Fuzzy Logic Programming and Fuzzy Control

by
Giangiacomo Gerla
Department of Mathematics and Computer Science
Via S. Allende, 84081, Baronissi (Salerno) Italy

Abstract. The paper concerns fuzzy logic programming. As an example, we show that is not restrictive to confine ourselves to fuzzy Herbrand interpretations in giving a semantics for fuzzy programs. Also, we show that the resulting apparatus gives a unifying theoretical framework for fuzzy control.

Keywords: Fuzzy logic programming, Herbrand interpretation, fuzzy control.

1. Introduction and preliminaries

The basic ideas of fuzzy logic (in narrow sense) where formulated by L. Zadeh (see [19], [20]), J. A. Goguen (see [9]) and J. Pavelka (see [14], [15], and [16]). The aim of such a logic is to formalize the *"approximate reasoning"* we use in everyday life and this by admitting predicates as *"big"*, *"near"*, *"slow"* which are vague in nature. These predicates are interpreted by the notion of *"fuzzy subset"*, i.e. generalized characteristic functions with values in a complete lattice. Fuzzy logic programming is a very promising chapter of fuzzy logic whose purpose is to build up intelligent data-base systems with "flexible" answers, expert systems able to consider vague predicates and so on (see, for example [5]).

In this paper at first we recall some basic notions in fuzzy logic programming, which are well known in literature (see, for example, [2], [13], [17] and [18]). Also, we show that is not restrictive to confine ourselves to fuzzy Herbrand interpretations in giving a semantics for fuzzy programs. This is done by showing that any homomorphism preserves the universal formulas. Moreover, by developing some ideas sketched in [8], we relate fuzzy logic programming with fuzzy control (see also P. Hájek in [10]). This gives a unifying and rigorous framework for both conjunction-based and implication-based fuzzy control.

Let $\boldsymbol{L}$ be a complete, completely distributive, lattice whose elements we call *truth values* and let S be a nonempty set. Then, an *L-subset* or *fuzzy subset* of S is any map $s : S \to \boldsymbol{L}$ from S to $\boldsymbol{L}$. Given $x \in S$, we say that the value $s(x)$ is the *membership degree of* x *to* s. For any $\lambda \in \boldsymbol{L}$, the set $\{x\in S : s(x)\geq\lambda\}$ is called the *λ-cut* of s. The subset $Supp(s) = \{x \in S : s(x) \neq \mathbf{0}\}$ is called the *support* of s. If $\mathbf{0}$ denotes the minimum and $\mathbf{1}$ the maximum of $\boldsymbol{L}$, then we call *crisp* any L-subset whose values are in $\{\mathbf{0},\mathbf{1}\}$. We can identify any subset X of S with the crisp subset $c_X : S \to \{\mathbf{0},\mathbf{1}\}$ such that $c_X(x) = \mathbf{1}$ if $x \in X$ and $c_X(x) = \mathbf{0}$ in the case that $x\notin X$. In particular, we identify $\varnothing$ with the map constantly equal to $\mathbf{0}$ and S with the map constantly equal to $\mathbf{1}$. We say that a fuzzy subset s is *contained* in a fuzzy subset s' provided that $s(x) \leq s'(x)$ for any $x \in S$. In such a case, we write $s \subseteq s'$. Given a family $(s_i)_{i\in I}$ of fuzzy subsets, the *union* $\bigcup_{i\in I}s_i$ is the fuzzy subset defined by setting $\bigcup_{i\in I}s_i(x) = Sup\{s_i(x) : i\in I\}$ for any $x \in S$. The *intersection* $\bigcap_{i\in I}s_i$ is defined by setting $\bigcap_{i\in I}s_i(x) = Inf\{s_i(x) : i\in I\}$. With respect to these operations, the class $\mathcal{F}(S)$ of all the L-subsets of S is a complete lattice, i.e. the direct power of $\boldsymbol{L}$ with set index S. Let $S_1,...,S_n$ be sets, then an n-ary *L-relation* on $S_1,...,S_n$ is any L-subset of the Cartesian product $S_1\times...\times S_n$. A binary L-relation $f : S_1\times S_2 \to \boldsymbol{L}$ is also called (*non-deterministic*) *L-function* from S_1 to S_2. The idea is that, given the input $x \in S_1$, the output is not an element in S_2 but the fuzzy subset $s : S_2 \to \boldsymbol{L}$ of elements of S_2 defined by setting $s(y) = f(x,y)$ for any $y\in S_2$. Let $\otimes$ be a binary operation in $\boldsymbol{L}$ and $a : S_1\to \boldsymbol{L}$ and $b : S_2 \to \boldsymbol{L}$ two fuzzy subsets of S_1 and S_2, respectively. Then the ⊗-*Cartesian* product $a\times b : S_1\times S_2 \to \boldsymbol{L}$ is defined by setting, for any $(x,y) \in S_1\times S_2$,

$$(a\times b)(x,y) = a(x)\otimes b(y).$$

2. Fuzzy interpretations of a first order language

Let $\mathcal{L}$ be a first order language defined, as usual, by:

- a set LC of *logical connectives*,
- the *universal quantifier* $\forall$,
- a nonempty set C of *constants*,
- a set OS of *operation symbols*,
- a set RS of *relation symbols*,
- an *arity* map $ar : (RS \cup OS \cup LC) \rightarrow N.$

As usual, we consider also a sequence $x_1, x_2, \ldots$ of symbols that we call *variables* and the brakets (,). If s is a symbol such that $ar(s) = n$, then we say that s is *n-ary*. Terms and formulas are defined as in classic first order logic. Namely, we assume that:
- the constants and the variables are terms,
- if h is an k-ary operation symbol and $t_1,...,t_k$ are terms, then $h(t_1,...,t_k)$ is a term.

A term is *ground* if no variable occurs in it. Also, an *atomic formula* is an expression like $r(t_1,...,t_n)$, where r is an n-ary relation symbol and $t_1,...,t_n$ are terms. The whole set F of *formulas* is defined by assuming that:
- each atomic formula is a formula,
- if l is an k-ary logical connective and $\alpha_1,...,\alpha_k$ are formulas, then $l(\alpha_1,...,\alpha_k)$ is a formula,
- if α is a formula and x_i a variable, then $\forall x_i(\alpha)$ is a formula.

A formula is *ground* if no variable or quantifier occur in it, a ground atom is called a *fact*. Then a ground formula is obtained from facts by the logical connectives. The notions of *free occurrence* of a variable in a formula, of *quantifier-free formula*, of *universal formula* and so on, are defined as usual. We write $t(x_1,...,x_n)$ to emphasize that the variables occurring in the term t are among $x_1,...,x_n$ and, similarly, $\alpha(x_1,...,x_n)$ to emphasize that the variables in the formula α are among $x_1,...,x_n$.

We will define the semantics by taking in account the possibility that some relation symbols represent vague predicates that we model by the notion of L-subset.

Definition 2.1. An ***L**-interpretation* or *fuzzy interpretation* of $\mathcal{L}$ is a pair (D,I), where D is a nonempty set we call *domain* and I is a map associating any n-ary function symbol h with a function $I(h) : D^n \rightarrow D$, any constant c with an element $I(c) \in D$ and any n-ary relation symbol r with an *L-relation* $I(r): D^n \rightarrow \boldsymbol{L}$.

Sometimes we write I to denote the fuzzy interpretation (D,I). We interpret the terms as in the classical case.

Definition 2.2. Given a fuzzy interpretation (D,I) of $\mathcal{L}$ and a term $t(x_1,...,x_n)$, the interpretation of t is the map $I(t) : D^n \rightarrow D$ defined by setting:
- $I(x_i)(d_1,...,d_n) = d_i$,
- $I(c)(d_1,...,d_n) = I(c)$,
- $I(h(t_1,...,t_k))(d_1,...,d_n) = I(h)(I(t_1)(d_1,...,d_n),...,I(t_k)(d_1,...,d_n))$.

To interpret the formulas, we assume that any n-ary logical connective $l \in LC$ is associated with an n-ary operation $\underline{l} : \boldsymbol{L}^n \rightarrow \boldsymbol{L}$. The following definition enables us to associate any formula with the related *extension*.

Definition 2.3. Given a formula $\alpha(x_1,...,x_n)$, the *extension* $I(\alpha)$: $D^n \rightarrow \boldsymbol{L}$ of α is the n-ary fuzzy relation defined by setting:
- $I(r(t_1,...,t_p))(d_1,...,d_n) = I(r)(I(t_1)(d_1,...,d_n),..., I(t_p)(d_1,...,d_n))$,
- $I(l(\alpha_1,...,\alpha_k))(d_1,...,d_n) = \underline{l}(I(\alpha_1)(d_1,...,d_n),...,I(\alpha_k)(d_1,...,d_n))$,
- $I(\forall x_i(\alpha))(d_1,...,d_n) = Inf_{d \in D}\, I(\alpha)(d_1,\ldots,d_{i-1}, d,d_{i+1},\ldots,d_p)$.

Trivially, if α is a closed formula, then $I(\alpha)$ is a constant map, i.e. the quantity $I(\alpha)(d_1,...,d_n)$ does not depend on the elements $d_1,...,d_n$. In such a case we denote again by $I(\alpha)$ such a constant.

Definition 2.4. Let α be any formula and $\forall x_1...,\forall x_n(\alpha)$ be its universal closure. Then we set

$$\|\alpha\|_I = I(\alpha)(\forall x_1,...,\forall x_n(\alpha)).$$

As an obvious extension of the notion of set of axioms in classical logic, we call *fuzzy set of hypotheses* or *fuzzy theory* any fuzzy set $\tau : F \to \boldsymbol{L}$ of formulas.

