APPROXIMATE REASONING TO UNIFY NORM-BASED AND IMPLICATION-BASED FUZZY CONTROL

by
Giangiacomo Gerla
Dipartimento di Matematica ed Informatica
Via S. Allende, 84081, Baronissi (Salerno) Italy.

Abstract. In Gerla [2000] a fuzzy logic in narrow sense is proposed as a theoretical framework for triangular norm based fuzzy control. In this note we show that the resulting theory is also able to express the implication-based fuzzy control.

Introduction

The aim of control theory is to define a function f : $X \to Y$ whose intended meaning is that f(x) is the correct answer given the input x. Fuzzy approach to control, as devised in Zadeh [1965], [1975]a, [1975]b and in Mamdani [1981], furnishes an approximation of such a (ideal) function f : $X \to Y$ on the basis of pieces of fuzzy information (fuzzy granules). This approximation is represented by a system of fuzzy IF-THEN rules like

IF x is A_i THEN y is B_i

where $i = 1,...,n$ and A_i and B_i are labels for fuzzy subsets a_i and b_i. We associate the i-rule with the Cartesian product $a_i \times b_i$ and the whole system with the fuzzy function $f = \cup_{i=1,...,n} a_i \times b_i$. A suitable defuzzification process enable us to define a function f' we consider a suitable approximation of f. Now, as it is well known, the interpretation of such a rule as a logical implication $A(x) \to B(y)$ in a formalized logic is rather questionable (see, e.g., Hájek [1998]). Then in Gerla [2000]a, we propose to give a logical meaning to a fuzzy IF-THEN rule by translating the system of rules into the set

$A_i(x) \wedge B_i(y) \to Good(x,y)$

of first order formulas. The intended meaning of $Good(x,y)$ is that given x the value y gives a correct control (see also Gerla [2000]). Since it is natural to assign suitable weights to these formulas, the information carried on by a system of fuzzy IF-THEN rules is represented by a fuzzy theory in a fuzzy logic. Such a theory is a fuzzy program, i.e. a fuzzy set of Horn clauses. So, the computation of the fuzzy function f is equivalent to the computation of the least fuzzy Herbrand model of this fuzzy program.

Now, in literature we have an alternative procedure for fuzzy control based on an implication operation (see, for example, Klir and Yuan [1995]). Namely, let $\to$ be a binary operation in U able to interpret the logical connective "implies". Then any rule "IF x is A_i THEN y is B_i" is associated with the fuzzy relation $a_i \to b_i$ defined by setting, for any $x \in X$ and $y \in Y$, $(a_i \to b_i)(x,y) = a_i(x) \to b_i(y)$. The whole system of rules is associated with the fuzzy function $f = \cap_{i=1,\dots,n}\, a_i \to b_i$.

In this note we show that also implication-based fuzzy control can be represented by a suitable fuzzy theory in fuzzy logic. This is achieved by admitting also negative information, i.e. more general theories. Then our proposal contains both the norm-based and implication-based fuzzy control.

1. Preliminaries

Let S be a set, then we denote by $\mathcal{P}(S)$, $\mathcal{P}_f(S)$, $\mathcal{F}(S)$) the class of all subsets, finite subsets, fuzzy subsets of S, respectively. A fuzzy subset of S, i.e. a map $s : S \to [0,1]$ it is also called *fuzzy granule* of S. Given $\lambda \in U$, we denote by $C(s,\lambda)$ the *λ-cut* $\{x \in S : s(x) \geq \lambda\}$ of s. The set $Supp(s) = \{x \in S : s(x) \neq 0\}$ is called *the support of* s. If $(s_i)_{i\in I}$ is *directed*, i.e., for any $i,j \in I$, an index h exists such that both s_i and s_j are contained in s_h, then the union $\cup_{i\in I} s_i$ is also denoted by $lim_{i\in I}s_i$. A *continuous T-norm*, in brief a *norm*, is any continuous, associative, commutative operation $\odot : U \times U \to U$, non-decreasing with respect both the variables and such that $x \odot 1 = x$. A *continuous T-co-norm*, in brief a *co-norm*, is an operation $\oplus$ obtained from a norm $\odot$ by setting

$$x \oplus y = 1\text{-}(1\text{-}x) \odot (1\text{-}y)$$

for any x, y in U. A basic example of norm is the minimum, we denote by $\sqcap$, whose associated co-norm is the maximum, we denote by $\sqcup$. Łukasiewicz norm is defined by setting $x \odot y = (x+y\text{-}1)\sqcup 0$, the related co-norm is defined by setting $x \oplus y = (x+y)\sqcap 1$. Another simple norm is the usual product whose relate co-norm is defined by setting $x \oplus y = x + y - xy$. Given a triangular norm $\odot$, two set X and Y and two fuzzy subsets $a : X \to U$ and $b : Y \to U$, the *Cartesian product* is the fuzzy subset $a\times b : X \times Y \to U$ of $X \times Y$ defined by setting

$$(a\times b)(x,y) = a(x)\odot b(y)$$

for any $x \in X$ and $y \in Y$. Given a finite subset X of S, we define the *inclusion degree* $Incl(X,s)$ *of* X *in* s (with respect. to $\odot$) by setting:

$$Incl(X,s) = \begin{cases} 1 & \text{if } X = \varnothing \\ & \end{cases}$$

$$s(x_1) \odot ... \odot s(x_n) \qquad \text{if } X = \{x_1,...,x_n\}.$$

A *fuzzy function* $f : X \sim> Y$ from X to Y is any fuzzy relation, i.e. any fuzzy subset f of $X \times Y$. We call *fuzzy operator* in S any map $J : \mathcal{F}(S) \to \mathcal{F}(S)$ and we say that J is *continuous* if

$$lim_{i\in I}\, \mathcal{D}(s_i) = \mathcal{D}(lim_{i\in I}\, s_i)$$

for every directed family $(s_i)_{i\in I}$ of elements in $\mathcal{F}(S)$. Moreover, we say that J is a *fuzzy closure operator* if:

(i) $s \subseteq s' \Rightarrow J(s) \subseteq J(s')$ (order-preserving),
(ii) $s \subseteq J(s)$ (inclusion),
(iii) $J(J(s)) = J(s)$ (idempotence).

A *closure system* is a class C of fuzzy subsets closed under intersections. A *fixed point* of J is a fuzzy subset s such that $J(s) = s$. We recall the following basic theorem (for the proof, see Gerla [2000]) .

Theorem 1.1. *Let $H : \mathcal{F}(S) \to \mathcal{F}(S)$ be a continuous operator such that $H(s) \supseteq s$ for any $s \in \mathcal{F}(S)$. Then the class of fixed points of H is a closure system. Let $\mathcal{D}(s)$ the last fixed point of H containing s. Then*

$$\mathcal{D}(s) = \cup_{n\in N} H^n(s).$$

The resulting operator $\mathcal{D} : \mathcal{F}(S) \to \mathcal{F}(S)$ is a continuous closure operator whose fixed points coincides with the fixed points of H.

