_1, \dots, t_n\}$, $(p(B,s_0), 1)$ is equivalent to $(p(t_1,s_0) \vee \dots \vee p(t_n,s_0), 1)$. Then, from $(\neg p(x,y) \vee q(x,y), \mu_A(x))$ applied to $x = t_1, \dots, x = t_n$ and $(p(t_1,s_0) \vee \dots \vee p(t_n,s_0), 1)$, it can be easily checked that we can derive

$$(q(t_1,s_0) \vee \dots \vee q(t_n,s_0), \min(\mu_A(t_1), \dots, \mu_A(t_n))).$$

This is precisely pattern (III) when A is fuzzy and B is non-fuzzy.

3.3. Inference with Fuzzy Constants and Fuzzily Restricted Quantifiers

Let us first consider a pattern of reasoning with a restricted, but *non-fuzzy*, quantifier (represented by A), and a fuzzy constant B having a finite number k of β -cuts, such that $\beta_1 = 1 > \beta_2 > \dots > \beta_k > \beta_{k+1} = 0$. In possibilistic terms, it reads

$$\left\{ \begin{array}{l} (\neg p(x,y) \vee q(x,y), \mu_A(x)) \\ (p(B_{\beta_i}, s_0), 1 - \beta_{i+1}), \text{ for } i=1, k. \end{array} \right. \quad \text{(IV)}$$

Thus, we can apply pattern III, looking for β_i such that $B_{\beta_i} \subseteq A$ with $1 - \beta_{i+1}$ as large as possible (since we are interested in the most certain conclusions). Thus, from (IV), we can derive

$$(q(B, s_0), \max_{i=1, k} \min(1 - \beta_{i+1}, \text{Inc}(B_{\beta_i}, A))),$$

where $\text{Inc}(B_{\beta_i}, A) = 1$ if $B_{\beta_i} \subseteq A$ and is 0 otherwise. It can be shown in the infinite case, with continuous membership functions (e.g., Prade, 1982) that

$$\sup_{\beta} \min(1 - \beta, \text{Inc}(B_{\beta}, A)) = 1 - \inf(\{\beta \text{ s.t. } B_{\beta} \subseteq A\}) = \inf_{t \notin A} 1 - \mu_B(t)$$

where $\inf(E)$ denotes the infimum of the values in the set E . When B has a finite number of level cuts, we also have

$$\max_i \min(1 - \beta_{i+1}, \text{Inc}(B_{\beta_i}, A)) = 1 - \min_i (\{\beta_{i+1} \text{ s.t. } B_{\beta_i} \subseteq A\}) = \min_{t \notin A} 1 - \mu_B(t).$$

Thus from (IV), we can derive

$$(q(B, s_0), \min_{t \notin A} 1 - \mu_B(t)).$$

The general case, where both A and B are fuzzy, can be handled in terms of level-cuts as well. Indeed we have the possibilistic clauses $(\neg p(x, y) \vee q(x, y), \alpha)$ for all $x \in A_{\alpha}$, and $\exists x \in B_{\beta_i}, (p(x, s_0), 1 - \beta_{i+1})$. We are then trying to maximize $\min(\alpha, 1 - \beta_{i+1})$ under the constraint $B_{\beta_i} \subseteq A_{\alpha}$. It has been proved in the infinite case, for continuous membership functions that (Prade, 1982)

$$\sup_{\alpha, \beta} \min(\alpha, 1 - \beta, \text{Inc}(B_{\beta}, A_{\alpha})) = \inf_t \max(\mu_A(t), 1 - \mu_B(t)).$$

In the finite case, we still have (see Dubois et al. (1997) for a proof)

$$\max_{\alpha, i} \min(\alpha, 1 - \beta_{i+1}, \text{Inc}(B_{\beta_i}, A_{\alpha})) = \min_t \max(\mu_A(t), 1 - \mu_B(t)).$$

Thus, it establishes the general pattern (which generalizes the cases where only A or B is fuzzy)

$$\frac{(\neg p(x, y) \vee q(x, y), \mu_A(x))}{(p(B, s_0), 1)} \text{-----} (V)$$

$$(q(B, s_0), N_B(A))$$

where $N_B(A) = \inf_t \max(\mu_A(t), 1 - \mu_B(t))$ is the necessity measure of the fuzzy event A based on fuzzy information B .