Definition 2.5. Let $\tau : F \to \boldsymbol{L}$ be a fuzzy set of hypotheses. An L-interpretation (D, I) is called a *model* of τ (for short, $I \vDash \tau$), if $\|\varphi\|_I \geq \tau(\varphi)$ for any $\varphi \in F$.

Observe that, in accordance with such a definition of model, the value $\tau(\varphi)$ is not intended as the truth value of φ but as a lower-bound constraint on the possible truth value of φ. In other words, the information carried on by a fuzzy set of hypothesis τ is that, for any formula φ *"the truth value of φ is greater than or equal to $\tau(\varphi)$"*.

Definition 2.6. Let $\tau : F \to \boldsymbol{L}$ be a fuzzy set of hypotheses. Then the fuzzy set $Lc(\tau)$ of *logical consequences* of τ is defined by setting:

$$Lc(\tau)(\varphi) = Inf\{\|\varphi\|_I : I \vDash \alpha\}. \tag{2.1}$$

The operator $Lc : \mathcal{F}(F) \to \mathcal{F}(F)$ defined by (2.1) is called the *logical consequence operator*.

In a sense, $Lc(\tau)(\varphi)$ is the best lower-bound constraint on the truth value of φ that we can find given the available information τ. It is easy to prove that Lc is a closure operator in the lattice $\mathcal{F}(F)$.

3. Homomorphisms and preserving theorems

Given two $\boldsymbol{L}$-interpretations (D,I) and (D',I'), we call *homomorphism from* (D,I) *into* (D',I') any map $f : D \to D'$ such that:

- f is a homomorphism of the algebraic structures defined by I and I', i.e.,

$$f(I(c)) = I'(c) \quad ; \quad f(I(h)(d_1,...,d_n)) = I'(h)(f(d_1),...,f(d_n)),$$

for any $c \in C$ and any n-ary operation symbol h and $d_1,...,d_n \in D$,

- for every n-ary relation r,

$$I(r)(d_1,...,d_n) \leq I'(r)(f(d_1),...,f(d_n)). \tag{3.1}$$

We say that f is *full* if

$$I(r)(d_1,...,d_n) = I'(r)(f(d_1),...,f(d_n)). \tag{3.2}$$

An *isomorphism* is a homomorphism which is one-to-one and full. Equivalently, an isomorphism is an one-to-one homomorphism whose inverse is again a homomorphism. Observe that the notion of homomorphism does not depend on the interpretation of the logical connectives. Given an interpretation (D,I) and $\lambda \in \boldsymbol{L}$, we call λ*-cut* of I the classical interpretation (D,I_λ) defined by setting $I_\lambda(r) = C(I(r),\lambda)$ for any relation symbol r, $I_\lambda(c) = I(c)$ for any $c \in C$ and $I_\lambda(h) = I(h)$ for any operation symbol h.

Proposition 3.1. *Let* (D,I) *and* (D',I') *be two* $\boldsymbol{L}$*-interpretations and* $f : D \to D'$ *be a map. Then* f *is a* (*full*) *homomorphism from* (D,I) *into* (D',I') *if and only if, for any* $\lambda \in \boldsymbol{L}$, f *is a* (*full*) *homomorphism from the cut* (D,I_λ) *into the cut* (D',I'_λ). *Consequently,* f *is an isomorphism between* (D,I) *and* (D',I') *if and only if* f *is an isomorphism between the cuts* (D,I_λ) *and* (D',I'_λ) *for any* $\lambda \in \boldsymbol{L}$.

Proof. Assume that f is a homomorphism from (D,I) to (D',I'). Then, for any $\lambda \in \boldsymbol{L}$ and for any *n-ary* relation symbol r,

$$(d_1,...,d_n) \in C(I(r),\lambda) \Leftrightarrow I(r)(d_1,...,d_n) \geq \lambda$$

$$\Rightarrow I'(r)(f(d_1),...,f(d_n)) \geq I(r)(d_1,...,d_n) \geq \lambda \Leftrightarrow (f(d_1),...,f(d_n)) \in C(I'(r),\lambda).$$

This proves that f is a homomorphism from (D,I_λ) into (D',I'_λ). Conversely, assume that, for any $\lambda \in \boldsymbol{L}$, f is a homomorphism from (D,I_λ) into (D',I'_λ) and therefore that

$$I(r)(d_1,...,d_n) \geq \lambda \Rightarrow I'(r)(f(d_1),...,f(d_n)) \geq \lambda.$$

Then, by setting $\lambda = I(r)(d_1,...,d_n)$, we obtain that $I'(r)(f(d_1),...,f(d_n)) \geq I(r)(d_1,...,d_n)$. This proves that f is a homomorphism from (D,I) into (D',I'). The remaining part of the proof is trivial. □

As in classical logic, given a homomorphism f and a term $t(x_1,...,x_n)$, for every $d_1,...,d_n \in D$,

$$f(I(t)(d_1,...,d_n)) = I'(t)(f(d_1),...,f(d_n)). \tag{3.3}$$

Theorem 3.2. *Let (D,I) and (D',I') be two fuzzy interpretations, $\varphi(x_1,...,x_n)$ a quantifier-free formula and $f : D \rightarrow D'$ be a full homomorphism. Then, for any $d_1,...,d_n \in D$,*

$$I(\varphi)(d_1,...,d_n) = I'(\varphi)(f(d_1),...,f(d_n)). \tag{3.4}$$

If $\psi \in F$ is universal,

$$I(\psi)(d_1,...,d_n) \geq I'(\psi)(f(d_1),...,f(d_n)). \tag{3.5}$$

and therefore

$$\|\psi\|_I \geq \|\psi\|_{I'}. \tag{3.6}$$

Proof. We will prove (3.4) by induction on the complexity of φ. Indeed, assume φ is the atomic formula $r(t_1,...,t_p)$, where r is an n-ary relation symbol and $t_1,...,t_p$ are terms. Then, by (3.2) and (3.3),

$$\begin{aligned} I(\varphi)(d_1,...,d_n) &= I(r)(I(t_1)(d_1,...,d_n),...,I(t_p)(d_1,...,d_n)) \\ &= I'(r)(f(I(t_1)(d_1,...,d_n)),...,f(I(t_p)(d_1,...,d_n))) \\ &= I'(r)(I'(t_1)(f(d_1),...,f(d_n)),...,I'(t_p)(f(d_1),...,f(d_n))) \\ &= I'(\varphi)(f(d_1),...,f(d_n)). \end{aligned}$$

Assume that $\varphi = l(\alpha_1,...,\alpha_k)$ and that (3.4) is satisfied by $\alpha_1,...,\alpha_k$. Then,

$$\begin{aligned} I(l(\alpha_1,...,\alpha_k))(d_1,...,d_n) &= \underline{l}(I(\alpha_1)(d_1,...,d_n),..., I(\alpha_k)(d_1,...,d_n)) \\ &= \underline{l}(I'(\alpha_1)(f(d_1),...,f(d_n)),..., I'(\alpha_k)(f(d_1),...,f(d_n)) \\ &= I'(l(\alpha_1,...,\alpha_k))(f(d_1),...,f(d_n)). \end{aligned}$$

This proves the first part of the theorem. Let ψ be an universal formula. We will prove (3.5) by induction on the number m of quantifiers in ψ. If $m = 0$, (3.5) follows from (3.4). Assume that $m \neq 0$ and that ψ is equal to a formula like $\forall x_i(\alpha)$, where α contains m-1 universal quantifiers. Then, in accordance with the induction hypothesis and the inclusion $\{f(d) : d \in D\} \subseteq D'$,

$$\begin{aligned} I(\forall x_i(\alpha))(d_1,...,d_n) &= Inf_{d\in D}\, I(\alpha)(d_1,...,d,...,d_n) \\ &\geq Inf_{d\in D}\, I'(\alpha)(f(d_1),...,f(d),...,f(d_n)) \geq Inf_{d'\in D'}\, I'(\alpha)(f(d_1),...,d',...,f(d_n)) \\ &= I'(\forall x_i(\alpha))(f(d_1),...,f(d_n)). \end{aligned}$$

□

Not surprisingly, the following theorem proves that two isomorphic fuzzy interpretations are *logically equivalent*, i.e. they satisfy each formula at the same degree.

Theorem 3.3. *Let f be an isomorphism between (D,I) and (D', I'). Then for any formula α, we have that*

$$I(\alpha)(d_1,...,d_n) = I'(\alpha)(f(d_1),...,f(d_n)). \tag{3.7}$$

In particular, if (D,I) and (D', I') are isomorphic, then, for any formula α,

$$\|\alpha\|_I = \|\alpha\|_{I'}.$$

Proof. We prove (3.7) by induction on the complexity of α. If α is atomic, (3.7) was just proved in Theorem 3.2. Again, in proving Theorem 3.2 we showed that if l is a logical k-ary connective and $\alpha_1,...,\alpha_k$ are formulas satisfying (3.7), then $l(\alpha_1,...,\alpha_k)$ satisfies (3.7). Assume that α satisfies (3.7). Then, since $\{f(d) : d \in D\} = D'$,

$$\begin{aligned} I(\forall x_i\alpha)(d_1,...,d_n) &= Inf_{d\in D}\, I(\alpha)(d_1,\ldots,d,\ldots,d_p) \\ &= Inf_{d\in D}\, I'(\alpha)(f(d_1),\ldots,f(d),\ldots,f(d_p)) = Inf_{d'\in D'}\, I'(\alpha)(f(d_1),\ldots,d',\ldots,f(d_p)) \end{aligned}$$

$= I'(\forall x_i(\alpha)(f(d_1),...,f(d_n))$.