We call $\mathcal{D}$ *the closure operator generated by H.*

2. Classical fuzzy control

The following is the main definition in fuzzy control based on a triangular norm. We consider a system $\mathcal{S}$ of IF-THEN fuzzy rules like

$$\begin{array}{l} \text{IF } x \text{ is } A_1 \text{ THEN } y \text{ is } B_1 \\ \qquad\qquad ... \\ \text{IF } x \text{ is } A_n \text{ THEN } y \text{ is } B_n \end{array} \tag{2.1}$$

where the labels A_i and B_i are interpreted by the fuzzy granules $a_i : X \to U$ and $b_i : Y \to U$. We associate any rule with the Cartesian product $a_i \times b_i : X \times Y \to U$ and the whole system of rules with the fuzzy function $f : X \sim> Y$ defined by

$$f = \cup_{i=1,...,n}\, a_i \times b_i. \tag{2.2}$$

Obviously, such a procedure depends on the triangular norm we consider to define the Cartesian product.

A totally different procedure for fuzzy control is based on an implication operation and not by a triangular norm (see, for example, Klir and Yuan [1995]). Namely, let $\rightarrow$ be a binary operation in U able to interpret the logical connective "implies" and consider the system

IF x is C_1 THEN y is D_1

$$\ldots \tag{2.3}$$

IF x is C_n THEN y is D_n

of IF-THEN fuzzy rules. Also, assume that each C_i is interpreted by the fuzzy subset c_i and each D_i by the fuzzy subset d_i. Then in the implication-based fuzzy control any rule "IF x is C_i THEN y is D_i" is associated with the fuzzy relation $c_i \rightarrow d_i$ defined by setting, for any $x \in X$ and $y \in Y$

$$(c_i \rightarrow d_i)(x,y) = c_i(x) \rightarrow d_i(y).$$

The whole system of rules is associated with the fuzzy function $f : X \rightsquigarrow Y$ defined by

$$f = \cap_{i=1,\ldots,n}\, c_i \rightarrow d_i. \tag{2.4}$$

In both cases a *defuzzification process* enables us to associate the fuzzy function f with a classical function f'. Usually the defuzzification process is obtained by the *centroid method* where we set, for every $r \in X$,

$$f'(r) = \frac{\int_Y f(r,y)\cdot y\, dy}{\int_Y f(r,y)\, dy}$$

In Picture 3 both the fuzzy function f and the result f' of the defuzzification process are represented (the triangular norm is the minimum $\sqcap$).

The final phase is the learning process in which the rules and the fuzzy granules associated with the labels are changed until we can accept f' as a good approximation of the ideal function $\underline{f}$. More information on fuzzy control are in Gottwald [1993] and Gerla [2000].

3. Fuzzy deduction systems

We denote by $\mathbb{F}$ a set whose elements we interpret as sentences of a logical language and we call *formulas*. If α is a formula and $\lambda \in U$, the pair (α,λ) is called a *signed formula*. To denote the signed formula (α,λ) we can write also

$\alpha \qquad (\lambda)$.

Any fuzzy set of formulas $s : \mathbb{F} \to U$ can be identified with the set $\{(\alpha,\lambda) \in \mathbb{F} \times U : s(\alpha) = \lambda\}$ of signed formulas. Conversely, any set T of signed formulas is associated with the fuzzy subset

$$s(x) = Sup\{\lambda : (x,\lambda) \in T\}.$$

We define a *fuzzy Hilbert system* as a pair $\mathcal{S} = (a, \mathbb{R})$ where a is a fuzzy subset of $\mathbb{F}$, the *fuzzy subset of logical axioms*, and $\mathbb{R}$ is a set of fuzzy rules of inference. In turn, a *fuzzy inference rule* is a pair $r = (r',r'')$, where

- r' is a partial n-ary operation on $\mathbb{F}$ whose domain we denote by $Dom(r)$,
- r'' is an n-ary operation on U preserving the least upper bound in each variable, i.e.

$$r''(x_1,..., Sup_{i\in I}\, y_i, ..., x_n) = Sup_{i\in I}\, r''(x_1, ..., y_i, ..., x_n). \qquad (3.1)$$

In other words, an inference rule r consists

- of a *syntactical component* r' that operates on formulas (in fact, it is a rule of inference in the usual sense),
- of a *valuation component* r'' that operates on truth-values to calculate how the truth-value of the conclusion depends on the truth-values of the premises (Zadeh [1975], Pavelka [1979]).

We indicate an application of an inference rule r by the picture

$$\frac{\alpha_1,...,\alpha_n}{r'(\alpha_1,...,\alpha_n)} \quad ; \quad \frac{\lambda_1,...,\lambda_n}{r''(\lambda_1,...,\lambda_n)}$$

whose meaning is that:

IF

you know that $\alpha_1,\ldots,\alpha_n$ are true (at least) to the degree $\lambda_1,\ldots,\lambda_n$

THEN

$r'(\alpha_1,\ldots,\alpha_n)$ is true (at least) at level $r''(\lambda_1,\ldots,\lambda_n)$.

A *proof* π *of a formula* α is a sequence $\alpha_1,...,\alpha_m$ of formulas such that $\alpha_m = \alpha$, together with the related *"justifications"*. This means that, for any formula α_i, we must specify whether

(i) α_i is assumed as a logical axiom; or

(ii) α_i is assumed as an hypothesis; or

(iii) α_i is obtained by a rule (in this case we must indicate also the rule and the formulas from $\alpha_1,...,\alpha_{i-1}$ used to obtain α_i).

We call *length of* π the number m. Observe that we have only two proofs of α whose length is equal to 1. The formula α with the justification that

α is assumed as a logical axiom and the formula α with the justification that α is assumed as an hypothesis. Moreover, as in the classical case, for any $i \leq m$, the initial segment $\alpha_1,...,\alpha_i$ is a proof of α_i we denote by $\pi(i)$. Differently from the crisp case, the justifications are necessary since different justifications of the same formula give rise to different valuations. Let $v : \mathbb{F} \to U$ be any initial valuation and π a proof. Then the *valuation* $Val(\pi,v)$ *of* π *with respect to* v is defined by induction on the length m of π as follows. If the length of π is 1, then we set

$Val(\pi,v) = a(\alpha_m)$ if α_m is assumed as a logical axiom,

$Val(\pi,v) = v(\alpha_m)$ if α_m is assumed as an hypothesis.

Otherwise, we set

$$Val(\pi,v) = \begin{cases} a(\alpha_m) & \text{if } \alpha_m \text{ is assumed as a logical axiom,} \\ v(\alpha_m) & \text{if } \alpha_m \text{ is assumed as an hypothesis,} \\ r''(Val(\pi(i(1)),v),\ldots,Val(\pi(i(n)),v)) & \text{if } \alpha_m = r'(\alpha_{i(1)},\ldots,\alpha_{i(n)}) \end{cases}$$

where, $1\leq i(1)<m,\ldots,1\leq i(n)<m$. If α is the formula proven by π, the meaning we assign to $Val(\pi,v)$ is that:

given the information v, π *assures that* α *holds at least at level* $Val(\pi,v)$.

Different proofs of the same formula gives different valuations. This suggests the following definition.

Definition 3.3. Given a fuzzy Hilbert's system $\mathcal{S}$, we call *deduction operator* associated with $\mathcal{S}$ the operator $\mathcal{D} : \mathcal{F}(\mathbb{F}) \to \mathcal{F}(\mathbb{F})$ defined by setting,

$$\mathcal{D}(v)(\alpha) = Sup\{Val(\pi,v) : \pi \text{ is a proof of } \alpha\}, \tag{3.2}$$

for every initial valuation v and every formula α.