The following example illustrates pattern (V) where A refers to a Cartesian product

$$\frac{(\text{age}(x,t) \wedge \text{age}(y,u) \rightarrow \neg \text{father}(x,y), \mu_{\text{approx_equal}}(t,u))}{(\text{age}(\text{John,about_30}),1), (\text{age}(\text{Tom,about_25}),1)} \text{-----}$$

$$(\neg \text{father}(\text{Tom,John}), N_{\text{about_30}} \times \text{about_25}(\text{approx_equal})).$$

Pattern (V) can be straightforwardly generalized when $\mu_A(x)$ is changed into $\mu_A(x_1, \dots, x_n)$ and μ_A can be decomposed under the form $\min_{i=1, \dots, n} \mu_{A_i}(x_i)$ (i.e., $A = A_1 \times \dots \times A_n$). Then if x_i is unified with a fuzzy constant B_i , $N_B(A)$ in (V) is changed into $N_{B_1 \times \dots \times B_n}(A) = \min_{i=1, \dots, n} N_{B_i}(A_i)$.

Pattern (V) is in agreement with the application of the generalized modus ponens (Zadeh, 1979) to the fact "X is B" and the rule "the more X is A, the more certain Z is C" which can be represented by the fuzzy relation $\mu_R(x,z) = \max(1 - \mu_A(x), \mu_C(z))$ (see Dubois and Prade, 1991). Indeed, the generalized modus ponens leads to

$$\sup_x \min(\mu_B(x), \mu_R(x,z)) = \max(\mu_C(z), 1 - N_B(A))$$

which is analogous to the conclusion given by (V), when C is not fuzzy: "Z is C" is certain at the degree $N_B(A)$.

Pattern (V) produces conclusions which are all the stronger as A is large and B is small. Indeed $N_B(A) \geq N_B(A')$ if $\mu_{A'} \leq \mu_A$ and $N_B(A) \geq N_{B'}(A)$ if $\mu_{B'} \leq \mu_B$. This points out that it is interesting to have the first premise in (V) established with the greatest possible weight (using the fusion inference rule). Thus, from $(\neg p(x) \vee q, \mu_A(x))$ and $(\neg p(x) \vee r, \mu_B(x))$ by proposition weakening and fusion, we can conclude $(\neg p(x) \vee q \vee r, \max(\mu_A(x), \mu_B(x)))$, which together with $(p(C), 1)$ enables us to deduce $(q \vee r, N_C(A \cup B))$, using (V). Note that we may have $N_C(A \cup B) > 0$, while $N_C(A) = 0 = N_C(B)$ leading to a trivial result if we apply (V) independently to each of the two premises we start with.

We may also think of cutting C into two parts C_1 and C_2 with $C = C_1 \cup C_2$ (where the union is defined by max operation) in such a way that $N_{C_1}(A) > 0$ and $N_{C_2}(B) > 0$. Then $(p(C), 1)$ entails $(p(C_1) \vee p(C_2), 1)$ (this can be seen by observing that $(C_1 \cup C_2)_\alpha = C_{1\alpha} \cup C_{2\alpha}$ and $\exists x \in C_{1\alpha} \cup C_{2\alpha} \Leftrightarrow \exists x \in C_{1\alpha} \vee \exists x \in C_{2\alpha}$). Then, by applying (V) twice, we can prove

$$\frac{(\neg p(x) \vee q, \mu_A(x))}{(\neg p(x) \vee r, \mu_B(x))} \text{-----}$$

$$\frac{(p(C_1) \vee p(C_2), 1)}{(q \vee r, \min(N_{C_1}(A), N_{C_2}(B)))}$$

However, we only have $\min(N_{C_1}(A), N_{C_2}(B)) \leq N_C(A \cup B)$ in general. So only the first procedure guarantees that the best lower bound is reached, as expected.