Thus, (3.7) is satisfied by any formula. □

4. Fuzzy Herbrand interpretations and universal theories.

As usual, we call *Herbrand universe* the set $\mathcal{U}$ of *ground* terms of the language $\mathcal{L}$ and *Herbrand base* the set $B_{\mathcal{L}}$ of *facts*, i.e. atomic ground formulas of $\mathcal{L}$.

Definition 4.1. An $\boldsymbol{L}$-interpretation (D,I) is an $\boldsymbol{L}$-*Herbrand interpretation* (for short *H-interpretation*) provided that:

- D is equal to the Herbrand universe $\mathcal{U}$,
- for any *n-ary* operation symbol h, $I(h)(t_1,...,t_n) = h(t_1,...,t_n)$,
- for any constant c, $I(c) = c$.

Then, two H-interpretations differ only for the relational part. As in the classical case, we can identify the H-interpretations with the fuzzy subsets of $B_{\mathcal{L}}$.

Proposition 4.2. *There is a one-one-correspondence between the H-interpretations and the fuzzy subsets of* $B_{\mathcal{L}}$. *Namely, we associate any H-interpretation* $(\mathcal{U},\mathcal{I})$ *with the fuzzy subset* $s_{\mathcal{I}}$ of $B_{\mathcal{L}}$ *defined by setting*:

$$s_{\mathcal{I}}(r(t_1,...,t_n)) = \mathcal{I}(r(t_1,...,t_n)),$$

for any fact $r(t_1,...,t_n)$. *Moreover, we associate any fuzzy subset s of* $B_{\mathcal{L}}$ *with the H-interpretation* $(\mathcal{U},\mathcal{I}_s)$ *where, for any n-ary relation symbol r,* $\mathcal{I}_s(r)$ *is the fuzzy relation in* $\mathcal{U}$ *defined by setting*

$$\mathcal{I}_s(r)(t_1,...,t_n) = s(r(t_1,...,t_n))$$

for any $t_1,...,t_n$ *in* $\mathcal{U}$.

In account of such a proposition, in the following we call *H-interpretation* any fuzzy subset of $B_{\mathcal{L}}$. To prove some basic properties of the H-interpretations, we need some further notations. Namely, let $t(x_1,...,x_n)$ be a term. Then, given the terms $t_1,...,t_n$, we write $t(t_1,...,t_n)$ to denote the term obtained by substituting each x_i with t_i in $t(x_1,...,x_n)$. Trivially, if t is ground, then $t(t_1,...,t_n)$ is equal to t. Likewise, given the formula $\alpha(x_1,...,x_n)$, if $t_1,...,t_n$ are terms, $\alpha(t_1,...,t_n)$ denotes the formula obtained from $\alpha(x_1,...,x_n)$ by substituting each variable x_i which is free in α with t_i. Trivially, if α is closed, then $\alpha(t_1,...,t_n)$ coincides with α.

Proposition 4.3. *Let* $(\mathcal{U},\mathcal{I})$ *be an H-interpretation,* $t(x_1,...,x_n)$ *to be a term and* $t_1,...,t_n \in \mathcal{U}$. *Then*

$$\mathcal{I}(t)(t_1,...,t_n) = t(t_1,...,t_n). \tag{4.1}$$

Moreover, if $\alpha(x_1,...,x_n)$ *is a formula,*

$$\mathcal{I}(\alpha)(t_1,...,t_n) = ||\alpha(t_1,...,t_n)||_{\mathcal{I}}. \tag{4.2}$$

Consequently,

$$||\forall x_1...\forall x_n(\alpha)||_{\mathcal{I}} = Inf\{||\alpha(t_1,...,t_n)||_{\mathcal{I}} \ : t_1,...,t_n \in \mathcal{U}\}. \tag{4.3}$$

Proof. We will prove (4.1) by induction on the complexity of t as in the classical case. Indeed, if t is either a variable or a constant, then (4.1) is obvious. Assume that (4.1) is satisfied by the terms $s_1,...,s_k$ and let h be an k-ary operation symbol. Then (4.1) is satisfied by $t = h(s_1,..,s_m)$. Indeed,

$$\mathcal{I}(t)(t_1,..,t_n) = \mathcal{I}(h)(\mathcal{I}(s_1)(t_1,...,t_n),..., \mathcal{I}(s_m)(t_1,...,t_n))$$

$$= \mathcal{I}(h)(s_1(t_1,...,t_n),...,s_k(t_1,...,t_n)) = h(s_1(t_1,...,t_n),...,s_m(t_1,...,t_n)) = t(t_1,...,t_n).$$

Again, we will prove (4.2) by induction on the number $n(\alpha)$ of connectives and quantifiers in α. Indeed, assume that $n(\alpha) = 0$, and therefore that α is an atomic formula as $r(s_1,...,s_m)$. Then

$$\mathcal{I}(\alpha)(t_1,...,t_n) = \mathcal{I}(r)(\mathcal{I}(s_1)(t_1,...,t_n),..., \mathcal{I}(s_m)(t_1,...,t_n))$$
$$= \mathcal{I}(r)(s_1(t_1,...,t_n),..., s_m(t_1,...,t_n)) = \mathcal{I}(r(s_1(t_1,...,t_n),..., s_m(t_1,...,t_n))$$
$$= \|\alpha(t_1,...,t_n)\|_{\mathcal{I}} .$$

Assume that $n(\alpha) \neq 0$ and that $\alpha = l(\alpha_1,...,\alpha_k)$. Then, since $n(\alpha_1)<n(\alpha)$, ... ,$n(\alpha_k)<n(\alpha)$, by induction hypothesis, (4.2) is satisfied by each α_i. Consequently,

$$\mathcal{I}(l(\alpha_1,...,\alpha_k))(t_1,...,t_n) = \underline{l}(\mathcal{I}(\alpha_1)(t_1,...,t_n),..., \mathcal{I}(\alpha_k)(t_1,...,t_n))$$
$$= \underline{l}(\|\alpha_1(t_1,...,t_n)\|_{\mathcal{I}},..., \|\alpha_k(t_1,...,t_n)\|_{\mathcal{I}}) = \|l(\alpha_1,...,\alpha_k)(t_1,...,t_n)\|_{\mathcal{I}}.$$

This proves that (4.2) is satisfied by $l(\alpha_1,...,\alpha_k)$. Finally, assume that α is the formula $\forall x_i(\beta)$ and let $t_1,...,t_n$ be elements in $\mathcal{U}$. Then, $n(\beta)<n(\alpha)$ and $n(\beta(t_1,...,x_i,...t_n))<n(\alpha)$. Thus, by induction hypothesis, for any $t\in\mathcal{U}$, $\mathcal{I}(\beta)(t_1,...,t,...,t_n) = \|\beta(t_1,...,t,...,t_n)\|_{\mathcal{I}}$ and $\mathcal{I}(\beta(t_1,...,x_i,...,t_n))(t) = \|\beta(t_1,...,t,...,t_n)\|_{\mathcal{I}}$. Consequently, since $\forall x_i(\beta(t_1,...,x_i,...,t_n))$ is equal to $(\forall x_i(\beta))(t_1,...,t_n)$, we get:

$$\mathcal{I}(\forall x_i(\beta))(t_1,...,t_n) = Inf\{\mathcal{I}(\beta)(t_1,...,t,...,t_n) : t\in\mathcal{U}\} = Inf\{\|\beta(t_1,...,t,...,t_n)\|_{\mathcal{I}} : t\in\mathcal{U}\}$$
$$= Inf\{\mathcal{I}(\beta(t_1,...,x_i,...,t_n))(t) : t\in\mathcal{U}\} = \|\forall x_i(\beta(t_1,...,x_i,...,t_n))\|_{\mathcal{I}}$$
$$= \|(\forall x_i(\beta))(t_1,...,t_n)\|_{\mathcal{I}}.$$

This proves that (4.2) is satisfied by the formula $\forall x_i(\beta)$, too. □

Given an $\boldsymbol{L}$-interpretation (D,I), the *H-interpretation associated with* (D,I) is the H-interpretation $(\mathcal{U},\mathcal{I})$ defined by the fuzzy set of facts which are true in (D,I). This means that, for any n-ary relation symbol r, it is

$$\mathcal{I}(r)(t_1,...,t_n) = I(r)(I(t_1),...,I(t_n)) = I(r(t_1,...,t_n)),$$

for any $t_1,...,t_n \in \mathcal{U}$. The proof of the following proposition is trivial.

Proposition 4.4. *Let* (D,I) *be an* $\boldsymbol{L}$*-interpretation and* $(\mathcal{U},\mathcal{I})$ *be the fuzzy H-interpretation associated with* (D,I)*. Then the map* $f : \mathcal{U}\to D$*, defined by setting* $f(t) = I(t)$ *for any* $t\in\mathcal{U}$*, is a full homomorphism from* $\mathcal{I}$ *into I. Consequently, if* φ *is a quantifier-free formula, then for any* $t_1,..., t_n \in \mathcal{U}$:

$$\mathcal{I}(\varphi)(t_1,...,t_n) = I(\varphi)(I(t_1),...,I(t_n)). \tag{4.4}$$

If ψ *is universal,*

$$\mathcal{I}(\psi)(t_1,...,t_n) \geq I(\psi)(I(t_1),...,I(t_n)), \tag{4.5}$$

and therefore

$$\|\psi\|_{\mathcal{I}} \geq \|\psi\|_I. \tag{4.6}$$

We say that a fuzzy theory $\tau : F \to \boldsymbol{L}$ is *universal* if all the formulas in $Supp(\tau)$ are universal. As an immediate consequence of (4.6), we have the following proposition:

Proposition 4.5. *Let* τ *be an universal fuzzy theory,* (D,I) *be an* $\boldsymbol{L}$*-interpretation and* $(\mathcal{U}, \mathcal{I})$ *be the H-interpretation associated with* (D,I)*. Then*

(D,I) *is a model of* τ $\Rightarrow$ $(\mathcal{U}, \mathcal{I})$ *is a model of* τ.