The meaning of $\mathcal{D}(v)(\alpha)$ is still

given the information v, we may prove that α *holds at least at level* $\mathcal{D}(v)(\alpha)$,

but we have also that

$\mathcal{D}(v)(\alpha)$ *is the best possible valuation we can draw from the information v.*

The following proposition holds:

Proposition 3.4. *The deduction operator $\mathcal{D}$: $\mathcal{F}(\mathbb{F}) \to \mathcal{F}(\mathbb{F})$ of a fuzzy logic is a continuous closure operator.*

We say that a proof $\pi = \alpha_1,...,\alpha_n$ is *normalized* if the formulas in π are pairwise different and two integers h and k exist such that:
- $\alpha_1,...,\alpha_h$ are the formulas assumed as hypothesis,
- $\alpha_{h+1},...,\alpha_k$ the formulas justified as logical axiom,
- $\alpha_{k+1},...,\alpha_n$ are obtained by an inference rule.

In computing $\mathcal{D}(v)(\alpha)$ we can limit ourselves only to normalized proofs, obviously.

We are interested to a very simple logic in which $\mathbb{F}$ is the set of formulas of a first order logic, a the characteristic function of the set *Taut* of all logically true formulas and $\mathbb{R}$ contains the two fuzzy rules:

Generalization

$$\frac{\alpha}{\forall x_i(\alpha)} \quad ; \quad \frac{\lambda}{\lambda}$$

Fuzzy Modus Ponens

$$\frac{\alpha\,,\ \alpha \to \beta}{\beta} \quad ; \quad \frac{\lambda,\ \mu}{\lambda \odot \mu}$$

We call *canonical extension of a first order logic by a continuous triangular norm* $\odot$ such a kind of fuzzy logic. Also, we can consider some derived rule. As an example, if $Q(\alpha)$ denotes the universal closure of the formula α, we can consider the *Extended Generalization*

$$\frac{\alpha}{Q(\alpha)} \quad ; \quad \frac{\lambda}{\lambda}$$

that we can obtain by an iterate application of Generalization Rule. We have also the *Extended fuzzy Modus Ponens*

$$\alpha_1, \ldots ,\alpha_n\,,\ \alpha_1 \wedge \ldots \wedge \alpha_n \to \alpha \quad ; \quad \lambda_1, \ldots ,\lambda_n, \lambda$$

$$\frac{}{\alpha} \qquad \frac{}{\lambda_1 \odot \ldots \odot \lambda_n \odot \lambda}$$

we can obtain by observing that the formula

$$(\alpha_1 \wedge \ldots \wedge \alpha_n \to \alpha) \to (\alpha_1 \to (\ldots(\alpha_n \to \alpha)\ldots)$$

is logically true. Finally, we have the *Particularization Rule*

$$\frac{\alpha(x_1,\ldots,x_n)}{\alpha(t_1,\ldots,t_n)} \quad ; \quad \frac{\lambda}{\lambda}$$

where $t_1,\ldots,t_n$ are ground terms. Such a rule can be obtained by observing that the formula $\alpha(x_1,\ldots,x_n) \to \alpha(t_1,\ldots,t_n)$ is logically true.

Theorem 3.5. *Let $\mathcal{D}$ be the deduction operator of a canonical extension of a first order logic. Then,*

$$\mathcal{D}(v)(\alpha) = Sup\{Incl(X, v) : X \in \mathcal{P}_f(\mathbb{F}) \text{ and } X \vdash \alpha\} \tag{3.3}$$

Proof. Assume that $\alpha \in Taut$. Then, $\mathcal{D}(v)(\alpha) = 1$ and, since $\emptyset \vdash \alpha$ and $Incl(\emptyset,v) = 1$, (3.3) is proved. Otherwise, set

$$d = Sup\{v(x_1)\odot\ldots\odot v(x_n) : x_1, \ldots , x_n \vdash \alpha\}$$

and let $\alpha_1,\ldots,\alpha_n$ formulas such that $\alpha_1,\ldots,\alpha_n \vdash \alpha$. We claim that a proof π of α exists such that $Val(\pi,v) = v(\alpha_1)\odot\ldots\odot v(\alpha_n)$. In fact, recall that in first order calculus a weak form of Deduction Theorem holds and therefore that $\alpha_1,\ldots,\alpha_n \vdash \alpha$ entails that $Q(\alpha_1) \to (\ldots (Q(\alpha_n) \to \alpha))$ is logically true where $Q(\alpha)$ denotes the universal closure of α. Then, if $\alpha_1,\ldots,\alpha_n \vdash \alpha$, we obtain the following proof together with the related valuation

α_1,	$v(\alpha_1)$,
… ,	…
α_n, ,	$v(\alpha_n)$,
$Q(\alpha_1)$,	$v(\alpha_1)$,
…	…
$Q(\alpha_n)$,	$v(\alpha_n)$,
$Q(\alpha_1) \to (\ldots (Q(\alpha_n) \to \alpha))$,	1
$Q(\alpha_2) \to (\ldots (Q(\alpha_n) \to \alpha))$,	$v(\alpha_1)$
. . .	…
α	$v(\alpha_1) \odot \ldots \odot v(\alpha_n)$.

Since $Val(\pi,v) = v(\alpha_1) \odot ... \odot v(\alpha_n)$, this proves that $d \leq \mathcal{D}(v)(\alpha)$.

Conversely, to prove that $d \geq \mathcal{D}(v)(\alpha)$, observe that, for any $x \in U$, it is $x \odot x \leq x \odot 1 \leq x$ and therefore $x^n \leq x$ for any integer n. Let $\pi = \alpha_1,...,\alpha_m$ be any normalized proof of α and assume that $\alpha_1,...,\alpha_h$ are the formulas assumed as an hypothesis. Then it is immediate that $n(1), ..,n(h)$ exist such that

$$Val(\pi,v) = v(\alpha_1)^{n(1)} \odot ... \odot v(\alpha_h)^{n(h)}.$$

By observing that $\alpha_1,...,\alpha_k \vdash \alpha$ and that

$$v(\alpha_1)^{n(1)} \odot ... \odot v(\alpha_h)^{n(h)} \leq v(\alpha_1) \odot ... \odot v(\alpha_h),$$

we can conclude that $Val(\pi,v) \leq d$. Thus $\mathcal{D}(x) \leq d$. □

Proposition 3.6. *Let $\mathcal{D}$ be the deduction operator of the canonical extension of a first order logic by the minimum $\sqcap$. Then*

$$\mathcal{D}(v)(\alpha) = Sup\{\lambda \in U : C(v,\lambda) \vdash \alpha\}. \tag{3.4}$$

Proof. In the case that α is logically true, i.e. $\varnothing \vdash \alpha$, both the sides of (3.3) are equal to 1. Otherwise, observe that if X is a finite set such that $Incl(X,v) = \lambda$, then $X \subseteq C(v,\lambda)$ and therefore $C(v,\lambda) \vdash \alpha$. Conversely, if $C(v,\lambda) \vdash \alpha$, then a finite subset X of $C(v,\lambda)$ exist such that $X \vdash \alpha$. It is immediate that $Incl(X,v) = \lambda$. □

Note. Observe that (3.3) is based on a multivalued interpretation the metalogic claim

"a proof π of α exists whose hypotheses are contained in v".