3.4. Resolution with Mutually Exclusive Fuzzy Constants

By nature, constant values are mutually exclusive in case they pertain to single valued attributes, i.e., it writes $\neg p(x,a) \vee \neg p(x,b)$ for all $a \neq b$, if $p(x,y)$ can be true for at most one value of y . For instance, we should have $\neg \text{age}(\text{Peter}, 25) \vee \neg \text{age}(\text{Peter}, x)$, $\forall x \neq 25$. This can be generalized to imprecise constants under the form: $\neg(\exists y \in A \ p(y)) \vee \neg(\exists z \in B \ p(z))$ (or equivalently $(\forall y \in A \ \neg p(y)) \vee (\forall z \in B \ \neg p(z))$) holds provided that $A \cap B = \emptyset$. More generally, such a mutual exclusiveness condition can be written for imprecise restrictions A and B as

$$(\neg p(x,y) \vee \neg p(x,z), \min(\mu_A(y), \mu_B(z), \text{Disjoint}(A,B))) \quad (\text{VI})$$

where $\text{Disjoint}(A,B) = 1$ if $A \cap B = \emptyset$ and is 0 otherwise. Note that if $y = z$ is possible, we have $\text{Disjoint}(A,B) = 0$. Let us assume now that we have the two additional pieces of information

$$\begin{cases} (\neg p(x,A) \vee q, \gamma) \\ (\neg p(x,B) \vee r, \delta) \end{cases}$$

or equivalently

$$\begin{cases} (p(x, A_{\alpha_i}) \vee q, \min(1 - \alpha_{i+1}, \gamma)), i=1,k \\ (p(x, B_{\beta_j}) \vee r, \min(1 - \beta_{j+1}, \delta)), j=1, \dots \end{cases}$$

Then by unification with (VI) written for $A = A_{\alpha_i}$ and $B = B_{\beta_j}$, using pattern (III) we obtain

$$(q \vee r, \min(\gamma, \delta, \max_{i,j} \min(1 - \alpha_{i+1}, 1 - \beta_{j+1}, \text{Disjoint}(A_{\alpha_i}, B_{\beta_j}))))).$$

It can be checked that it is equivalent to

$$(q \vee r, \min(\gamma, \delta, 1 - \prod(A,B)))$$

where $\prod(A,B) = \sup_s \min(\mu_A(s), \mu_B(s))$. $\prod(A,B)$ represents the possibility to have $A \cap B \neq \emptyset$ and thus $1 - \prod(A,B)$ is the certainty to have $A \cap B = \emptyset$.

Thus, given the clause expressing the mutual exclusiveness of constant values, we have established the following pattern

$$\begin{cases} (\neg p(x,y) \vee \neg p(x,z), 1 - \prod(\{y\}, \{z\})) \\ (p(x,A) \vee q, \alpha) \end{cases}$$

$$\frac{(p(x,B) \vee r, \beta)}{(q \vee r, \min(\alpha, \beta, 1 - \prod(A,B)))}. \quad (\text{VII})$$

where $\prod(\{y\},\{z\}) = 1$ if $y = z$ and is equal to 0 otherwise. A similar pattern appears in (Dubois and Prade, 1990), but expressed in terms of fuzzy predicates rather than with fuzzy constants.

For example, we have the pattern

$$\frac{(\neg \text{age}(x,y) \vee \neg \text{age}(x,z), 1 - \prod(\{y\},\{z\})) \\ (\text{age}(\text{John}, \text{young}), 1) \\ (\text{age}(\text{John}, \text{middle-aged}), 1)}{(\perp, 1 - \prod(\text{young}, \text{middle-aged}))}$$

This corresponds, for instance, to establish (by refutation) that if "John is young", "John is not middle aged" with certainty $1 - \prod(\text{young}, \text{middle-aged})$.

4. ALGORITHMIC ISSUES

A possibilistic clause (p,γ) is composed of an *extended possibilistic clause (epc)* p , —the logical part of the clause—, and a *valuation* γ . To implement a resolution algorithm for the extension proposed here, one needs two auxiliary algorithms: a unification algorithm, which verifies if it is possible to match the logical part of two clauses, and the valuation algorithm, which verifies the compatibility of fuzzy constants and fuzzy restrictions (on quantifiers) in the valuation part of a clause.