The following theorem shows that we can define the fuzzy set of facts which are logical consequences of an universal fuzzy theory, by referring only to the fuzzy Herbrand models of the theory.

Theorem 4.6. *Let* τ *be an universal fuzzy theory. Then, for any fact* φ,

$$Lc(\tau)(\varphi) = Inf\,\{||\varphi||_{\mathcal{I}} : \mathcal{I} \text{ is an H-model of } \tau\}. \tag{4.7}$$

Proof. Let φ be a fact. Then, since by (4.4), $||\varphi||_{\mathcal{I}} = ||\varphi||_I$,

$$\begin{aligned} Lc(a)(\varphi) &= Inf\{||\varphi||_I : I \text{ is a model of } \tau\} \\ &= Inf\{||\varphi||_{\mathcal{I}} : \mathcal{I} \text{ is the } H\text{-interpretation associated with a model } I \text{ of } \tau\}. \\ &= Inf\,\{||\varphi||_{\mathcal{I}} : \mathcal{I} \text{ is an } H\text{-model of } \tau\}. \end{aligned}$$

□

Let $\tau : F \to \boldsymbol{L}$ be a universal fuzzy theory. Then we define the *fuzzy set* $Gr(\tau)$ *of ground instances of the formulas in* τ by setting, for every formula α,

$$Gr(\tau)(\alpha) = \begin{cases} \mathbf{0} & \text{if } \alpha \text{ is not ground,} \\ Sup\{\tau(\forall x_1 ... \forall x_n(\alpha')) : \alpha = \alpha'(t_1,...,t_n) \text{ for some } t_1,...,t_n \in \mathcal{U}\}, & \text{otherwise.} \end{cases}$$

The following theorem shows that if τ is an universal theory, then τ and $Gr(\tau)$ have the same H-models.

Theorem 4.7. *Let τ be an universal fuzzy theory, then, for any H-interpretation I,*

$$I \vDash \tau \quad \Leftrightarrow \quad I \vDash Gr(\tau).$$

Proof. Assume that $I \vDash \tau$ and let α be a formula. Then, if α is not ground, $Gr(\tau)(\alpha) = 0$ and therefore $||\alpha||_I \geq Gr(\tau)(\alpha)$. Assume that α is ground and let α' be a formula such that $\alpha = \alpha'(t_1,...,t_n)$ for some $t_1,...,t_n \in \mathcal{U}$. Then

$$||\alpha||_I \geq ||\forall x_1 ... \forall x_n(\alpha')||_I \geq \tau(\forall x_1 ... \forall x_n(\alpha')).$$

Consequently,

$$||\alpha||_I \geq Sup\{\tau(\forall x_1 ... \forall x_n(\alpha')) : \alpha = \alpha'(t_1,...,t_n) \text{ for some } t_1,...,t_n \in \mathcal{U}\} = Gr(\tau)(\alpha),$$

and this proves that I is a model of $Gr(\tau)$.

Conversely, let I be a H-model of $Gr(\tau)$ and let $\forall x_1 ... \forall x_n(\alpha')$ be any closed universal formula. Then, in accordance with the definition of $Gr(\tau)$, given $t_1,...,t_n$ in $\mathcal{U}$,

$$||\alpha'(t_1,...,t_n)||_I \geq Gr(\tau)(\alpha'(t_1,...,t_n)) \geq \tau(\forall x_1 ... \forall x_n(\alpha')).$$

Thus,

$$||\forall x_1 ... \forall x_n(\alpha')||_I = Inf\{||\alpha'(t_1,...,t_n)||_I : t_1,...,t_n \in \mathcal{U}\} \geq \tau(\forall x_1 ... \forall x_n(\alpha'))$$

and this proves that I is a model of τ. □

5. Fuzzy logic programming

In this section and in Section 6 we will expose some definitions and results which are well known in literature (see, for example, [2], [13], [17] and [18]). To define the notion of fuzzy program, we assume that a sequence $\otimes_1,\dots,\otimes_k$ of commutative and associative operations is defined in $\boldsymbol{L}$. We assume also that, for any x in $\boldsymbol{L}$, $x \otimes_k \mathbf{1} = x$ and, for any family $(y_j)_{j\in J}$ of elements in $\boldsymbol{L}$,

$$x \otimes_i (Sup_{j\in J} y_j) = Sup_{j\in J}(x \otimes_i y_j).$$

Under these conditions, we can define the implication operations $\to_i$ by setting, for any $x, y \in \boldsymbol{L}$,

$$x \to_i y = Sup\{z \in \boldsymbol{L} : x \otimes_i z \leq y\}.$$

It is easy to prove that each pair $\to_i$ and $\otimes_i$ satisfies the *adjunction property*, i.e, for any a, b, x in $\boldsymbol{L}$,

$$a \to_i b \geq x \;\Leftrightarrow\; b \geq x \otimes_i a.$$

We suppose that $\mathcal{L}$ contains logical connectives $\to_1,\dots,\to_k$ corresponding to the operations $\boldsymbol{\to_1},\dots,\boldsymbol{\to_k}$. Let $\mathcal{L}^*$ be the sentential calculus whose logical connectives are the same as in $\mathcal{L}$. Then we write $h(p_1,...,p_n)$ to denote a formula of $\mathcal{L}^*$ whose propositional variables are among $p_1,...,p_n$. As usual, we can associate any formula $h(p_1,...,p_n)$ with a *truth-table* $\underline{h} : \mathbf{L}^n \to \mathbf{L}$. Let $\alpha_1,...,\alpha_n$ be atomic formulas of $\mathcal{L}$ and $h(\alpha_1,...,\alpha_n)$ be the formula in $\mathcal{L}$ obtained from $h(p_1,...,p_n)$ by substituting each occurrence of p_i with α_i. Then we say that $h(\alpha_1,...,\alpha_n)$ is a *combination* of the formulas $\alpha_1,...,\alpha_n$. If $\underline{h}$ preserves the least upper bounds (and therefore is monotone), then we say that $h(\alpha_1,...,\alpha_n)$ is a *positive combination of* $\alpha_1,...,\alpha_n$. Trivially, this happens every time $h(p_1,...,p_n)$ is defined only by conjunctions and disjunctions interpreted by continuous norms and co-norms, respectively.

Definition 5.1. A (*positive*) *implicative clause* is either an atomic formula or a formula like $h(\alpha_1,...,\alpha_n)\to_i\alpha$ where α, $\alpha_1,\dots,\alpha_n$ are atomic formulas and $h(\alpha_1,...,\alpha_n)$ is a (positive) combination of $\alpha_1,\dots,\alpha_n$. A fuzzy subset p: $F\to\mathbf{L}$ of formulas is a (*positive*, *ground*) *fuzzy program* if $Supp(p)$ is a set of (positive, ground) implicative clauses.

A simple characterization of the fuzzy models of a fuzzy program is the following.

Proposition 5.2. *Let $p : F \to \mathbf{L}$ be a fuzzy program and (D,I) be an interpretation. Then I is a fuzzy model of p if and only if,*

i) $I(\alpha)(d_1,...,d_h) \geq p(\alpha)$

for any atomic formula $\alpha \in Supp(p)$ and $d_1,...,d_h$ in D ;

ii) $I(\alpha)(d_1,...,d_h) \geq p(h(\alpha_1,...,\alpha_n)\to_i\alpha)\otimes_i\underline{h}(I(\alpha_1)(d_1,...,d_h),...,I(\alpha_n)(d_1,...,d_h))$

for any implicative clause $h(\alpha_1,...,\alpha_n)\to_i\alpha$ in $Supp(p)$ and $d_1,...,d_h$ in D.

Proof. An interpretation I is a model of p if and only if,

$$\|\alpha\|_I = Inf\,\{I(\alpha)(d_1,...,d_h) : d_1,...,d_h\in D\} \geq p(\alpha)$$

for any atomic formula $\alpha \in Supp(p)$ and

$$\|h(\alpha_1,...,\alpha_n)\to_i\alpha\|_I = Inf\{I(h(\alpha_1,...,\alpha_n)\to_i\alpha)(d_1,...,d_h) : d_1,...,d_h\in D\}\geq p(h(\alpha_1,...,\alpha_n)\to_i\alpha),$$

for any formula $h(\alpha_1,...,\alpha_n)\to_i\alpha$ in $Supp(p)$. The first condition is equivalent to the claim that $I(\alpha)(d_1,...,d_h) \geq p(\alpha)$ for any $d_1,...,d_h$ in D and therefore it is equivalent with i). The second condition is equivalent to say that, for any $d_1,...,d_h$ in D,

$$I(h(\alpha_1,...,\alpha_n)\to_i\alpha)(d_1,...,d_h) = \underline{h}(I(\alpha_1)(d_1,...,d_h),...,I(\alpha_n)(d_1,...,d_h))\to_i I(\alpha)(d_1,...,d_h)$$
$$\geq p(h(\alpha_1,...,\alpha_n)\to_i\alpha).$$

By the adjunction property, such an inequality is equivalent with ii). □

By Propositions 4.2, 5.2 and Theorem 4.7, we obtain the following theorem:

Theorem 5.3. *Let $p : F \to \mathbf{L}$ be a fuzzy program and $s : B_{\mathcal{L}} \to \mathbf{L}$ be a fuzzy set of facts. Then s is a H-model of p if and only if for any fact $\alpha \in Supp(Gr(p))$,*

i) $s(\alpha) \geq Gr(p)(\alpha)$

and, for any implicative clause $h(\alpha_1,...,\alpha_n)\to_i\alpha$ in $Supp(Gr(p))$,

ii) $s(\alpha) \geq Gr(p)(h(\alpha_1,...,\alpha_n)\to_i\alpha)\otimes_i \underline{h}(s(\alpha_1)...,s(\alpha_n))$.