This in accordance with the fact that in first order multivalued logics and in fuzzy logic the existential quantifier is usually interpreted by the operator $Sup : \mathcal{P}(U) \rightarrow U$. Now, this is rather questionable everywhere the logical connective "and" is interpreted by a triangular norm different from the minimum. In fact the operator used to interpret $\exists$ must extend to the infinitary case the interpretation of the binary connective "or", i.e. the co-norm $\oplus$ associated with $\odot$. Obviously, Sup satisfies such a condition only in the case that $\odot$ is the minimum and therefore $\oplus$ is the maximum. Then a natural candidate for the general case is the operator $\oplus : \mathcal{P}(U) \rightarrow U$ defined by setting, for any subset X of U,

$$\oplus(X) = Sup\{x_1 \oplus ... \oplus x_n : x_1,...,x_n \in X\}.$$

In accordance, should be interesting examine a fuzzy logic whose deduction operator is defined by

$$\mathcal{D}(v)(\alpha) = \oplus(\{Incl(X, v) : X \in \mathcal{P}_f(\mathbb{F}) \text{ and } X \vdash \alpha\}) \qquad (3.5)$$

Such a proposal requires further investigation, obviously. For example, it is not clear whether $\mathcal{D}$ is a closure operator or not.

4. Fuzzy programs and fuzzy Herbrand models

We recall some basic notions in logic programming (see, e.g., Lloyd [1987]). Let $\mathcal{L}$ be a first order language with some constants and denote by $\mathbb{F}$ the related set of formulas. A *ground term* of $\mathcal{L}$ is a term not containing variables, the set $U_{\mathcal{L}}$ of ground terms of $\mathcal{L}$ć is called the *Herbrand universe* for $\mathcal{L}$. If $\mathcal{L}$ is function-free, then $U_{\mathcal{L}}$ is the set of constants. A *ground atom* is an atomic formula not containing variables and the set $B_{\mathcal{L}}$ of ground atoms is called the *Herbrand base* for $\mathcal{L}$. We call an *Herbrand interpretation* any subset M of $B_{\mathcal{L}}$. The name is justified by the fact that M defines an interpretation of $\mathcal{L}$ in which:

- the domain is the Herbrand universe $U_{\mathcal{L}}$,
- every constant in $\mathcal{L}$ is assigned with themselves,
- any n-ary function symbol f in $\mathcal{L}$ is interpreted as the map associating any $t_1,...,t_n$ in $U_{\mathcal{L}}$ with the element $f(t_1,...,t_n)$ of $U_{\mathcal{L}}$
- any n-ary predicate symbol r is interpreted by the n-ary relation r' defined by setting

$$(t_1,...,t_n) \in r' \Leftrightarrow r(t_1,...,t_n) \in M.$$

A *ground instance* of a formula α is a closed formula β obtained from α by suitable substitutions of the free variables with closed terms. Given a set X of formulas, we set

$$Ground(X) = \{\alpha \in \mathbb{F} : \alpha \text{ is a ground instance of a formula } \beta \in X\}.$$

A *definite program clause* is either an atomic formula or a formula of the form $\beta_1 \wedge ... \wedge \beta_n \rightarrow \beta$ where β, $\beta_1,...,\beta_n$ are atomic formulas. We denote by PC the set of program clauses. A *definite program* is a set $\mathbb{P}$ of definite program clauses. We associate $\mathbb{P}$ with the operator $J_{\mathbb{P}} : \mathcal{P}(B_{\mathcal{L}}) \rightarrow \mathcal{P}(B_{\mathcal{L}})$ defined by setting, for any subset X of $B_{\mathcal{L}}$,

$$J_{\mathbb{P}}(X) = \{\alpha \in B_{\mathcal{L}} : \alpha_1 \wedge ... \wedge \alpha_n \rightarrow \alpha \in Ground(\mathbb{P}),\ \alpha_1, \ldots, \alpha_n \in X\}$$
$$\cup \{\alpha \in B_{\mathcal{L}} : \alpha \in Ground(\mathbb{P})\} \cup X.$$

$J_{\mathbb{P}}$ is called the *immediate consequence operator.* Such an operator is continuous, then, in accordance with Theorem 1.1, we denote by $\mathcal{H}_{\mathbb{P}}$ the closure operator generated by $J_{\mathbb{P}}$, i.e., for any set X of ground atoms

$$\mathcal{H}_{\mathbb{P}}(X)\ \check{S}= \cup_{n \in N}(J_{\mathbb{P}})^n(X). \tag{4.1}$$

Definition 4.1. We call *Herbrand model* of $\mathbb{P}$ any fixed point of $J_{\mathbb{P}}$ (equivalently, of $\mathcal{H}_{\mathbb{P}}$). Given a set X of ground atoms, we say that $\mathcal{H}_{\mathbb{P}}(X)$ is the *least Herbrand model for $\mathbb{P}$ containing X.* We denote by $M_{\mathbb{P}}$ the model $\mathcal{H}_{\mathbb{P}}(\emptyset)$ and we call it the *least Herbrand model for $\mathbb{P}$.*

The following theorem shows that the least Herbrand model for $\mathbb{P}$ is the set of ground atoms that we can derive from $\mathbb{P}$.

Theorem 4.2. *For every program $\mathbb{P}$,*

$$M_{\mathbb{P}} = \{\alpha \in B_{\mathcal{L}} : \mathbb{P} \vdash \alpha\}. \tag{4.2}$$

The above definitions can be extended in an obvious way to many-sorted languages.

To extend the above notions of logic programming to the fuzzy framework, observe that there is no semantics for the proposed fuzzy logic (see also the observation at the end of the paper). So, we define a *fuzzy Herbrand interpretation of $\mathcal{L}$* as the restriction of a fuzzy theory to $B_{\mathcal{L}}$. Like the classical case, m defines a multi-valued interpretation of $\mathcal{L}$ in the Herbrand universe in which any n-ary predicate symbol r is interpreted by the fuzzy n-ary relation r' on $U_{\mathcal{L}}$ defined by setting

$$r'(t_1,...,t_n) = m(r(t_1,...,t_n)).$$

We call *fuzzy program* any fuzzy subset $p : PC \rightarrow U$ of program clauses. We define the least fuzzy Herbrand model of p as the fuzzy subset of ground atoms that can be proved from p.

Definition 4.3. Let $\mathcal{D}$ be the deduction operator of a canonical extension of a predicate calculus by a norm and let p be a fuzzy program. Then, the *least fuzzy Herbrand model* for p is the fuzzy set $m_p : B_{\mathcal{L}} \to U$ defined by setting, for any $\alpha \in B_{\mathcal{L}}$,

$$m_p(\alpha) = \mathcal{D}(p)(\alpha). \tag{4.3}$$

Then, if α is a ground atom, in accordance with (3.3)

$$m_p(\alpha)= Sup\{Incl(\mathbb{P},p) : \mathbb{P} \in \mathcal{P}_f(Supp(p)) \text{ s.t. } \alpha \in M_{\mathbb{P}}\}. \tag{4.4}$$

Assume that the triangular norm under consideration is the minimum and denote by $\mathbb{P}(\lambda)$ the program $C(p,\lambda)$. Then, by Proposition 3.6,

$$m_p(\alpha) = Sup\{\lambda \in U : \alpha \in M_{\mathbb{P}(\lambda)}\}.$$

In the case that $Supp(p)$ is finite, in the co-domain of p there are only a finite number of elements $\lambda(1) > \lambda(2) > ... > \lambda(n)$ different from zero. As a consequence, to calculate $m_p(\alpha)$ it is sufficient to calculate the least Herbrand models $M_{\mathbb{P}(\lambda(1))} \subseteq \;...\; \subseteq M_{\mathbb{P}(\lambda(n))}$ by a parallel process.