The main body of the resolution algorithm in our framework follows the same basic steps as the traditional resolution algorithm (Robinson, 1965); see also (Bittencourt, 1996). At the beginning of the process, the query one is interested to prove is negated (with valuation $\alpha=1$) and inserted in the knowledge base. Then two clauses $C_1 = (p_1, \gamma_1)$ and $C_2 = (p_2, \gamma_2)$ are selected. Given epc's p_1 and p_2 , the resolution algorithm creates a new epc p_3 , if a positive literal of p_1 "matches" with a negative literal from p_2 and vice-versa (the matching process is performed by the unification algorithm given below). p_3 is called the *resolvent* of p_1 and p_2 . The valuation for the new clause is then created and is evaluated by the valuation algorithm. This process goes on until an inconsistency is found, which proves that the original query is true, or it is terminated by some other means, leaving the truth of the query undefined.

Let l_1 (resp. l_2) be a literal of p_1 (resp. p_2), such that either l_1 or l_2 is a negative literal, but not both. If this condition is satisfied we define l_1' as being literal l_1 whence we remove its sign. Let V_i denote the variables involved in clause C_i and Δ be a set of expressions $\{x \leftarrow t / x \in V_1 \cup V_2, t \text{ is a term}\}$, where $x \leftarrow t$ means "t replaces x". Let us suppose $V_1 \cap V_2 = \emptyset$, l_1' will "match" l_2 if it is possible to find a set Δ such that $l_1' \hat{\delta} \Delta = l_2 \hat{\delta} \Delta$, where $l_1' \hat{\delta} \Delta$ indicates that all substitutions in Δ are applied to the variables in l_1' . In other words, l_1

"matches" l_2 if substituting the variables in $V_1 \cup V_2$ by the terms assigned to them in Δ , makes l_1' and l_2' become the same.

The resolution algorithm uses the unification algorithm to find the most general set of substitutions capable of making l_1 and l_2 match. Such a set is called a most general unifier. For instance the most general unifier for $l_1 = p(x_1, x_2, x_1, A)$, and $l_2 = \neg p(x_3, x_3, B, x_4)$, is given by $\Delta = \{x_1 \leftarrow B, x_2 \leftarrow B, x_3 \leftarrow B, x_4 \leftarrow A\}$.

If the unification algorithm cannot find a most general unifier Δ for l_1 and l_2 then resolution fails. Otherwise, the process continues as follows. Let p_1' (resp. p_2') be the remainder of epc p_1 (resp. p_2) when l_1 (resp. l_2) is taken away from it. Let $p_3' = (p_1' \vee p_2')$ and $\gamma_3' = \min(\gamma_1, \gamma_2)$. The resolution algorithm "resolves" (p_1, γ_1) and (p_2, γ_2) by creating a new clause (p_3, γ_3) , such that $p_3 = p_3' \hat{\circ} \Delta$ and $\gamma_3 = \gamma_3' \hat{\circ} \Delta$. Then the valuation algorithm is used to evaluate γ_3 . If $\gamma_3 = 0$ then resolution fails, otherwise (p_3, γ_3) is inserted in the knowledge base and the process restarts.

In the following we specify the language that models (p, γ) ; then we outline the unification and valuation algorithms. Issues such as the use of Horn clauses, or of any resolution techniques, and questions such as completeness are out of the scope of this paper.

4.1. Syntax of (p, γ)

The formal language that model clauses in the extended possibilistic logic proposed here is defined using:

- a set of punctuation symbols $S = \{(" , ") , " , " \}$.
- a set of connectives $\{\neg , \vee\}$.
- a set of domains $U = \{U_1 , U_2 , \dots\}$.
- a set of variables $V = \{x_1^{U_1} , x_2^{U_2} , \dots\}$, where $U_i \in U$ indicates the domain of the i^{th} variable.
- a set of well-defined function symbols $F = \{f_1^{U_1}_{U_{11}, \dots, U_{1m1}} , f_2^{U_2}_{U_{21}, \dots, U_{2m2}} , \dots\}$, where $U_i \in U$ indicates the image of the i^{th} function, and $U_{ij} \in U$ indicates the domain of each of its arguments.
- a set of predicate symbols $P = \{p_1 , p_2 , \dots\}$.
- a set of values $K = \{\alpha_1 , \alpha_2 , \dots\}$, $0 \leq \alpha_i \leq 1$.
- a set of fuzzy sets $A = \{A_1^{U_1} , A_2^{U_2} , \dots\}$, where $U_i \in U$ indicates the domain of the i^{th} fuzzy set.