To calculate the H-models of a fuzzy program, it is very useful the notion of immediate consequence operator (see, also, P. Vojtas [17] and [18]).

Definition 5.4. Let $p : F \to \mathbf{L}$ be a fuzzy program. Then the *immediate consequence operator associated with p* is the operator $T_p : \mathcal{F}(B_{\mathcal{L}}) \to \mathcal{F}(B_{\mathcal{L}})$ defined by setting, for any $s\in\mathcal{F}(B_{\mathcal{L}})$ and $\alpha \in B_{\mathcal{L}}$,

$T_1(s)(\alpha) = Gr(p)(\alpha)\vee s(\alpha)$,

$T_2(s)(\alpha) = (Sup_{h(\alpha_1,...,\alpha_n)\to_i\alpha\in Supp(Gr(p))}\, Gr(p)(h(\alpha_1,...,\alpha_n)\to_i\alpha)\otimes_i\underline{h}(s(\alpha_1),...,s(\alpha_n)))$,

$T_p(s) = T_1(s)\cup T_2(s)$.

Observe that T_p is not monotonic, in general and that inclusion property $T_p(s)\supseteq s$ is satisfied by definition. As in the classical case, the immediate consequence operator enables us to give the following elegant and useful characterization of the H-models of a fuzzy program.

Theorem 5.5. *Let $p : F \to \mathbf{L}$ be a fuzzy program and $s \in \mathcal{F}(B_{\mathcal{L}})$. Then s is an H-model of p if and only if s is a fixed point for the immediate consequence operator T_p.*

Proof. It is sufficient to apply Theorem 5.3. Indeed, let s be a fixed point for T_p, i.e. assume that $T_p(s)\subseteq s$. Then we have $T_1(s)\subseteq s$ and $T_2(s)\subseteq s$. Let α be a fact in $Supp(Gr(p))$. Then, since $s(\alpha) \geq T_1(s)(\alpha) \geq Gr(p)(\alpha)$, we have that condition i) is satisfied. Let $h(\alpha_1,...,\alpha_n)\to_i\alpha$ be a ground implicative clause in $Supp(Gr(p))$. Then, since

$$s(\alpha) \geq T_2(s)(\alpha) = Sup_{h(\alpha_1,...,\alpha_n)\to_i\alpha\in Supp(Gr(p))}\, Gr(p)(h(\alpha_1,...,\alpha_n)\to_i\alpha) \otimes_i \underline{h}(s(\alpha_1),...,s(\alpha_n)),$$

we have that

$$s(\alpha) \geq Gr(p)(h(\alpha_1,...,\alpha_n)\to_i\alpha) \otimes_i \underline{h}(s(\alpha_1),...,s(\alpha_n))$$

for any $h(\alpha_1,...,\alpha_n)\to_i\alpha$ in $Supp(p)$. Hence, condition ii) is satisfied. This proves that s is a H-model of p.

Conversely, assume that s is a H-model of p and therefore that both i) and ii) are satisfied. Then, by i), we have that, for any fact α, $s(\alpha) \geq Gr(p)(\alpha)$ and therefore $s(\alpha) \geq T_1(\alpha)$. Moreover, by ii),

$$s(\alpha) \geq Gr(p)(h(\alpha_1,...,\alpha_n)\to_i\alpha) \otimes_i \underline{h}(s(\alpha_1),...,s(\alpha_n))$$

and therefore

$$s(\alpha) \geq (\, Sup_{h(\alpha_1,...,\alpha_n)\to_i\alpha\in Supp(Gr(p))}\, Gr(p)(h(\alpha_1,...,\alpha_n)\to_i\alpha)\otimes_i\underline{h}(s(\alpha_1),...,s(\alpha_n)))\vee s(\alpha) = T_2(s)(\alpha).$$

This proves that $s(\alpha) \geq T_p(s)(\alpha)$. Thus, s is a fixed point for T_p. □

6. Calculus of the fuzzy Herbrand models.

In accordance with Theorem 5.3, any fuzzy program p admits as a H-model the map $s : B_{\mathcal{L}}\to\mathbf{L}$ constantly equal to 1. We are interested to the H-models which represents, in a sense, the informative content of p. As an example, we are interested to the *least Herbrand model* of a program p, i.e. to a H-model $m_p : B_{\mathcal{L}} \to \mathbf{L}$ of p which is contained in any H-model of p. In order to prove the existence of m_p and to calculate m_p, we recall some basic results in the theory of fixed points in a complete lattice. We say that a family $(s_i)_{i\in I}$ of fuzzy subsets of a set S is *directed* provided that for every s_i and s_j there is s_h such that $s_i\subseteq s_h$ and $s_j\subseteq s_h$. An operator $J : \mathcal{F}(S) \to \mathcal{F}(S)$ is *continuous* provided that

$$J(\cup_{i\in I}s_i) = \cup_{i\in I}J(s_i)$$

for any directed family $(s_i)_{i\in I}$ of fuzzy subsets of S. A *continuous almost closure operator* is a continuous operator J such that $J(s) \supseteq s$.

Proposition 6.1. *Let $J_1 : \mathcal{F}(S) \to \mathcal{F}(S)$ and $J_2 : \mathcal{F}(S) \to \mathcal{F}(S)$ be two operators satisfying the continuity property. Then the union $J = J_1\cup J_2$ satisfies the continuity property.*

Proof. Let $(s_j)_{j\in J}$ be a directed family of fuzzy subsets of S. Then

$$J(\cup_{i\in I}s_i) = J_1(\cup_{i\in I}s_i)\cup J_2(\cup_{i\in I}s_i) = (\cup_{i\in I}J_1(s_i))\cup(\cup_{i\in I}J_2(s_i)) = \cup\{J_1(s_i)\cup J_2(s_j) : i\in I, j\in I\}.$$

Now, since $(s_j)_{j\in J}$ is directed, we have that any $i\in I$ and $j\in I$, there is $h\in I$ such that $s_i\subseteq s_h$ and $s_j\subseteq s_h$, and therefore $J_1(s_i) \subseteq J_1(s_h)$ and $J_2(s_j)\subseteq J_2(s_h)$. Then,

$$J(\cup_{i\in I}s_i) = \cup\{J_1(s_i)\cup J_2(s_j) : i\in I, j\in I\} = \cup_{h\in I}(J_1(s_h)\cup J_2(s_h)) = \cup_{h\in I}J(s_h).$$ □

The following theorem summarizes the main properties of the continuous almost closure operators (see, for example, [7]).

Theorem 6.2. *Let $J : \mathcal{F}(S) \to \mathcal{F}(S)$ be a continuous almost closure operator, then:*

- the class of fixed points of J is a closure system, i.e. it is closed under arbitrary intersections,
- the least fixed point operator $D : \mathcal{F}(S) \to \mathcal{F}(S)$ *defined by setting, for any fuzzy subset s,*

$$D(s) = \bigcap \{s' : J(s) = s, s' \supseteq s\},$$

is a closure operator
- for any fuzzy subset s,

$$D(s) = \bigcup_{n\in N} J^n(s). \quad (6.1)$$

We can apply this theorem to the immediate consequence operator.

Theorem 6.3. *Let p be a positive fuzzy program. Then* T_p *is a continuous almost closure operator. As a consequence, the class of fuzzy H-models of a positive fuzzy program p is a closure system and the least fuzzy Herbrand model* m_p *for p is given by*:

$$m_p = \bigcap \{s \in \mathcal{F}(B_{\mathcal{L}}) : s \text{ is a model of } p\}. \quad (6.2)$$

Moreover,

$$m_p = \bigcup_{n\in N} T_p{}^n(\emptyset). \quad (6.3)$$

Proof. Let T_1 and T_2 be as in Definition 5.4. Then it is immediate that T_1 satisfies the continuity property. To prove that T_2 satisfies the continuity property, we assume that $(s_j)_{j\in J}$ is a directed family of fuzzy subsets of $B_{\mathcal{L}}$. Moreover, in order to simplify our notations, we denote the expression $Sup_{h(\alpha_1,...,\alpha_n)\to_i \alpha \in Supp(Gr(p))}$ by *Sup*. Then

$$T_2(\bigcup_{j\in J} s_j)(\alpha) = Sup\{Gr(p)(h(\alpha_1,...,\alpha_n)\to_i\alpha)\otimes_i \underline{h}(Sup_{j\in J}s_j(\alpha_1),..., Sup_{j\in J}s_j(\alpha_n))\}$$
$$= Sup\{Gr(p)(h(\alpha_1,...,\alpha_n)\to_i\alpha)\otimes_i(Sup_{j(1),...,j(n)}\underline{h}(s_{j(1)}(\alpha_1),...,s_{j(n)}(\alpha_n))\}.$$