5. Fuzzy control and logic programming

Consider a fuzzy system $\mathbb{S}$ of IF-THEN rules like

$$\begin{cases} \text{IF } x \text{ is } A_1 \text{ THEN } y \text{ is } B_1 \\ \quad ... \\ \text{IF } x \text{ is } A_n \text{ THEN } y \text{ is } B_n \end{cases} \tag{5.1}$$

To give a logical interpretation of such a system, we consider A_i and B_i as names for fuzzy predicates and not labels for fuzzy granules. In accordance, we interpret *"x is A_i"* and *"y is B_i"* as *"x satisfies A_i"* and *"y satisfies B_i"*, respectively. Moreover, we associate the IF-THEN fuzzy system (5.1) with the set

$$\begin{cases} A_1(x) \wedge B_1(y) \to Good(x,y) \qquad (\lambda_1) \\ \quad ... \\ A_n(x) \wedge B_n(y) \to Good(x,y) \qquad (\lambda_n) \end{cases} \tag{5.2}$$

of signed clauses, where $\lambda_1 =...=\lambda_n = 1$ and $Good(x,y)$ is a new predicate whose intended meaning is

"given x, y is a good value for the control variable".

The meaning of the value λ_i is that the i-rule is accepted at level λ_i. In the general case $\lambda_1,...,\lambda_n$ can be different from 1 and are the result of a learning process. Also, by assuming that A_i and B_j are interpreted by the fuzzy subsets a_i and b_j, we consider, for $i,j = 1,...,n$, $r \in X$ and $t \in Y$, the signed ground atoms

$A_i(r) \qquad (a_i(r))$

$B_j(t) \qquad (b_j(t))$.

In other words, we associate the system (5.1) with the fuzzy program $p : PC \to U$ defined by setting

$$p(\alpha) = \begin{cases} \lambda_i & \text{if } \alpha \text{ is the clause } A_i(x) \wedge B_i(y) \to Good(x,y), \\ a_i(r) & \text{if } \alpha \text{ is the ground atom } A_i(r), \\ & \\ b_i(t) & \text{if } \alpha \text{ is the ground atom } B_i(t), \\ 0 & \text{otherwise.} \end{cases} \tag{5.3}$$

Each element in X or in Y is considered as a constant. Therefore, the Herbrand universe of p is $X \cup Y$.

Theorem 5.1. *Define the fuzzy relation good* : $X \times Y \to U$, *by setting, for any* $r \in X$ *and* $t \in Y$

$$good(r,t) = \mathcal{D}(p)(Good(r,t)).$$

Then good coincides with the fuzzy function associated with the fuzzy control system (5.1).

Proof. Consider the fuzzy program p associated with the system (5.1)

$A_1(x) \wedge B_1(y) \to Good(x,y) \quad [\lambda_1]$

. . .

$A_n(x) \wedge B_n(y) \to Good(x,y) \quad [\lambda_n]$

$A_i(r) \qquad [a_i(r)]$

. . .

$B_j(t) \qquad [b_j(t)]$

where $\lambda_1, ... ,\lambda_n$ are elements in U, r varies in X and t varies in Y. Then, given the constants r and t, we can try to prove the ground atom $Good(r,t)$. Consider the ground instance of the first rule,

$A_1(r) \wedge B_1(r) \to Good(r,t)$

and the ground atoms

$A_1(r)$,

$B_1(t)$.
Then, by the extended fuzzy Modus Ponens rule, we can prove *Good*(r,t) at level $\lambda_1 \odot a_1(r) \odot b_1(t)$. Likewise, from the second fuzzy clause we obtain a proof of *Good*(r,t) able to prove *Good*(r,t) at level $\lambda_2 \odot a_2(r) \odot b_2(t)$ and so on. It is immediate that these are the only possible proofs of *Good*(r,t) and therefore that

$$good(r,t) = \mathcal{D}(p)(Good(r,t)) = Max\{\lambda_1 \odot a_1(r) \odot b_1(t), \dots, \lambda_n \odot a_n(r) \odot b_n(t)\}.$$

By using the notion of Cartesian product, and assuming that $\lambda_1 = \dots = \lambda_n = 1$, we can conclude that

$$good = (a_1 \times b_1) \cup \dots \cup (a_n \times b_n),$$

in accordance with Definition 2.1. □

Theorem 5.1 shows that we can look at the calculus of the fuzzy function associated with a IF-THEN system as at the calculus of the least Herbrand model of a suitable program. More precisely, in account of the fact that *"Good"* is the only predicate occurring in the head of a rule, we have complete information about all the predicates different from *"Good"*, and the only calculus we have to do is related to ground atoms like *"Good(r,t)"*. In other words, while Pictures 1 and 2 are given, Picture 3 is calculated. These tree pictures represent the least fuzzy Herbrand model of the fuzzy program *p.*

As we will show in the following, such a logical approach gives the possibility of expressing the information of an expertise in a more complete way.

6. Fuzzy control by implications and negative information.

In order to give a logical interpretation of implication-based fuzzy control, we assume that $\lambda \rightarrow \mu$ is equal to $\sim\lambda \oplus \mu$ where $\oplus$ is the Łukasiewicz disjunction. Moreover, we consider a new binary predicate name *Bad*(x,y) and the following system of fuzzy rules:

$$\begin{array}{l} (\neg C_1(x)) \wedge D_1(y) \rightarrow Bad(x,y)\ [\lambda_1] \\ \quad \dots \\ (\neg C_n(x)) \wedge D_n(y) \rightarrow Bad(x,y)\ [\lambda_n]. \end{array} \tag{6.2}$$

Theorem 6.1. *Consider the fuzzy control system* (7.1). *Then the calculus of the fuzzy function obtained by the implication procedure is equivalent to the calculus of the least fuzzy Herbrand model of the fuzzy definite program given by* (7.3) for $\lambda_1 = \dots = \lambda_n = 1$. *More precisely, the fuzzy* *function* (7.2) *coincides with the complement of the interpretation of the vague predicate Bad in such a model*.

Proof. Let *bad* be the interpretation of *Bad* in the least fuzzy Herbrand model of (7.3). Then, by proceeding as in Section 5, we have that

$$bad = \cup_{i=1,\dots,n} (-c_i)\times d_i.$$

Equivalently,

$$\sim bad(x,y) = \cap_{i=1,\dots,n} \sim(\sim c_i(x)\odot d_i(y)) = \cap_{i=1,\dots,n} (c_i(x)\oplus\sim d_i(y)) = \cap_{i=1,\dots,n} c_i(x) \to d_i(y)$$

and therefore

$$\sim bad = \cap_{i=1,\dots,n} c_i \to d_i = g. \tag{7.4}$$

□

Such a logical approach to fuzzy control enables us to emphasize the different meaning of the two procedures (see also Dubois and Prade [1997]). Indeed, in a sense, fuzzy control by a triangular norm is useful to give positive information (the fuzzy set of pairs (x,y) we consider good). Fuzzy control by an implication is useful to manage negative information (the fuzzy subset of pairs (x,y) we will avoid). Also, this approach enable us to consider the two different approaches at the same time. In fact we can consider a fuzzy definite program containing rules for the predicate *Good* and rules for the predicate *Bad*. Moreover, by adding to the language the predicate *Optimum* and the rule

$$Good(x,y)\wedge\neg Bad(x,y) \to Optimum(x,y),$$

we can compose the two different kinds of information. It is immediate to verify that the fuzzy relation *optimum* interpreting *Optimum* is defined by setting

$$optimum(x,y) = \begin{cases} 0 & \text{if } good(x,y) \le bad(x,y), \\ good(x,y) - bad(x,y) & \text{otherwise.} \end{cases}$$