When no confusion is possible, subscripts and superscripts will be dropped in the remaining of the text.

A constant in this framework is modeled by a fuzzy set A . A constant s in classical logic is modeled by a fuzzy set A in U , such that $\mu_A(s) = 1, \forall s' \in U, s' \neq s, \mu_A(s') = 0$.

A *term* is a variable, a constant, or expressions such as $f(t_1, \dots, t_n)$ where f is a function symbol and t_1, \dots, t_n are terms. *Literals* are defined as follows:

- If t_1, t_2, \dots, t_m are terms then p_i and $p_i(t_1, t_2, \dots, t_m)$ are *positive literals*, where $i, m \geq 1$.
- If t_1, t_2, \dots, t_m are terms, then $\neg p_i$ and $\neg p_i(t_1, t_2, \dots, t_m)$ are *negative literals*, where $i, m \geq 1$.

An *epc* is either a literal or a disjunction of epc's $p_1 \vee p_2$.

Valuations are defined as follows:

- $\alpha \in [0, 1]$ is a valuation.
- If A is a (n -ary) fuzzy set and t_1, \dots, t_n are terms, then $\mu_A(t_1, \dots, t_n)$ is a valuation.
- If each t_1, \dots, t_n are terms, then $1 - \Pi(t_1, t_2, \dots, t_n)$ is a valuation.
- if v_1 and v_2 are valuations then $\min(v_1, v_2)$ is a valuation.

It is important to note that in this framework we do not allow the use of Skolem constants or functions. Moreover, all functions are supposed to be well-defined, i.e., given fuzzy sets as arguments of a function, it is possible to calculate its value.

4.2. Unification and Valuation Algorithms

In the following we discuss some representation issues and then present the unification and valuation algorithms for the extension proposed here. Efficiency has not been looked for in the algorithms below, trying rather to preserve clarity.

The algorithms require that the domains of all constants, variables and functions used in an application are well specified. In the unification process, whenever 2 terms are compared, the compatibility between their domains is verified. For instance, we will only be able to calculate how much B is contained in A , if A and B are defined on compatible universes.

Since we are dealing with fuzzy constants, the unification process cannot be purely syntactical as in classical logic. In order to illustrate why a semantical unification process is also necessary, let us suppose we try to resolve clauses $(\neg p(A), 1)$ and $(p(A), 1)$. A pure syntactical unifier would resolve these clauses yielding $(\perp, 1)$ as resolvent. However, since $(p(A), 1)$ and $(\neg p(A), 1)$ are respectively equivalent to $(\exists x \in A p(x), 1)$ and $(\exists x \in A \neg p(x), 1)$ we should obtain $(\perp, 1)$ only if A is a precise fuzzy set, otherwise resolution should fail. Similarly, $(\neg p(x, x) \vee q(x), 1)$ and $(p(A, A), 1)$ should only yield $(q(A), 1)$ if A is a precise fuzzy set.

All functions have to be well-defined, i.e., should a function become grounded at a given moment, the system must have the means of calculating its value. This makes it possible to unify two different functions or a function with a constant. For instance, let us suppose that we originally start with 4 clauses $(p(f(x)) \vee \neg q(x), 1)$, $(\neg p(g(y)) \vee r(y), 1)$, $(q(A), 1)$ and $(\neg r(B), 1)$. In classical logic, with A and B simple non-fuzzy point-valued constants, it would be impossible to resolve these clauses, due to the exclusive use of syntactical unification. Here, since we only use well-defined functions, we would obtain $(\perp, 1)$, in the particular case where $f(A)$ and $g(B)$ are precise fuzzy sets (i.e., which reduce to ordinary singletons) and such that $f(A) = g(B)$.

Unification Algorithm

Let $\tau_1 = l_1'(t_{11}, \dots, t_{1n})$ and $\tau_2 = l_2'(t_{21}, \dots, t_{2n})$ respectively be the (unsigned) literals from p_1 and p_2 that we want to unify. Note that, in order to improve readability in the unification algorithm below, we do not make explicit the verification of the compatibility between universes of discourse.