Now, we observe that, since $(s_j)_{j\in J}$ is directed, given $s_{j(1)},..., s_{j(n)}$ there is $j\in J$ such that $s_{j(1)}\subseteq s_j,..., s_{j(n)}\subseteq s_j$ and therefore $\underline{h}(s_{j(1)}(\alpha_1),..., s_{j(n)}(\alpha_n)) \le \underline{h}(s_j(\alpha_1),..., s_j(\alpha_n))$. Consequently,

$$T_2(\bigcup_{j\in J} s_j)(\alpha) = Sup\{Gr(p)(h(\alpha_1,...,\alpha_n)\to_i\alpha)\otimes(Sup_{j\in J}\underline{h}(s_j(\alpha_1),..., s_j(\alpha_n))\}$$
$$= Sup_{j\in J}\, Sup\{Gr(p)(h(\alpha_1,...,\alpha_n)\to_i\alpha)\otimes(\underline{h}(s_j(\alpha_1),..., s_j(\alpha_n))\}$$
$$= Sup_{j\in J} T_2(s_j)(\alpha) = \bigcup_{j\in J} T_2(s_j)(\alpha)$$

and this proves that T_2 is continuous. The remaining part of the theorem is trivial. □

Equation (6.3) suggests an algorithm to calculate, for any fact α, the value $m_p(\alpha)$. More precisely, if we adopt the definition of recursive enumerability for fuzzy sets proposed in [1] and [6], then, under very natural hypotheses, it is easy to show that m_p is a recursively enumerable fuzzy subset of $B_{\mathcal{L}}$.

The following theorem shows that the least fuzzy Herbrand model of a positive fuzzy program represents the informative content of *p*.

Theorem 6.4. *Let p be a positive fuzzy program. Then the least fuzzy Herbrand model of p is equal to the fuzzy subset of facts which are logical consequences of p, i.e., for any fact* α,

$$m_p(\alpha) = Lc(p)(\alpha). \quad (6.4)$$

Proof. By Theorem 4.6 and formula (6.2), for any fact α,

$$Lc(p)(\alpha) = Inf\{\|\alpha\|_{\mathcal{I}} : \mathcal{I} \text{ is an H-model of } p\} = m_p(\alpha). \quad □$$

As in the classical case, several difficulties exist for fuzzy programs which are non positive. We confine ourselves only to the very simple case of hierarchical fuzzy programs. We call *level mapping* any map $lev : B_{\mathcal{L}}\to N$ from the Herbrand base to the set of natural numbers. We say that a fuzzy program *p* is *hierarchical* if a level mapping exists such that for any rule $h(\alpha_1,...,\alpha_n)\to\alpha$ in $Supp(Gr(p))$, $lev(\alpha)>lev(\alpha_1),..., lev(\alpha)>lev(\alpha_n)$. Trivially, it is no restrictive to assume that the level of the atomic formulas is equal to 1.

Proposition 6.5. *Let p be a hierarchical fuzzy program. Then, for any* $\alpha \in B_{\mathcal{L}}$,

$$lev(\alpha) \leq n \ \Rightarrow\ T_p^{n+1}(\varnothing)(\alpha) = T_p^{n}(\varnothing)(\alpha) \qquad (6.5)$$

Consequently, $\bigcup_{n \in N} T_p{}^n(\varnothing)$ *is a H-model of p.*

Proof. We will prove (6.5) by induction on n. In the case $n = 1$, we have that α is atomic and that no rule in $Supp(Gr(p))$ exists whose head is α. Then, $T_2(T_p(\varnothing))(\alpha) = 0$ and $T_2(\varnothing)(\alpha) = 0$. Therefore, since $T_p(\varnothing)(\alpha) = Gr(p)(\alpha)$,

$$T_p^2(\varnothing)(\alpha) = T_p(T_p(\varnothing))(\alpha) = T_1(T_p(\varnothing))(\alpha) \vee T_2(T_p(\varnothing))(\alpha) = T_1(T_p(\varnothing))(\alpha)$$
$$= Gr(p)(\alpha) \vee T_p(\varnothing)(\alpha) = T_p(\varnothing)(\alpha).$$

Assume that $n \neq 1$, that $lev(\alpha) \leq n$ and that $h(\alpha_1, \ldots \alpha_n) \rightarrow_i \alpha$ is a rule in $Supp(Gr(p))$. Then, since $lev(\alpha_i) < lev(\alpha) \leq n$ and therefore $lev(\alpha_i) \leq n\text{-}1$, by inductive hypothesis $T_p^{n}(\varnothing)(\alpha_i) = T_p^{n-1}(\varnothing)(\alpha_i)$. Then,

$$T_p^{n+1}(\varnothing)(\alpha) = Gr(p)(\alpha) \vee T_p^{n}(\varnothing)(\alpha) \vee$$
$$(Sup_{h(\alpha_1,\ldots,\alpha_n) \rightarrow_i \alpha \in Supp(Gr(p))}\, Gr(p)(h(\alpha_1,\ldots,\alpha_n) \rightarrow_i \alpha) \otimes_i \underline{h}(T_p^{n}(\alpha_1), \ldots, T_p^{n}(\alpha_n))).$$
$$= Gr(p)(\alpha) \vee T_p^{n}(\varnothing)(\alpha) \vee$$
$$(Sup_{h(\alpha_1,\ldots,\alpha_n) \rightarrow_i \alpha \in Supp(Gr(p))}\, Gr(p)(h(\alpha_1,\ldots,\alpha_n) \rightarrow_i \alpha) \otimes_i \underline{h}(T_p^{n-1}(\alpha_1), \ldots, T_p^{n-1}(\alpha_n))).$$

By noticing that $Gr(p)(\alpha) \leq T_p^{n}(\varnothing)(\alpha)$ and

$$(Sup_{h(\alpha_1,\ldots,\alpha_n) \rightarrow_i \alpha \in Supp(Gr(p))}\, Gr(p)(h(\alpha_1,\ldots,\alpha_n) \rightarrow_i \alpha) \otimes_i \underline{h}(T_p^{n-1}(\alpha_1), \ldots, T_p^{n-1}(\alpha_n))) \leq T_p^{n}(\varnothing)(\alpha),$$

we obtain that $T_p^{n+1}(\varnothing)(\alpha) \leq T_p^{n}(\varnothing)(\alpha)$ and therefore $T_p^{n+1}(\varnothing)(\alpha) = T_p^{n}(\varnothing)(\alpha)$. □

In the following we say that $\bigcup_{n \in N} T_p{}^n(\varnothing)$ is the *canonical H-model* of *p*.

7. Fuzzy control

As an application of the sketched fuzzy logic programming theory, we will consider the main success of fuzzy set theory: *fuzzy control*. Recall that the aim of classical control theory is to individuate a function $\underline{f} : X \rightarrow Y$ such that $\underline{f}(x)$ gives the correct control given the input x. To do this, the starting point is a general theory about the phenomenon under consideration. From this theory we obtain some differential equations and then $\underline{f}$ is obtained as a solution of these equations. The paradigm used in fuzzy control theory, as devised by Zadeh in [19], [20] and by Mamdani in [12], is totally different. Indeed, in fuzzy control one tries to obtain $\underline{f}$ from the verbal information given by an expert on the control under consideration (imagine the expert as a cleaver "old hand" with no theoretical knowledge or mathematical competence). The expert expresses its experience by a system of IF-THEN rules like

$$\begin{cases} \text{If } x \textit{ is Little}, \text{ then I suggest to put } y \textit{ Slow}, \\ \text{If } x \textit{ is Small}, \text{ then I suggest to put } y \textit{ Fast}, \\ \text{If } x \textit{ is Medium}, \text{ then I suggest to put } y \textit{ Moderate}, \\ \text{If } x \textit{ is Big}, \text{ then I suggest to put } y \textit{ Veryfast}, \\ \text{If } x \textit{ is Verybig}, \text{ then I suggest to put } y \textit{ Moderate}. \end{cases} \qquad (7.1)$$

Successively, fuzzy control theory interprets such an information, qualitative in nature, by translating it into a function $f' : X \rightarrow Y$ in such a way that f' can be considered an adequate control function. This is done by suitable procedures based on the interpretation the words *Little*, *Slow*, ... by suitable *fuzzy quantities*, or *fuzzy granules*, *little* : $X \rightarrow [0,1]$, *slow* : $Y \rightarrow [0,1]$, ... By proceeding in a general way, consider a general IF-THEN system:

$$\begin{cases} \text{IF } x \text{ is } A_1 \text{ THEN } y \text{ is } B_1, \\ \quad \ldots \\ \text{IF } x \text{ is } A_n \text{ THEN } y \text{ is } B_n. \end{cases} \qquad (7.2)$$

Then, in fuzzy control such a system can be interpreted in two different ways. In the *conjunction-based* fuzzy control we consider the following procedure:

- **Step 1.** A tentative interpretation of the constants $A_1,\ldots,A_n$, $B_1,\ldots,B_n$ by suitable *"fuzzy quantities"* $a_1 : X \to [0,1],\ldots, a_n : X \to [0,1]$, $b_1 : Y \to [0,1]$, $b_n : Y \to [0,1]$ is proposed.
- **Step 2.** Each rule "IF x is A_i THEN y is B_i" is associated with the *fuzzy point* $a_i \times b_i$ obtained as the $\otimes$-*Cartesian* product of the fuzzy quantities a_i and b_i.
- **Step 3.** The whole system of rules is associated with the fuzzy function $f : X \times Y \to [0,1]$ defined as an union of these fuzzy points, i.e.