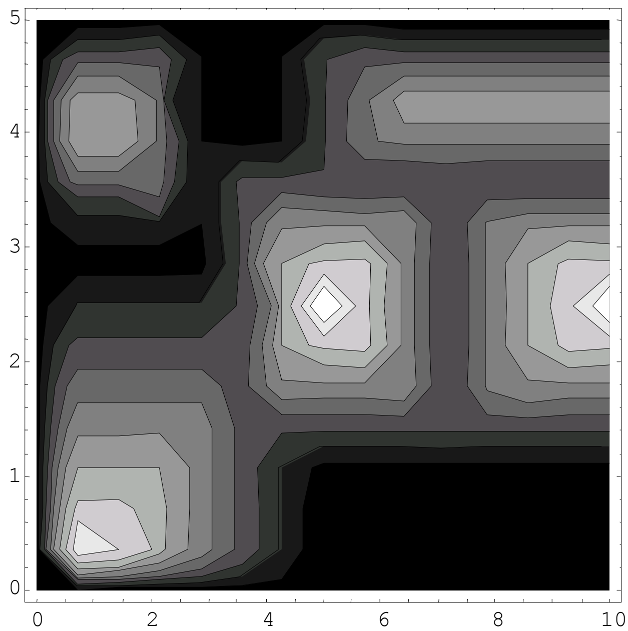
Clearly, we have to refer to the predicate *"Optimum"* and not *"Good"* in the successive defuzzification process.

This leads to the more general problem of using *"negative"* information in fuzzy logic programming. Unfortunately, this is an hard problem even in classical logic programming and it is faced, for example, by the *closed world rule* (see, e.g. Lloyd [1987]):

if a ground atom A is not a logical consequence of a program $\mathbb{P}$, then we are entitled to infer $\neg A$.

Obviously, this rule is rather questionable both from a semantic and a computational viewpoint. We can try to extend closed world rule to fuzzy logic programming by assuming that the negation $\neg A$ of a ground atom A is true at degree $1-\mathcal{D}(p)(A)$. As in the classical case, this *"rule"* creates several problems. As an example, if a proof π gives a lower bound $Val(\pi,p)$ for the truth value of A then $1-Val(\pi,p)$ gives an upper bound for the truth value of $\neg A$. On the other hand, the fuzzy logic deduction machinery as proposed in literature is not able to manage these upper bounds. Perhaps, the extended notion of fuzzy logic by constraints examined in Chapter 5 should be the appropriate tool to manage both positive and negative information.

In any case, if the negation occurs in the body of the rules of the simple fuzzy definite programs we associate with a fuzzy IF-THEN system, no difficulty arises.



Picture 6 : the predicate *Safe*

Example 1 (Negative information for a safe control). Suppose we need to take into account that there are some control actions we have to avoid. For instance, assume that a too fast control y is considered dangerous. Then, we can express this by adding to the fuzzy definite program giving *Good* the following rule:

$$Clearly(Veryfast)(y) \rightarrow Dangerous(y).$$

Obviously, it should be possible to consider a more complex definition of the vague predicate *Dangerous*. Once the vague predicate *Dangerous* is defined, we can define the predicate *"Safe"* by adding the rule

$$Good(x,y) \wedge \neg(Dangerous(y)) \rightarrow Safe(x,y).$$

Denote the interpretations of *Dangerous* and *Safe* by *dangerous* : $Y \rightarrow U$ and *safe* : $X \times Y \rightarrow U$, respectively. Then, given $r \in X$ and $t \in Y$, the first clause enables us to calculate

$$dangerous(t) = \mathcal{D}(p)(Dangerous(t)) = clearly(veryfast(t)).$$

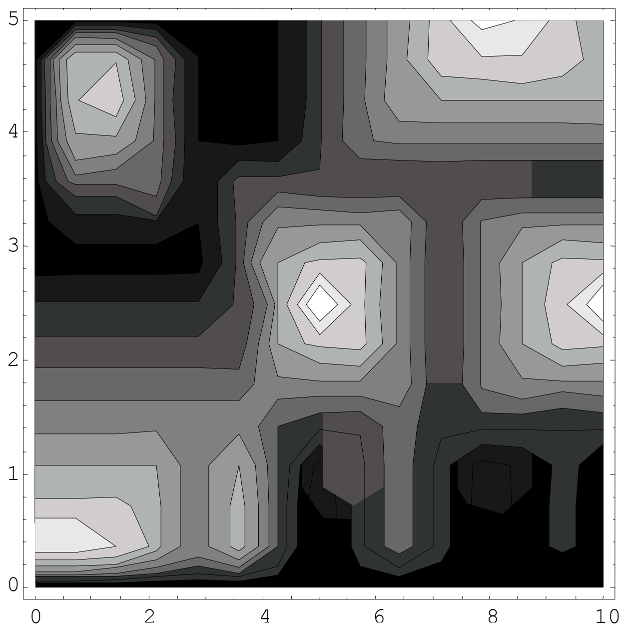
By the closed world rule

$$\mathcal{D}(p)(\neg Dangerous(t)) = 1 - \mathcal{D}(p)(Dangerous(t)) = 1 - clearly(veryfast(t)).$$

Then, by the second clause

$$safe(r,t) = \mathcal{D}(p)(Safe(r,t)) = good(r,t) \odot (1 - clearly(veryfast(t)).$$

Clearly, in such a case we have to refer to the predicate *"Safe"* and not *"Good"* in the successive defuzzification process. Picture 6 represents such a new predicate.



Picture 7 : adding the default rule

Example 2 (Negative information for a default rule). Another interesting use of the negation connective is the possibility of defining a *"default"* rule, i.e., a rule enabling us to choose a control in the case in which no condition $A_1,...,A_n$ in (5.2) is satisfied. As an example, assume that in this case an expert suggests a slow y. Then, by assuming that $Domain(x)$ is the formula $A_1(x) \vee ... \vee A_n$, we can add the rule

$$\neg Domain(x) \wedge Slow(y) \rightarrow Good(x,y).$$

In such a case the degree of completeness of the system increases (see Picture 7).

Observe that in the just given examples, the fuzzy relation *safe* is contained in the fuzzy relation *good* while the default rule increases such a relation. This shows that, by adding new information, it is possible both:

- to increase the area of the fuzzy upper covering of the objective function $\underline{f}$, in order to get completeness, i.e., to be sure that the whole set of points of $\underline{f}$ is covered,
- to decrease such an area, in order to get a more precise representation of the objective function $\underline{f}$.

Negative information for a safe control. The use of *"negative"* information is very delicate in classical logic programming. This is obtained by the *closed world rule*, for example (see, e.g. Lloyd [1987]). Such a rule says that if a ground atom A is not a logical consequence of a program $\mathbb{P}$, then we are entitled to infer $\neg A$. Such a rule is useful in several cases but rather questionable both from a semantical and computational viewpoint. We can tray to extend it to fuzzy logic programming by assuming that the negation $\neg A$ of a ground atom A is true at level $1\text{-}\mathcal{D}(p)(A)$. As in the classical case, this *"rule"* originates several difficulties. As an example, if a proof π gives a lower bound $Val(\pi,p)$ for the truth value of A then $1\text{-}Val(\pi,p)$ gives an upper bound for the truth value of $\neg$A. Unfortunately, the fuzzy logic deduction machinery as proposed in literature is not able to manage these upper bounds. Some suggestions for an approach to fuzzy logic in which this is possible can be find in Gerla [1999]).