Unify (τ_1, τ_2):

```

for i = 1, 2 do
  if  $\tau_i = \varphi_i(t_{i1}, \dots, t_{ik})$  where  $\varphi_i$  is a function symbol and all  $t_{ij}$  are constants
    then  $\tau_i = A$ , where A is the constant obtained by applying  $\varphi_i$  to the  $t_{ij}$ 's
  if  $\tau_1$  or  $\tau_2$  is a variable or a constant
  then if  $\tau_1 = \tau_2$ 
    then if ( $\tau_1$  and  $\tau_2$  are precise fuzzy sets) or ( $\tau_1$  and  $\tau_2$  are variables)
      then return {}
      else return fail
    else if ( $\tau_1$  is a variable) and ( $\tau_1$  does not appear in  $\tau_2$ )
      then return  $\{\tau_1 \leftarrow \tau_2\}$ 
      else if ( $\tau_2$  is a variable) and ( $\tau_2$  does not appear in  $\tau_1$ )
        then return  $\{\tau_2 \leftarrow \tau_1\}$ 
        else return fail
  else let  $\tau_1 = \varphi_1(t_{11}, \dots, t_{1k})$  and  $\tau_2 = \varphi_2(t_{21}, \dots, t_{2l})$ ,
    where each  $\varphi_i$  is a predicate or a function symbol
  if  $\varphi_1 = \varphi_2$  and  $k = l = n$ 
  then  $\Delta = \emptyset$ 
    for i = 1, ..., n do
       $\theta = \text{unify}(t_{1i}, t_{2i})$ 
      if  $\theta = \text{fail}$ 
        then return fail
      else for j = i+1, ..., n do
         $t_{1j} = t_{1j} \hat{\circ} \theta$ 
         $t_{2j} = t_{2j} \hat{\circ} \theta$ 

```

```
     $\Delta = \Delta \cup \theta$   
    return  $\Delta$   
else return fail
```

Valuation Algorithm

Let (p, γ) be a resolvent clause to which the substitutions obtained in the unification algorithm have already been applied. A, A_1, \dots, A_n and B, B_1, \dots, B_n denote fuzzy sets, $h(A)$ denote the height of fuzzy set A and $\alpha \in [0,1]$.

Each step in the algorithm below is repeated until its conditions no longer hold before moving to the next step.

Evaluate (γ) :

- Step 1. If $\gamma = \min(\gamma', \mu_A(\cdot))$ and such that $f(B_1, \dots, B_n)$ appears in one of the arguments of $\mu_A(\cdot)$, we substitute $f(B_1, \dots, B_n)$ by B , where B is the constant obtained when f is applied to the B_i 's. If $\gamma=0$ then return fail.
- Step 2. If $\gamma = \min(\gamma', \mu_A(B))$, we make $\gamma = \min(\gamma', \alpha)$, where $\alpha = N_B(A)$. If $\gamma=0$ then return fail.
- Step 3. If $\gamma = \min(\gamma', \mu_{A_1}(x), \mu_{A_2}(x))$ we make $\gamma = \min(\gamma', \mu_{A_1 \cap A_2}(x))$. When this step cannot be applied anymore, for each variable x there is at most a function $\mu_A(x)$ in γ . If $\gamma = 0$ then return fail.
- Step 4. If $\gamma = \min(\gamma', \mu_A(x))$ and x does not appear in p or in γ' , we make $\gamma = \min(\gamma', \sup_x \mu_A(x))$, which yields $\gamma = \min(\gamma', \alpha)$, where $\alpha = h(A)$. If $\gamma = 0$ then return fail.
- Step 5. If $\gamma = \min(\gamma', 1 - \Pi(A_1, \dots, A_n))$, we make $\gamma = \min(\gamma', \alpha)$, where α is the value obtained when we apply Π to the A_i 's. If $\gamma = 0$ then return fail.
- Step 6. If not fail then return γ .

4.3. Examples

In the following we show two examples which help to clarify and illustrate the functioning of the unification and valuation algorithms. Let U_t denote the domain of a term t .