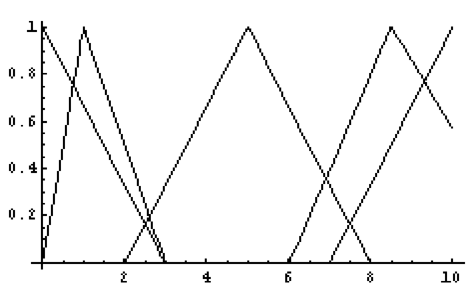
$$f = \cup_{i=1,\ldots,n}(a_i \times b_i). \qquad (7.3)$$

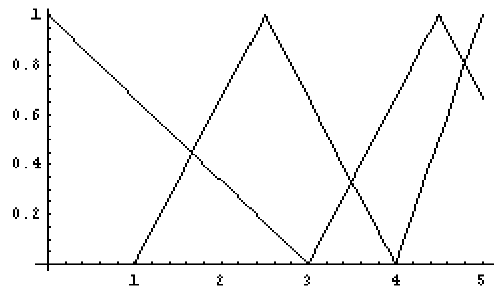
- **Step 4.** A suitable *defuzzification* process associates the fuzzy function f with a crisp function f'. Usually, this is obtained by the *centroid method*, where we set, for every $r \in X$,

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

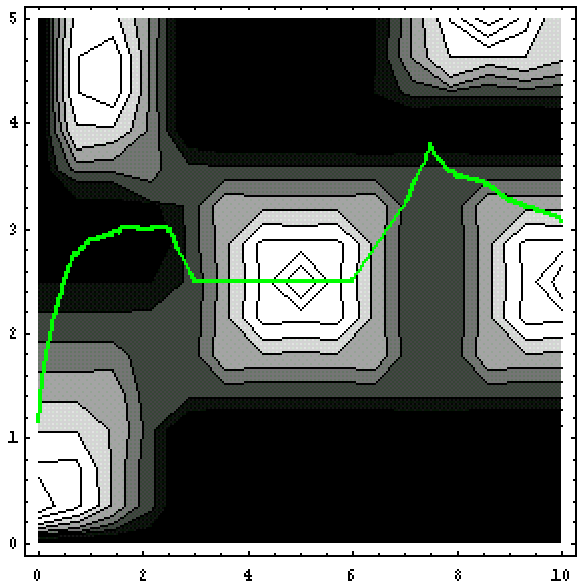
- **Step 5.** If f' is satisfactory (it is sufficiently close to f), then the procedure terminates, otherwise the interpretation of the expert's words is chanced in accordance with a suitable strategy. After this, one goes back to Step 2.

Notice that the basic feature of such a procedure is the *tuning process*, where the interpretation of the constants $A_1,\ldots,A_n,B_1,\ldots,B_n$ is changed until we obtain a satisfactory interpretation. By referring to the system (6.1), set $X = [0,10]$, $Y = [0,5]$ and interpret the constants *Little*, *Small*, *Medioum, Big*, *Verybig*, *Slow*, *Moderate*, *Fast*, *Veryfast* by suitable triangle functions:





Also, consider the triangular norm of the minimum. Then, the following picture represents both the corresponding fuzzy function f and the crisp function f'.



In the *implication-based* procedure (see, for example, [11]), instead of steps 2 and 3 we have:

- Step 2'. Each 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 $y \in Y$, $(a_i \to b_i)(x,y) = a_i(x) \to b_i(y)$.
- Step 3'. The whole system of rules is associated with the fuzzy function:

$$g = \bigcap_{i=1,\dots,n}(a_i \to b_i). \tag{7.4}$$

8. Fuzzy control and logic programming.

In spite of the great success of fuzzy control, there is no convincing justification of the applied techniques. Obviously, the first temptation is to interpret the IF-THEN structure of the rules as a logical implication. As an example, we could interpret a rule as "If x is *Little* then y is *Slow*" by the first order formula $Little(x) \to Slow(y)$. Unfortunately, this cannot be done in a so direct way. Indeed assume that r is an input such that $Little(r)$ is false. Then the formula $Little(r) \to Slow(t)$ is true for any t in Y and therefore any $t \in Y$ gives a correct control.

As a consequence of such considerations, in [8] we sketched a logical approach to fuzzy control based on the idea that, given the input r, and a possible output t, the number $f(r,t)$ is the truth degree of the claim *"t is a good answer given r"*. This means that f is obtained as an interpretation of a vague predicate *Good* in a multivalued first-order logic. Namely, we show that f is the interpretation of *Good* in the least fuzzy Herbrand model of a suitable fuzzy program. To illustrate this idea, we consider the multivalued logic defined by the complete lattice [0,1] equipped with a triangular norm $\otimes$ and the map $\sim : [0,1] \to [0,1]$ defined by setting $\sim(x) = 1-x$. These operations are devoted to interpret the conjunction and the negation connectives, respectively. The disjunction is interpreted by the co-norm $\oplus$ defined by setting $x \oplus y = \sim(\sim x \otimes \sim y)$ and the implication by the operation $\to$ defined by setting $x \to y = (\sim x) \oplus y$. Also, the language $\mathcal{L}$ contains a binary relation symbol *Good* and unary predicate symbols. Also, we consider each element r in X and t in Y as a constant and therefore we assume that the Herbrand universe is $X \cup Y$. Then, we translate the system (7.2) into a fuzzy program $p : F \to [0,1]$ defined by setting:

$$\begin{cases} A_1(x) \wedge B_1(y) \to Good(x,y) & [1] \\ \dots & \\ A_n(x) \wedge B_n(y) \to Good(x,y) & [1] \\ A_1(r) & [a_1(r)] \\ \dots & \dots \\ A_n(r) & [a_n(r)] \\ B_1(t) & [b_1(t)] \\ \dots & \dots \\ B_n(t) & [b_n(t)], \end{cases} \tag{8.1}$$

where $r \in X$ and $t \in Y$. More precisely, p is defined by setting

$$p(\alpha) = \begin{cases} 1 & \text{if } \alpha \text{ is the clause } A_i(x) \wedge B_i(y) \to Good(x,y), \\ a_i(r) & \text{if } \alpha \text{ is the fact } A_i(r),\ r \in X \\ & \\ b_i(t) & \text{if } \alpha \text{ is the fact } B_i(t),\ t \in Y \\ 0 & \text{otherwise.} \end{cases}$$

Observe that $A_1,\dots,A_n$, $B_1,\dots,B_n$ are considered as unary predicate symbols and not constants for fuzzy quantities as in the system (7.1). Observe also that, while $p(A_i(r)) = a_i(r)$ for any $r \in X$, we have set $p(A_i(t)) = 0$ for any $t \in Y$, and that while $p(B_i(t)) = b_i(t)$ for any $t \in Y$, we have set $p(A_i(r)) = 0$ for any $r \in X$ (it is not restrictive to assume that $X \cap Y = \emptyset$). A better alternative should be to extend the results of this paper into a two sorted logic, one type for the elements in X and one type for the elements in Y.

The following theorem shows that the calculus of the fuzzy function f obtained by the conjunction-procedure is equivalent to the calculus of the least fuzzy Herbrand model of such a fuzzy program.

Theorem 8.1. *Consider the fuzzy control system* (7.2)*, let f be the associated fuzzy function by the conjunction procedure and let p be the fuzzy program given by* (8.1)*. Then, for any* $r\in X$ *and* $t\in Y$,

$$f(r,t) = m_p(Good(r,t)),$$

i.e., f is the extension of the predicate Good in the least fuzzy Herbrand model of p.

Proof. We observe only that, due to the one-level structure of the program p, $m_p = T_p^2(\varnothing)$. □

Likewise, in order to give a logical interpretation of implication-based fuzzy control, we consider a new binary predicate symbol *Bad* and the following hierarchical fuzzy program p:

$$\begin{cases} A_1(x)\wedge\neg B_1(y) \to Bad(x,y) & [1] \\ \dots & \\ A_n(x)\wedge\neg B_n(y) \to Bad(x,y) & [1] \\ A_1(r) & [a_1(r)] \\ \dots & \dots \\ A_n(r) & [a_n(r)] \\ B_1(t) & [b_1(t)] \\ \dots & \dots \\ B_n(t) & [b_n(t)], \end{cases} \tag{8.2}$$

where $r\in X$ and $t\in Y$. The following theorem shows that the calculus of the fuzzy function f obtained by the implication-procedure is equivalent to the calculus of the canonical fuzzy Herbrand model of such a fuzzy program.

Theorem 8.2. *Consider the fuzzy control system* (7.2)*, let g be the fuzzy function obtained by the implication-procedure and let p be the fuzzy program defined by* (8.1)*. Then, for any* $r\in X$ *and* $t\in Y$,

$$g(r,t) = \sim m_p(Bad(r,t)),$$

i.e, g is the extension of the predicate $\neg Bad$ *in the canonical Herbrand model of p.*

Proof. It is immediate that $m_p(Bad(r,t)) = Max_{i=1,\dots,n}\{a_i(r)\otimes\sim b_i(t)\}$ and therefore that,

$$\sim m_p(Bad(r,t)) = Min_{i=1,\dots,n} \sim(a_i(r)\otimes\sim b_i(t))$$
$$= Min_{i=1,\dots,n} (\sim a_i(r)\oplus b_i(t)) = \cap_{i=1,\dots,n}(a_i(r)\to b_i(t)) = g(r,t).$$ □

Theorems 8.1 and 8.2 enable us to emphasize the different meaning of the two procedures (see, also, [3]). Indeed, while conjunction-based procedure is useful to give positive information (given r, the fuzzy subset $good(r,y)$ of outputs that we consider good), the implication-based procedure is useful to manage negative information (given r, the fuzzy subset $bad(r,y)$ of outputs that we suggest to avoid).