In any case, in the simple fuzzy programs we associate with a fuzzy IF-THEN system no difficulty arises since we have a complete description of all the predicates different from *Good*. Consequently, the negation of such predicates is at semantical level, in a sense, and it can be achieved directly by the complement operator.

As an example suppose that we need to take into account that there are some control actions we have to avoid. For instance, assume that we consider dangerous a *"too fast"* control x. Then, we can express this by adding the following rule

$$Clearly(Veryfast)(y) \rightarrow Dangerous(y).$$

In accordance we can define the predicate *"Safe"* by adding the rule

$$Good(x,y) \wedge \neg(Dangerous(y)) \rightarrow Safe(x,y).$$

Denote by $dangerous : Y \rightarrow U$ and $safe : X\times Y \rightarrow U$ the interpretations of *Dangerous* and *Safe*, respectively. Then, given $r \in X$ and $t \in Y$, the first clause enables us to calculate

$$dangerous(t) = \mathcal{D}(p)(Dangerous(t)) = clearly(veryfast(t)).$$

By the closed world rule

$$\mathcal{D}(p)(\neg Dangerous(t)) = 1\text{-}\ \mathcal{D}(p)(Dangerous(t)) = 1\text{-}clearly(veryfast(t)).$$

Then, by the second clause

$$safe(r,t) = \mathcal{D}(p)(Safe(r,t)) = (good(r,t)\odot(1\text{-}clearly(veryfast(t)).$$

In the case of Łukasiewicz norm,

$$safe(x,y) = \begin{cases} 0 & \text{if } good(x,y) \leq clearly(veryfast(y)), \\ good(x,y) - clearly(veryfast(y)) & \text{otherwise.} \end{cases}$$

Obviously, we have to refer to the predicate *"Safe"* and not *"Good"* in the successive defuzzification process (see Picture 6).

Negative information for a default rule. Another interesting use of the negation is the possibility of defining a *"default"* rule, i.e., to suggest the control we have to choice in the case in which no condition *"Little"*, *"Medium"*, *"Big"*, *"Verybig"*, *"Small"* is satisfied. As an example, assume that in this case an expertise suggests to choose a slow y. Then, by assuming that $Domain(x)$ is the formula

$$Little(x) \vee Small(x) \vee Medium(x) \vee Big(x) \vee Verybig(x),$$

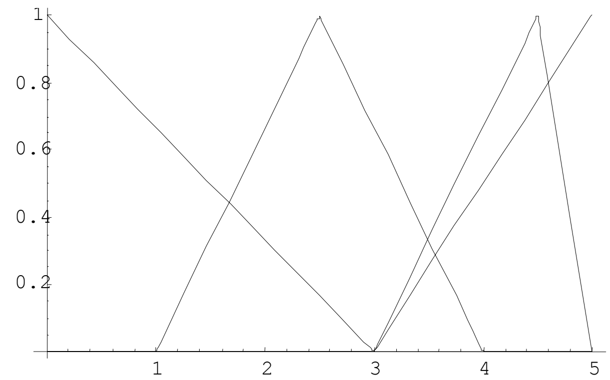
we can add the rule

$$\neg Domain(x) \wedge Slow(y) \rightarrow Good(x,y)$$

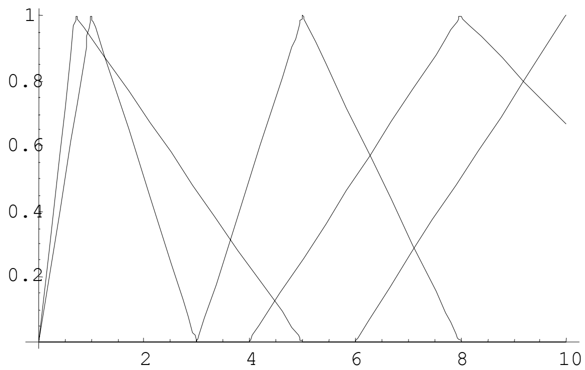
In such a case the predicate *"Good"* is represented by Picture 7 and the degree of completeness of the system increases.

Notice that the fuzzy relation *safe* is contained in the fuzzy relation *good* while the default rule increases the fuzzy relation interpreting the predicated *Good*. This shows that, by adding new information, it is possible both:

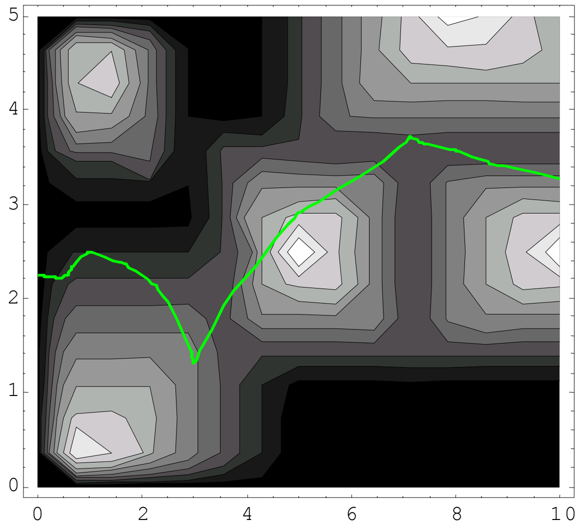
- to increase the area of the fuzzy upper covering of the ideal function $\underline{f}$, in order to obtain completeness, i.e. to be sure that the whole set of points of $\underline{f}$ is covered,
- to decrease such an area, in order to obtain a more precise representation of $\underline{f}$.