Example 1.

$$\frac{\begin{array}{l} (p(x) \vee q(x), \min(\alpha, \mu_A(x))) \\ (\neg p(B), \beta) \end{array}}{\begin{array}{l} (q(B), \min(\alpha, \beta, \mu_A(B))) \text{ if } U_x = U_B \\ \equiv (q(B), \min(\alpha, \beta, N_B(A))) \\ \text{no resolution, otherwise} \end{array}}$$

$$\begin{array}{l}
(q(B), \min(\alpha, \beta, N_B(A))) \\
(\neg q(y), \min(\gamma, \mu_C(y))) \\
\hline
\left\{ \begin{array}{l}
(\perp, \min(\alpha, \beta, \gamma, N_B(A), N_B(C))) \text{ if } U_y = U_B \\
\equiv (\perp, \min(\alpha, \beta, \gamma, N_B(A), N_B(C))) \\
\text{no resolution, otherwise}
\end{array} \right.
\end{array}$$

Note that the order of the resolution steps is not important here. If the other possible order was used to resolve this set of clauses we would obtain $(\perp, \min(\alpha, \beta, \gamma, N_B(A \cap C)))$ as result, which is equivalent to $(\perp, \min(\alpha, \beta, \gamma, N_B(A), N_B(C)))$ since $\min(N_B(A), N_B(C)) = N_B(A \cap C)$.

Example 2.

$$\begin{array}{l}
(\neg q(x,y) \vee p(x \oplus y), \min(\alpha, \mu_A(x), \mu_B(y))) \\
(\neg p(z), \min(\beta, \mu_C(z))) \\
\hline
\left\{ \begin{array}{l}
(\neg q(x,y), \min(\alpha, \beta, \mu_A(x), \mu_B(y), \mu_C(x \oplus y))) \\
\text{no resolution, otherwise}
\end{array} \right. \\
\\
(\neg q(x,y), \min(\alpha, \beta, \mu_A(x), \mu_B(y), \mu_C(x \oplus y))) \\
(q(D,z), \beta) \\
\hline
\left\{ \begin{array}{l}
(\perp, \min(\alpha, \beta, \mu_A(D), \mu_B(y), \mu_C(D \oplus y))) \text{ if } U_x = U_D \text{ and } U_y = U_z \\
\equiv (\perp, \min(\alpha, \beta, N_D(A), \mu_B(y), \mu_C(D \oplus y))) \\
\text{no resolution, otherwise}
\end{array} \right.
\end{array}$$

N.B.: The symbolic expression $\mu_C(D \oplus y)$ is nothing but $N_D(C \dot{\wedge} \{y\})$ since $\mu_C(x \oplus y) = \mu_{C \dot{\wedge} \{y\}}(x)$ where $C \dot{\wedge} \{y\}$ is the fuzzy set C translated by $-y$. The valuation obtained in Example 2 holds for any y , so we can maximize on y . However, the expression $\max_y \min(\mu_B(y), N_D(C \dot{\wedge} \{y\}))$ cannot be further reduced in the general case.

5. CONCLUDING REMARKS

This paper has shown how fuzzy information can be handled in the setting of possibilistic logic using fuzzily restricted quantifiers and fuzzy constants. However this work remains preliminary at least in two respects; indeed a theoretical characterization of what is representable in this way, as well as completeness results, are still missing; besides algorithmic issues have been only sketched and the relationship with classical types of resolution and unification has not been studied in detail.

A related issue which is particularly worth mentioning and investigating, is the idea of approximate unification. The idea is then to work with a sorted possibilistic logic, where each sort is equipped with a fuzzy proximity relation, as suggested by

the following example where we may want to unify $(\neg \text{age}(x,30) \vee q(x),\alpha)$, $(\text{age}(\text{John},29), 1)$ considering that 29 is close to 30, thus leading to something of the form $(q(\text{John}), \min(\alpha, \mu_{\text{close}}(30,29)))$. Note that here, μ_{close} does not appear in the possibilistic clauses we start with, but is attached to the sort pertaining to the age universe. Then the unification of precise constants is no longer purely syntactic and based on perfect identity, but approximate and semantic. See (Arcelli, Formato and Gerla, 1997) for a first proposal on similarity-based unification.

More generally, this work may open the road to a new way of handling fuzzy information in a logical setting, without resorting to many-valued logics. Thus, comparison with recent works by Godo and Hajek (1996) is in order.