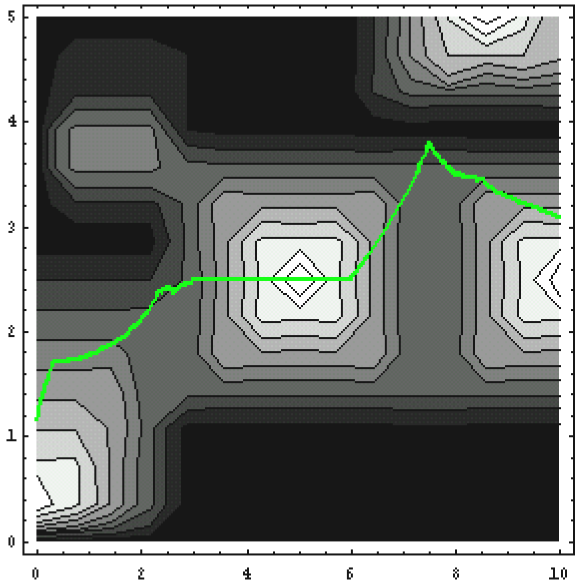
Also, this approach enables us to consider both the approaches at the same time. In fact we can consider a hierarchical fuzzy program containing both rules for the predicate *Good* and rules for the predicate *Bad*. As an example, we can consider the hierarchical fuzzy program obtained by adding at the fuzzy program (8.1), the rule

$(\neg Big(x))\wedge(Fast(y)\vee Verifast(y)) \to Bad(x,y)$ [1]

able to define *Bad* and the rule

$Good(x,y)\wedge\neg Bad(x,y) \to Optimum(x,y)$ [1]

to compose the two different kinds of information. In such a case the fuzzy function we have to consider is the extension of the predicate *Optimum* and not to the extension of the predicate *Good* in the successive defuzzification process, obviously. In the following picture we represent both such a fuzzy function and the corresponding crisp function f' obtained by the defuzzification process.



Notice that such an approach to fuzzy control is related with the approach sketched by P. Hájek in [10]. In fact, Hájek defined the predicate *Mamd*(x,y) by the axiom

$$Mamd(x,y) \leftrightarrow (A_1(x)\wedge B_1(y)) \vee\ldots\vee (A_1(x)\wedge B_1(y)).$$

It is immediate that in any *H*-model in which such an axiom is satisfied, the extension of the predicate *Mamd* is the fuzzy relation defined in fuzzy control. Then, while we associate any IF-THEN system with a fuzzy program p, Hájek refers to the completion of p.

9. Possible future researches on fuzzy control

As a matter of fact the proposed immersion of fuzzy control into fuzzy logic programming uses only a very small piece of this theory. Then, we can try to extend fuzzy control taking in account the whole potentialities of fuzzy logic programming. For example, we can indicate the following research directions.

9.1. Learning process. In defining the fuzzy programs (8.1), we can assign to a rule $A_i(x)\wedge B_i(y) \rightarrow Good(x,y)$ a weight λ_i different form 1 as a measure of the corresponding *"degree of confidence"* of the expert in such a rule. Then, the interpretation f of *Good* in the resulting canonical Herbrand model m_p is obtained by the expression

$$f = \cup_{i=1,\ldots,n} (\lambda_i \otimes (a_i \times b_i)). \qquad (9.1)$$

This means that, in the learning process, we can modify both the interpretations of the vague predicates and the parameters $\lambda_1,\ldots,\lambda_n$. Obviously, it is also possible that in such a process a parameter λ_i becomes equal to zero and therefore that the i-rule is deleted.

9.2. Further connectives and linguistic modifiers. We can consider a fuzzy logic with several kind of implication and further logical connectives in [0,1]. For instance, we can consider *linguistic modifiers* as *"Clearly"* and *"Vaguely"* interpreted by the functions $cl : [0,1]\rightarrow [0,1]$ and $vag : [0,1]\rightarrow [0,1]$ defined by setting $cl(x) = x^2$ and $vag(x) = x^{0.5}$ for any $x\in[0,1]$, respectively. Then we can consider a fuzzy control able to manage rules as

$Vaguely(Little(x)) \wedge Fast(y) \rightarrow Good(x,y)$,

$Little(x) \wedge Clearly(Slow(y)) \rightarrow Good(x,y)$

where $Vaguely(Little(x))$ is interpreted by the fuzzy subset $little(x)^{0.5}$ and $Clearly(Slow(y))$ is interpreted by the fuzzy set $slow(y)^2$.

9.3. Fuzzy control based on bilattices. We can imagine a fuzzy control based on a logic programming with a valuation structure ***L*** different from the interval [0,1]. For example, we can assume that S is a set of expert and that ***L*** is the power set Boolean algebra $\wp(S)$. In such a case a fuzzy program p associates any clause α with the set $p(\alpha)$ of experts whose opinion is that α is valid. In accordance with [4], should be very interesting to assume that ***L*** is a bilattice based on $[0,1]\times[0,1]$. The resulting fuzzy control should be able to manage both positive and negative information.

9.4. Rule chaining. The power of logic programming is mainly based on recursion and on many-steps deductions. Instead fuzzy control usually considers one-step deductions. Then, our translation suggests a fuzzy control able to define predicates which, in turn, are used to define new predicates and so on. A simple example is furnished in the previous section in defining *Optimum* from *Good* and *Bad*. To give a further example, assume that an element r in X exists such that $a_1(r) = ... = a_n(r) = 0$. Then $f(r,t) = 0$ for any $t\in Y$. This means that the proposed rules are not able to suggest a correct control given r. In such a case we can add to the rules used to define *Good* the default rule

$Undefined(x)\wedge By\_default(y) \rightarrow Good(x,y)$,

Where, in turn, the meaning of the predicate *Undefined* is furnished by the rule:

$\neg A_1(x)\wedge...\wedge\neg A_n(x) \rightarrow Undefined(x)$

and the meaning of $By\_default(y)$ is defined by suitable rules.

References

1. Biacino L., Gerla G., Recursively enumerable *L*-sets, *Zeitschr. f. math. Logik und Grundlagen d. Math.*, 33 (1987) pp. 107-113.
2. Damásio C.V., Pereira L.M., Hybrid probabilistic logic programs as residuated logic programs. In M.O. Aciego, I.P. de Guzmán, G. Brewka, and L.M. Pereira Editors, *Logic in A.I., Proceedings of JELIA'00,* LNAI 1919, Springer-Verlag, 2000, pp. 57-72.
3. Dubois D., Prade H., What are fuzzy rules and how to use them, *Fuzzy Sets and Systems*, 84 (1996) pp. 169-185.
4. Fitting M., Bilattices and semantics of logic programming, *Journal of Logic Programming*, 11 (1991) pp. 91-116.
5. Fontana A., Formato F., Gerla G., Fuzzy unification as a foundation of fuzzy logic programming. In *Logic Programming and Soft Computing*, (T. P. Martin and F. Arcelli Fontana Eds.). Research Studies Press, 1998, pp. 51-68.
6. Gerla G., Decidability, partial decidability and sharpness relation for L-subsets, *Studia Logica*, 46 (1987) pp. 227-238.
7. Gerla G., *Fuzzy logic: Mathematical Tools for Approximate Reasoning*, Kluwer Academic Publishers, Dordrecht, The Netherlands, 2000.
8. Gerla G., Fuzzy control as a fuzzy deduction system, *Fuzzy Sets and Systems*, 121 (2001) pp. 409-425.
9. Goguen J. A., The logic of inexact concepts, *Synthese*, 19 (1968/69) pp. 325-373.
10. Hájek P., *Metamathematics of Fuzzy Logic*, Kluwer Academic Publishers, Dordrecht, The Netherlands, 1998.
11. Klir G., Yuan B., *Fuzzy sets and fuzzy logic; theory and applications*, Prentice Hall, Englewood Cliffs, N. Y. (USA), 1995.
12. Mamdani, E., Application of fuzzy logic to approximate reasoning using linguistic systems, *Fuzzy Sets and Systems* 26 (1977) pp. 1182-1191.
13. Medina J., Ojeda-Arciego M., Vojtas P., Multi-adjoint logic programming with continuous semantics. In *Proc. LPNMR'01*, 2001, to appear.
14. Pavelka J., On fuzzy logic I: Many-valued rules of inference, *Zeitschr. f. math. Logik und Grundlagen d. Math.*, 25 (1979) pp. 45-52.
15. Pavelka J., On fuzzy logic II: Enriched residuated lattices and semantics of propositional calculi, *Zeitschr. f. math. Logik und Grundlagen d. Math.*, 25 (1979) pp. 119-134.

16. Pavelka J., On fuzzy logic III: Semantical completeness of some many-valued propositional calculi, *Zeitschr. f. math. Logik und Grundlagen d. Math.*, 25 (1979) pp. 447-464.
17. Vojtas P., Many valued logic programming handling uncertainty in AI, *Proceedings of LACS*, Warsawa, 1966.
18. Vojtas P., Fuzzy logic programming, *Fuzzy Sets and Systems*, 124 (2001) pp. 361-370.
19. Zadeh L.A., Fuzzy Sets, *Information and Control*, 12 (1965) pp. 338-353.
20. Zadeh L.A., Fuzzy logic and approximate reasoning, *Synthese*, 30 (1965) pp. 407-428.