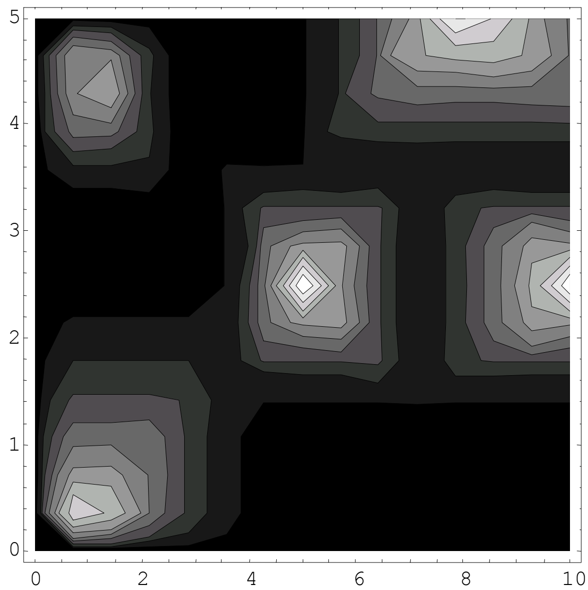
Picture 2: fuzzy granules of *Y*



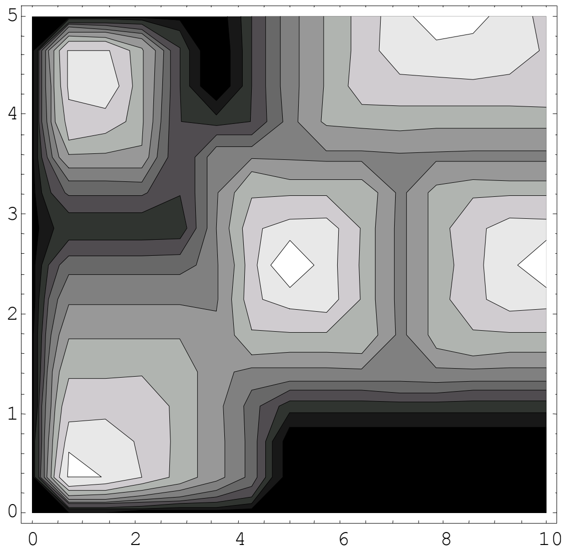
Picture 1: fuzzy granules of *X*



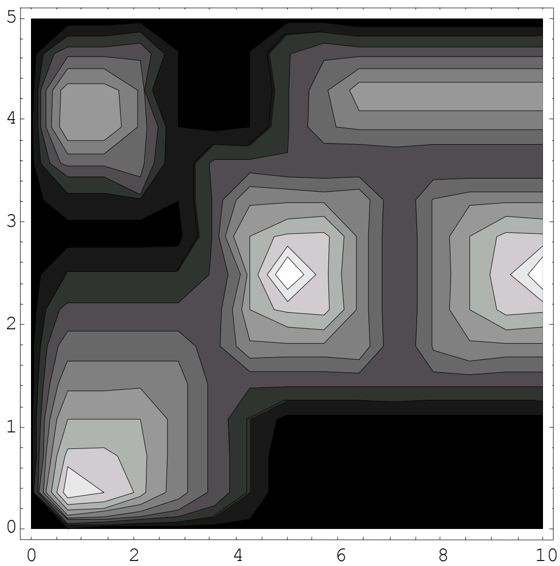
Picture 3: the fuzzy function and the result of the defuzzyfication process.



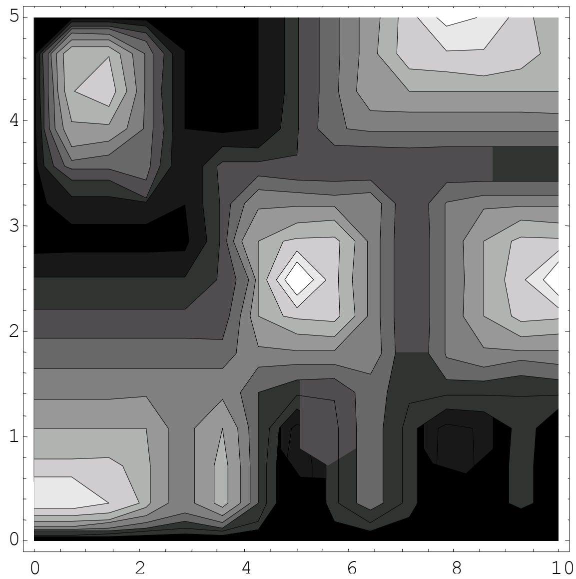
Picture 4 : the predicate *Clearly*(*Good*)



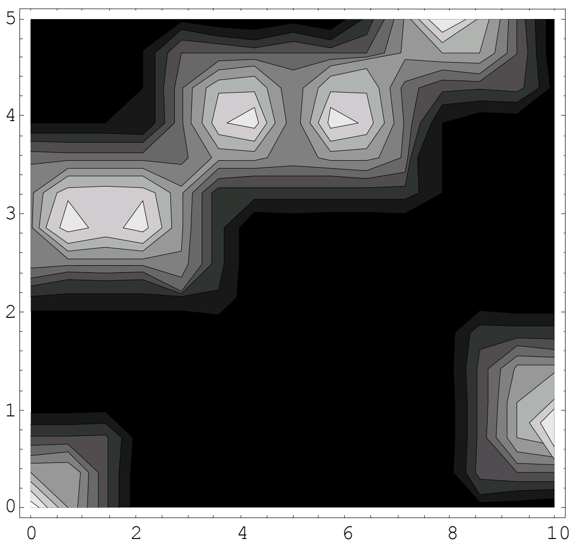
Picture 5: the predicate *Vaguely*(*Good*)



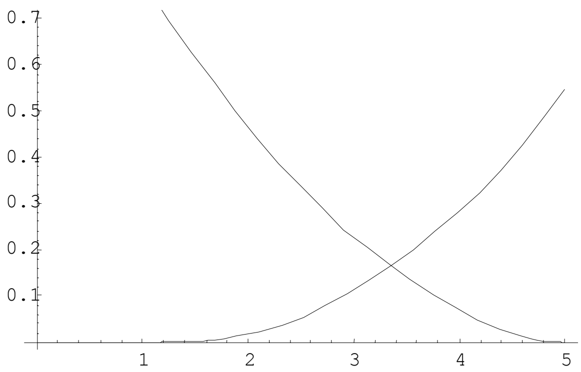
Picture 6 : the predicate *Safe*



Picture 7 : adding the default rule



Picture 8 : control by similarity



Picture 9

References

1. Biacino L., Gerla G. (1998) Logics with approximate premises, *International J. of Intell. Syst.*, **13**: 1-10.
2. Dubois D., Prade H. (1987) Necessity measures and the resolution principle, *IEEE Trans. Systems Man and Cybernetics*, **17**: 474-478.
3. Gerla G. (1994) An extension principle for fuzzy logics, *Mathematical Logic Quarterly*, **40**: 357-80.
4. Gerla G., (2000) Fuzzy control as a fuzzy deduction system, *Fuzzy Sets and Systems*, to appear.
5. Gerla G. (2000) Fuzzy Logic: *Mathematical Tools for Approximate Reasoning*, Kluwer Ac. Publishers.
6. Godo L., Hájek P. (1999) Fuzzy inference as Deduction, *J. of Applied Non-Classical Logics*, **9**, 37-60.
7. Goguen J.A. (1968/69) The logic of inexact concepts, *Synthese*, **19:** 325-373.
8. Gottwald S. (1993) *Fuzzy Sets and Fuzzy Logic*, Wiesbaden: Vieweg.
9. Hájek P. (1998) *Metamathematics of Fuzzy Logic*, Kluwer Ac. Publishers.
10. Hájek P. (1999) On the metamathematics of fuzzy logic, in *Discovering World with Fuzzy Logic: Perspectives and Approaches to Formalization of Human-Consistent Logical Systems*, (V. Novak, I. Perfilieva, editors), Springer-Verlag, Heidelberg.
11. Klawonn F., Novák V. (1996) The relation between inference and interpolation in the framework of fuzzy systems, *Fuzzy Sets and Systems,* **81**, 331-354.
12. Lloyd J.W. (1987) *Foundations of Logic Programming*, Springer-Verlag, Berlin.
13. Mamdani, E.H., Gaines, B.R. (1981) *Fuzzy Reasoning and its Applications*, Academic Press, London.
14. Mendelson E. (1964) *Introduction to Mathematical Logic*, D. Van Nostrand Company, Princeton.
15. Pavelka J. (1979) On fuzzy logic I: Many-valued rules of inference, *Zeitschr. f. math. Logik und Grundlagen d. Math.*, **25:** 45-52.
16. Ying M.S. (1994) A logic for approximate reasoning, *J. of Symbolic Logic*, **59**: 830-837.
17. Zadeh L.A. (1965) Fuzzy Sets, *Information and Control*, **12:** 338-353.
18. Zadeh L.A. (1965) Fuzzy logic and approximate reasoning, *Synthese*, **30:** 407-428.
19. Zadeh L.A. (1975) The concept of a linguistic variable and its application to approximate reasoning I, II, III, *Inf. Sci.* **8**: 199-275, 301-357; **9**: 43-80.