REFERENCES

- Arcelli F., Formato F., Gerla G. (1997) Extending unification through similarity relations. BUSEFAL (IRIT, Univ. P. Sabatier, Toulouse, France), No. 70, 3-12.
- Bel G., Dubois D., Farreny H., Prade H. (1986) Towards the use of fuzzy rule-based systems in the monitoring of manufacturing systems. In: Software for Discrete Manufacturing (J.P. Crestin, J.F. McWaters, eds.), North-Holland, 525-535.
- Bittencourt G. (1996) Inteligência Artificial: Ferramentas e Teorias. (in Portuguese) Instituto de Computação/Unicamp.
- Cayrol M., Farreny H., Prade H. (1982) Fuzzy pattern matching. Kybernetes, 11, 103-116.
- Dubois D., Lang J., Prade H. (1994a) Automated reasoning using possibilistic logic: Semantics, belief revision and variable certainty weights. IEEE Trans. on Data and Knowledge Engineering, 6(1), 64-71.
- Dubois D., Lang J., Prade H. (1994b) Possibilistic logic. In: Handbook of Logic in Artificial Intelligence and Logic Programming, Vol. 3 (D.M. Gabbay et al., eds.), Oxford Univ. Press, 439-513.
- Dubois D., Lang J., Prade H. (1994c) Handling uncertainty, context, vague predicates, and partial inconsistency in possibilistic logic. in: Fuzzy Logic and Fuzzy Control (Proc. of the IJCAI'91 Workshops on Fuzzy Logic and Fuzzy Control, Aug. 1991) (D. Driankov, P.W. Eklund, A.L. Ralescu, eds.), LNAI 833, Springer-Verlag, Berlin, 45-55.
- Dubois D., Prade H. (1988) Possibility Theory: An Approach to Computerized Processing of Uncertainty. Plenum Press, New York.
- Dubois D., Prade H. (1990) Resolution principles in possibilistic logic. Int. J. Approximate Reasoning, 4, 1-21.
- Dubois D., Prade H. (1991) Fuzzy sets in approximate reasoning — Part 1: Inference with possibility distributions. Fuzzy Sets and Systems, 40, 143-202.
- Dubois D., Prade H., Sabbadin R. (1997) A possibilistic logic machinery for qualitative decision. Working Notes of the AAAI'97 Spring Symp. Series on Qualitative Preferences in Deliberation and Practical Reasoning, Stanford, CA, March 24-26, 47-54.

- Fehrer D. (1990) A resolution calculus for a logic based on vaguely defined predicates. Proc. of the 9th Europ. Conf. on Artificial Intelligence (ECAI'90) (L. Carlucci Aiello, ed.), Stockholm, Sweden, Aug. 6-10, 268-273.
- Godo L., Hajek P. (1996) Fuzzy inference as deduction. Proc. of the 17th Linz Seminar on Fuzzy Set Theory, Linz, Austria, Feb. 20-24.
- Lang J., Dubois D., Prade H. (1991) A logic of graded possibility and certainty coping with partial inconsistency. Proc. of the 7th Conf. on Uncertainty in Artificial Intelligence, UCLA, Los Angeles, July 13-15, 1991, Morgan-Kaufmann, Palo Alto, CA, 188-196.
- Mukaidono M., Shen Z.L., Ding L. (1989) Fundamentals of fuzzy Prolog. Int. J. Approximate Reasoning, 3, 179-193.
- Novák V. (1990) On the syntactico-semantical complete-ness of first-order fuzzy logic. Kybernetika, 26, Part 1: 47-66 and Part 2: 134-154.
- Prade H. (1982) Modal semantics and fuzzy set theory. In: Fuzzy set and Possibility Theory — Recent Developments (R.R. Yager, ed.), Pergamon Press, 232-246.
- Rios-Filho L.G., Sandri S.A. (1995) Contextual fuzzy unification. Proc. of the 5th IFSA'95 Congress, São Paulo, Brazil, July, 81-84.
- Robinson J.A. (1992) Logic and logic programming. Communications of the ACM, 35(3), 41-65.
- Virtanen H.E. (1994) Fuzzy unification. Proc. of the IPMU'94 Conf., Paris, France, July 4-8, 1147-1152.
- Zadeh L.A. (1979) A theory of approximate reasoning. In: Machine Intelligence, Vol. 9 (J.E. Hayes et al., eds.), Elsevier, New York, 149-194.